

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт –  
филиал ФГАОУ ВО «Сибирский федеральный университет»

Кафедра прикладной информатики, математики  
и естественно-научных дисциплин

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Е. Н. Скуратенко

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.03 Прикладная информатика

Разработка виртуальной лабораторной работы по физике

Руководитель \_\_\_\_\_ доцент, канд. пед. наук И. В. Янченко  
подпись, дата

Выпускник \_\_\_\_\_ Е. В. Куриленок  
подпись, дата

Консультант  
по экономическому разделу \_\_\_\_\_ Е. Н. Скуратенко  
подпись, дата

Нормоконтролер \_\_\_\_\_ В. И. Кокова  
подпись, дата

Абакан 2021

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт –  
филиал ФГАОУ ВО «Сибирский федеральный университет»

Кафедра прикладной информатики, математики  
и естественно-научных дисциплин

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Е.Н. Скуратенко

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ  
в форме бакалаврской работы**

Студенту Куриленок Евгению Владимировичу  
Группа ХБ 17-03  
Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы «Разработка виртуальной лабораторной работы по физике»

Утверждена приказом директора № 222 от 08.04.2021 г.

Руководитель ВКР: И.В. Янченко, доцент кафедры ПИМиЕД, канд. пед. наук,  
ХТИ – филиал СФУ

Исходные данные для ВКР: общие требования заказчика к информационной системе виртуальных лабораторных работ, методические указания для выполнения лабораторных работ, шаблоны отчетов по лабораторным работам, фотографии реальных установок для выполнения лабораторных работ.

Перечень разделов ВКР:

1. Аналитический раздел: «Анализ процесса выполнения лабораторных работ по физике в ХТИ – филиале СФУ».
2. Проектный раздел: «Описание разработки информационной системы «Виртуальные лабораторные работы по физике»».
3. Экономический раздел: «Оценка экономической эффективности проекта».

Перечень графического материала: нет.

Руководитель ВКР

\_\_\_\_\_ подпись

И. В. Янченко

Задание принял к исполнению

\_\_\_\_\_ подпись

Е. В. Куриленок

« 08 » апреля 2021 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка виртуальной лабораторной работы по физике» содержит 142 страницы текстового документа, 12 таблиц, 154 рисунка, 15 использованных источников.

JAVA, SCRIPT, HTML, PHP, СЕРВЕР, MYSQL, 3D, ПОЛЬЗОВАТЕЛЬ, РЕГИСТРАЦИЯ ПОЛЬЗОВАТЕЛЯ, ВИРТУАЛЬНАЯ ЛАБОРАТОРНАЯ РАБОТА, THREEJS, МОДЕЛИРОВАНИЕ, IDEF0, IDEF3, DFD, ВЕБ-ПРИЛОЖЕНИЕ, КАПИТАЛЬНЫЕ ЗАТРАТЫ, TCO, СТОИМОСТЬ, РИСКИ.

Объект выпускной квалификационной работы: процесс выполнения лабораторных работ по физике в ХТИ – филиале СФУ.

Предмет: вопросы эмпирического характера, связанные с разработкой виртуальных лабораторных работ.

Целью выпускной квалификационной работы является: разработка виртуальных лабораторных работ по физике.

В современных условиях возникают ситуации, при которых выполнить лабораторные работы в аудитории института невозможно, такие, как например пандемия, когда институты были вынуждены отказаться от очного посещения на длительное время, заочные, дистанционные формы обучения или же, ситуации, когда студенты по личным уважительным причинам не могут посетить занятия (например, ограниченные возможности здоровья).

В первом разделе выпускной квалификационной работы описывается основная деятельность ХТИ – филиала СФУ. Проведен анализ процесса выполнения лабораторных работ в институте, построены диаграммы этого процесса в нотации IDEF0. Произведен обзор аналогичных программных продуктов, на основе которого обосновывается решение о необходимости разработки информационной системы «Виртуальные лабораторные работы по физике» и определяются требования к будущей системе. Принято решение о выборе трехуровневой клиент-серверной архитектуры. Описываются логическая и функциональная модель информационной системы с помощью диаграмм в нотациях IDEF0, IDEF3, DFD.

Во втором разделе поэтапно описывается процесс создания веб-приложения «Виртуальные лабораторные работы по физике», который сопровождается скриншотами кода программы и полученного интерфейса.

В третьем разделе выполнена оценка экономической эффективности проекта, выполнен анализ рисков, которые могут возникнуть в процессе разработки и эксплуатации веб-приложения, а также представлены возможные мероприятия по их снижению.

Разработанная система может быть использована в образовательных целях вузов с технической направленностью.

## SUMMARY

The theme of the graduation thesis is «Development of Virtual Lab Sessions in Physics» contains 142 pages of a text document, 12 tables, 154 figures, 15 reference items.

JAVA, SCRIPT, HTML, PHP, SERVER, MYSQL, 3D, USER, USER REGISTRATION, VIRTUAL LAB SESSIONS, THREEJS, SIMULATION, IDEF0, IDEF3, DFD, WEB APP, CAPITAL, TCO, COST, RISKS.

The object of the thesis: the process of performing laboratory classes in physics at KhTI - a branch of Siberian Federal University.

Subject: empirical issues related to the development of virtual laboratory classes.

The purpose of the graduation thesis: the development of virtual lab sessions in physics.

In modern world there appear situations when it is impossible to perform laboratory classes in the classrooms of the institute. This concerns the pandemic, when educational institutes were forced to refuse full-time attendance for a long time. It is also relevant for correspondence and distance training as well as for cases when students cannot attend classes for reasonable excuses (for example, the students with limited health capacities).

The first section of the thesis describes the general business of KhTI - a branch of SibFU. The analysis of the process of laboratory classes at the institute has been carried out; the diagrams of this process have been worked out in the notation IDEF0. A review of similar software products has been made, on the basis of which the decision on the need to develop an information system “Virtual laboratory classes in physics” has been substantiated and the requirements for the future system have been determined. It has been decided to choose a three-tier client-server architecture. The logical and functional model of the IT system has been described using diagrams in the notations IDEF0, IDEF3, DFD.

The second section describes the sequence of creating a web application “Virtual laboratory classes in physics”, which is accompanied by screenshots of the program code and the resulting interface.

The third section deals with the assessment of the economic efficiency of the project, the analysis of the risks that may arise during its development, the operation of a web application, and possible measures to reduce them.

The developed system can be applied at technical universities for educational training.

English language supervisor: \_\_\_\_\_

N.V. Chezybaeva

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Анализ процесса выполнения лабораторных работ по физике в ХТИ – филиале СФУ.....	6
1.1 Характеристика основной деятельности ХТИ – филиала СФУ.....	6
1.2 Функциональная модель процесса выполнения лабораторной работы AS-IS.....	9
1.3 Виртуальные лаборатории в образовательном процессе.....	13
1.4 Обзор программных средств выполнения виртуальных лабораторных работ.....	17
1.5 Выбор архитектуры и программных средств разработки.....	20
1.6 Описание процессов выполнения виртуальных лабораторных работы по физике методами структурного проектирования.....	24
1.6.1 Диаграмма вариантов использования (UseCase)	25
1.6.2 Модель организации процесса TO-BE	26
1.6.3 Диаграмма потоков данных (DFD)	27
1.6.4 Проектирование базы данных	29
1.6.5 Построение модели действий пользователя в информационной системе в нотации IDEF3	29
Выводы по разделу «Анализ процесса выполнения лабораторных работ по физике в ХТИ – филиале СФУ»	33
2 Описание разработки информационной системы «Виртуальные лабораторные работы по физике».....	34
2.1 Интерфейс системы.....	34
2.2 Программный код системы.....	44
2.2.1 База данных системы	44
2.2.2 Разметка и функционал веб-страниц	46

2.2.3	Создание имитационной модели физического процесса лабораторной работы «Изучение внутреннего трения жидкости»	59
2.2.4	Создание имитационной модели физического процесса лабораторной работы «Изучение явления дифракции света»	74
2.2.5	Создание имитационной модели физического процесса лабораторной работы «Изучение неупругого удара»	95
2.3	Описание последовательности действий пользователя при выполнении виртуальных лабораторных работ.....	110
	Выводы по разделу «Описание разработки информационной системы «Виртуальные лабораторные работы по физике».....	118
3	Оценка экономической эффективности проекта.....	119
3.1	Расчет затрат реализации проекта веб-приложения «Виртуальная лабораторная работа по физике».....	119
3.1.1	Затраты на заработную плату разработчиков	119
3.1.2	Затраты на средства вычислительной техники	121
3.1.3	Затраты на инструментальные программные средства	122
3.1.4	Эксплуатационные затраты	124
3.2	Расчет затрат реализации проекта методом TCO (TotalCostofOwership)	126
3.3	Определение экономической эффективности реализации проекта	128
3.3.1	Анализ рынка продуктов-аналогов. Установление стоимости программного продукта	128
3.3.2	Экономическая эффективность реализации проекта	130
3.4	Оценка рисков реализации IT-проекта.....	131
3.4.1	Оценка рисков проекта	132
3.4.2	Мероприятия по снижению вероятности рисков проекта	132
	Выводы по разделу «Оценка экономической эффективности проекта».....	134
	ЗАКЛЮЧЕНИЕ.....	135
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	136



## ВВЕДЕНИЕ

Для профессиональных компетенций учебные заведения, в том числе и ХТИ – филиал СФУ, включают в учебный план лабораторные занятия, на которых проводятся исследования каких-либо процессов в лабораториях с помощью различных установок, реактивов и т.д. Возникают ситуации, при которых выполнить лабораторную работу в аудитории института невозможно, например пандемия, когда институты были вынуждены отказаться от очного посещения на длительное время, заочные, дистанционные формы обучения или же, ситуации, когда студенты по личным уважительным причинам не могут посетить занятия (например, ограниченные возможности здоровья). Решить данную проблему можно при наличии виртуальной лаборатории, программного обеспечения, с помощью которого можно провести опыты без использования реального лабораторного оборудования и помещения, используя лишь компьютер, подключенный к сети Интернет. В настоящее время такое программное обеспечение имеет высокую актуальность, особенно, учитывая окружающую обстановку, когда большинство учебных заведений и организаций вынуждены были перейти на дистанционную форму обучения.

Объект выпускной квалификационной работы: процесс выполнения лабораторных работ по физике в ХТИ – филиале СФУ.

Предмет: вопросы эмпирического характера, связанные с разработкой виртуальных лабораторных работ.

Цель выпускной квалификационной работы: разработка виртуальных лабораторных работ по физике.

Для достижения цели необходимо решить следующие задачи:

1. охарактеризовать основную деятельность объекта исследования ХТИ – филиала СФУ;
2. выполнить анализ существующего процесса выполнения лабораторных работ в вузе;

3. проанализировать аналогичные программные средства для выполнения лабораторных работ;
4. осуществить выбор архитектуры ИС и инструментальных средств разработки ИС «Виртуальные лабораторные работы по физике»;
5. разработать ИС «Виртуальные лабораторные работы по физике»;
6. оценить экономическую эффективность разработки ИС «Виртуальные лабораторные работы по физике»;
7. оценить риски и способы их минимизации.

При выполнении выпускной квалификационной работы использованы методы исследования:

1. *теоретический*:
  - анализ основной деятельности филиала СФУ и процесс выполнения лабораторных работ по физике в институте;
  - предметной области для решения вопросов эмпирического характера, связанных с разработкой виртуальных лабораторных работ;
  - обоснование выбора конкретных средств разработки ИС;
  - функциональное моделирование бизнес-процесса проведения лабораторных работ;
  - сравнение аналогичных систем для выполнения лабораторных работ;
  - *эмпирический*: выбор инструментальных средств разработки;
  - практическая реализация разработки информационной системы для выполнения виртуальных лабораторных работ.

В данной работе представлены результаты теоретического анализа предметной области.

В проектной части представлено описание разработки трех виртуальных лабораторных работ.

В экономической части оценены капитальные затраты на создание информационной системы, эксплуатационные затраты, рассчитана совокупная

стоимость владения информационной системой, оценены риски и способы их минимизации.

# **1 Анализ процесса выполнения лабораторных работ по физике в ХТИ – филиале СФУ**

## **1.1 Характеристика основной деятельности ХТИ – филиала СФУ**

Хакасский технический институт – филиал федерального государственного автономного образовательного учреждения высшего образования «Сибирский федеральный университет» является обособленным подразделением федерального государственного автономного образовательного учреждения высшего образования «Сибирский федеральный университет» [1]. Институт был создан приказом Министерства высшего и среднего специального образования РСФСР в 1967 году и изначально именовался как общетехнический факультет Красноярского политехнического института в г. Абакане.

Миссией ХТИ – филиала СФУ является создание передовой образовательной, научно-исследовательской и инновационной инфраструктуры, продвижение новых знаний и технологий для решения задач социально-экономического развития Сибирского федерального округа, а также формирование кадрового потенциала — конкурентоспособных специалистов по приоритетным направлениям развития Сибири и Российской Федерации, соответствующих современным интеллектуальным требованиям и отвечающих мировым стандартам [2].

Институт осуществляет подготовку по очной, заочной и очно-заочной формам по направлениям специалитета, бакалавриата и магистратуры.

Ведётся подготовка специалистов и бакалавров в следующих областях: электроэнергетика; машиностроение; строительство; экономика; транспорт; информатика. Широко развиты программы дополнительного профессионального образования; повышение квалификации, профессиональная переподготовка, тематические семинары для руководителей и специалистов

предприятий и организаций.

Основными структурными подразделениями института являются:

- 5 кафедр;
- центр подготовки юного инженера;
- научно-образовательная лаборатория «Дендрэкология и экологический мониторинг»;
- отдел довузовской подготовки и нового набора;
- отдел информационных технологий;
- библиотека;
- центр студенческой культуры.

Имеется также 11 компьютерных классов (с выходом в сеть Интернет), медпункт, 2 спортивных зала, столовая, 2 студенческих общежития.

Институт располагает 2 учебными корпусами:

- корпус «А» - г. Абакан, ул. Щетинкина, 27;
- корпус «Б» - г. Абакан, ул. Комарова, 15.

Образовательные программы ХТИ – филиала СФУ направлены на обучение специалистов в сфере технической направленности.

Базовой дисциплиной для освоения всех специальных технических дисциплин является физика, которую студенты изучают на всех направлениях бакалавриата вуза, кроме экономических:

- 08.03.01 Строительство,
- 08.05.01 Строительство уникальных зданий и сооружений,
- 09.03.03 Прикладная информатика,
- 13.03.02 Электроэнергетика и электротехника,
- 15.03.05 Конструкторско-технологическое обеспечение машиностроительных производств,
- 23.03.03 Эксплуатация транспортно-технологических машин и комплексов.

Хакасский технический институт является филиалом очень крупного

Сибирского федерального университета, который применяет современные дистанционные технологии обучения посредством электронной информационно-образовательной среды. Взаимодействие студентов и преподавателей осуществляется через несколько сервисов, для обучения и размещения учебно-методических материалов используются eКурсы (<https://e.sfu-kras.ru>).

Задания можно выполнить в дистанционном формате и отправить на проверку преподавателю на eКурс дисциплины.

Однако это не относится к лабораторным работам по ряду дисциплин, в связи с тем, что необходимо применение лабораторного оборудования, в частности, для выполнения лабораторных работ по физике необходимо выполнить измерения в реальных лабораторных условиях.

Проблема выполнения лабораторных работ в вузе обострилась в период пандемии, когда вузы перешли на дистанционный формат занятий, посещение вуза оказалось невозможно.

Остро возникла потребность в замене реальных лабораторных работ на виртуальные, в проведении лабораторных работ в формате онлайн.

Кроме этого, в вузе осуществляется заочная и очно-заочная формы обучения, на которых обучаются:

- работающие граждане, в том числе вахтовым способом;
- молодые мамы, которым не с кем оставить ребенка;
- граждане с ограниченными возможностями.

Все перечисленное является обоснованием необходимости в разработке программного обеспечения, позволяющего провести лабораторные работы удаленно, вне лаборатории – виртуально.

В связи, с чем возникла идея разработки виртуальной лаборатории – программного обеспечения, с помощью которого можно провести опыты без использования реального лабораторного оборудования и помещения, используя лишь персональный компьютер.

В настоящее время такое программное обеспечение имеет высокую актуальность, особенно, учитывая окружающую обстановку, когда большинство вузов вынуждено переходить на дистанционный формат обучения.

## **1.2 Функциональная модель процесса выполнения лабораторной работы AS-IS**

На данный момент в институте выполнение лабораторных работ по физике реализуется в реальных лабораторных условиях. Преподаватель выдает задание студенту, в лаборатории проводит инструктаж по технике безопасности, выдает необходимые инструменты. Студент выполняет работу, делает необходимые замеры, записывает, анализирует. После выполнения лабораторной работы пишет отчет и сдает его преподавателю.

Функциональная модель процесса выполнения лабораторной работы AS-IS (как есть) спроектирована в нотации IDEF0. На рисунке 1 представлена контекстная диаграмма IDEF0 процесса «Выполнить лабораторную работу» AS-IS. Словарь стрелок диаграммы IDEF0 в таблице 1.

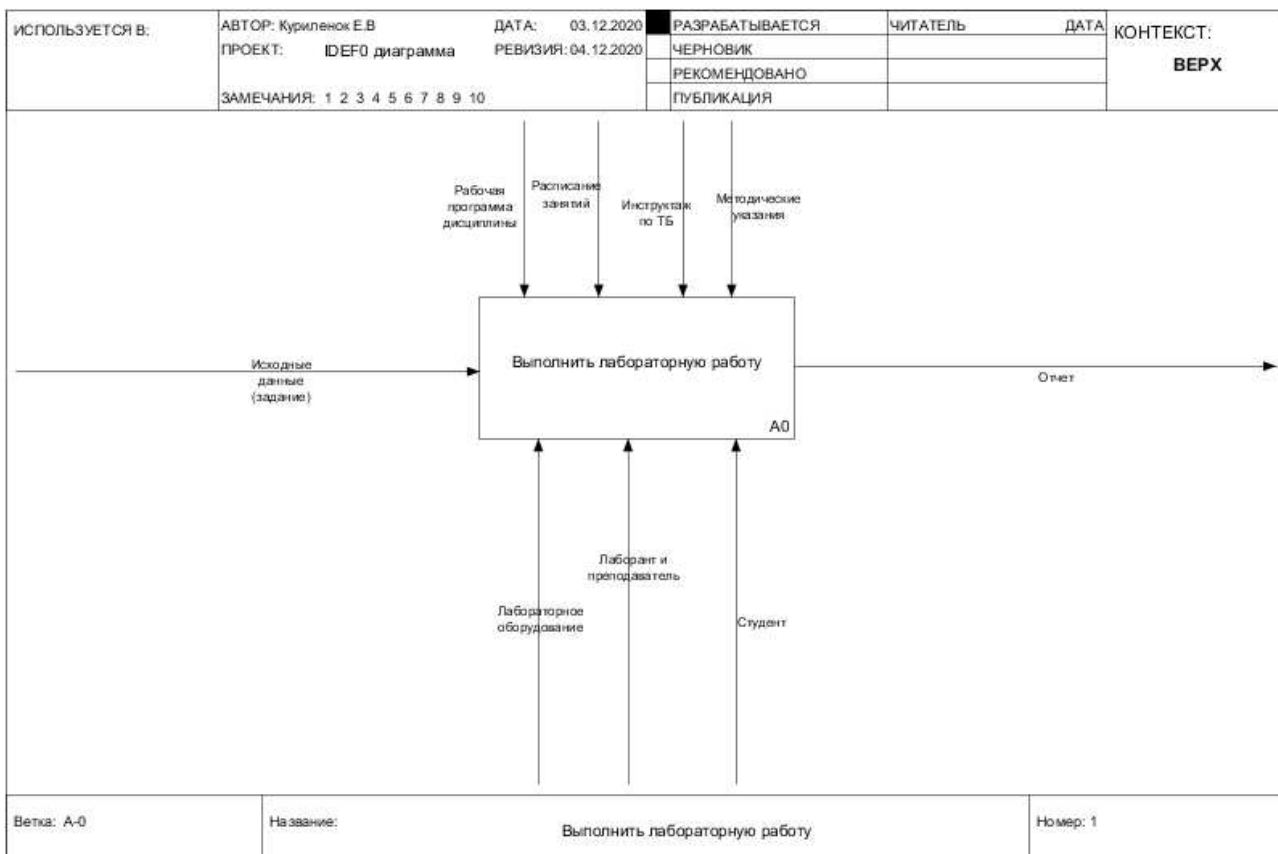


Рисунок 1 – Диаграмма AS-IS процесса выполнения лабораторной работы  
«Выполнить лабораторную работу»

Таблица 1 – Глоссарий контекстных стрелок блока A0

Поток данных	Наименование стрелки	Пояснение
Вход	Исходные данные (задание)	На основании управляющих документов студент получает информацию: дата и время занятия, вид занятия (лабораторное занятие), тема занятия, кто преподаватель и какие сроки выполнения отчета.
Управление	Рабочая программа дисциплины	Является частью образовательной программы, в которой указаны виды, темы занятий, формы отчетности и т.д.
	Расписание занятий	День и время, когда проводится занятие.
Механизм	Персонал образовательного учреждения	Готовит аудиторию, лабораторное оборудование.
	Студент	Проверяет оборудование, анализирует задание, выполняет работу, пишет отчет.
Выход	Отчет	Содержит полученные данные и выводы.

На рисунке 2 представлена декомпозиция блока A0 «Выполнить лабораторную работу».



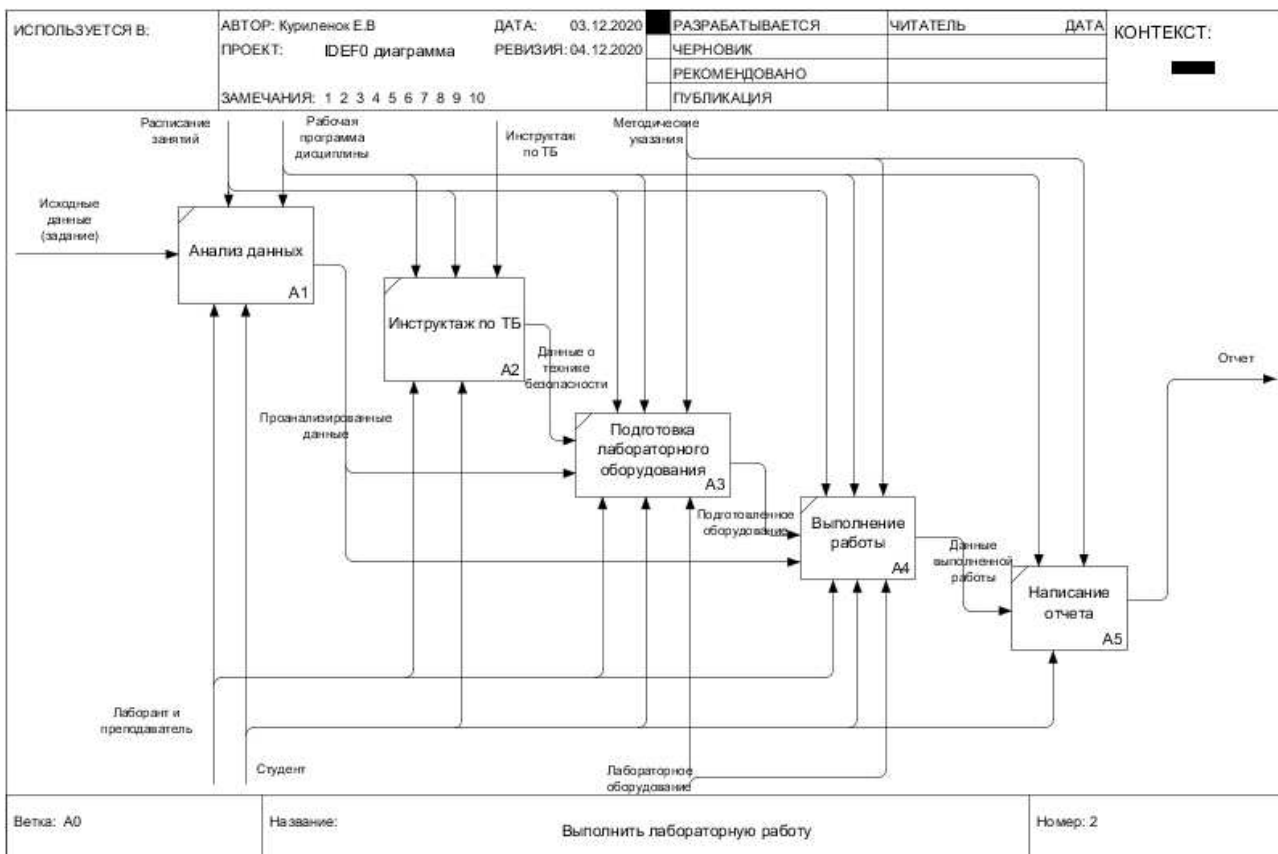


Рисунок 2 – Декомпозиция первого уровня блока A0 бизнес-процесса «Выполнить лабораторную работу»

Словарь стрелок декомпозиции первого уровня диаграммы IDEF0 представлен в таблице 2.

Таблица 2 – Глоссарий контекстных стрелок декомпозиции первого уровня блока A0

Номер блока	Поток данных	Наименование стрелки	Пояснение
A1	Вход	Исходные данные (задание)	Дата и время занятия, вид занятия (лабораторное занятие), тема занятия, кто преподаватель и какие сроки выполнения отчета.
		Управление	Рабочая программа дисциплины Расписание занятий
	Механизм	Лаборант и преподаватель	Готовит аудиторию, лабораторное оборудование.
		Студент	Узнает предварительную информацию,

			идет в аудиторию.
A1-A3	Выход	Проанализированные данные	Подготовленные данные для подготовки оборудования и выполнения лабораторной работы.
A1-A4			
A2	Управление	Рабочая программа дисциплины	Вид, тема занятий, форма отчетности и т.д.
		Расписание занятий	Специалисты УМО оповещают через сайт о расписании занятий
		Инструктаж ТБ	Преподаватель дает указания о технической безопасности работы в лаборатории.
	Механизм	Лаборант и преподаватель	Готовит аудиторию, лабораторное оборудование, журнал техники безопасности.
		Студент	Подтверждает соблюдение правил ТБ – росписью в журнале по ТБ
A2-A3	Выход	Данные о технике безопасности	Студент должен знать, как безопасно работать в аудитории в целом.
A3	Управление	Рабочая программа дисциплины	Тема занятий, форма отчетности, информация об оборудовании и т.д.
		Расписание занятий	Студент идет в указанное время в конкретную аудиторию.
		Методические указания	Методические указания как проводить измерения и с помощью каких приборов.
	Механизм	Лаборант и преподаватель	Готовит аудиторию, лабораторное оборудование.
		Студент	В аудитории с преподавателем и, следуя инструкциям по технике безопасности, подготавливает оборудование к выполнению работы.
		Лабораторное оборудование	В соответствии с темой занятия, подготавливается конкретное оборудование для выполнения лабораторной работы.

Продолжение таблицы 2

A3-A4	Выход	Подготовленное оборудование	Подготовленное, в соответствии с темой и техникой безопасности оборудование для выполнения работы
A4	Управление	Рабочая программа дисциплины	Тема занятий, форма отчетности, информация об оборудовании и т.д.
		Расписание занятий	Время занятия, место занятия
	Механизмы	Методические указания	Методические указания как проводить измерения и с помощью каких приборов, в соответствии с темой занятия.
		Лаборант и преподаватель	Контролирует выполнение работы студентом.
		Студент	В аудитории с преподавателем и, следуя инструкциям по технике безопасности, проводит

			опыты и выполняет измерения на подготовленном оборудовании, записывая результаты.
		Лабораторное оборудование	В соответствии с темой занятия, выполняется работа на оборудовании.
A4-A5	Выход	Данные выполненной работы	Полученные в процессе измерений данные.
A5	Управление	Рабочая программа дисциплины	Тема занятий, форма отчетности, информация об оборудовании и т.д.
		Методические указания	Методические указания как работать с данными, полученными в результате выполнения работы.
	Механизмы	Студент	Пишет отчет, придерживаясь методических указаний.
	Выход	Отчет	Написанный по всем стандартам отчет.

Процесс выполнения лабораторной работы начинается с блока А1 «Анализ данных», основная функция которого – организационная, связана с подготовкой к основному процессу (детальное описание стрелок представлено в таблице 2).

Далее отдельный блок А2, функция которого проведение техники безопасности перед выполнением лабораторной работы, указания, как безопасно подготовить оборудование, как на нем безопасно работать. После этого следует функция блока А3.

Функции блока А3 – подготовка лабораторного оборудования, в соответствии с методическими указаниями, указаниями преподавателя и техники безопасности.

Блок А4 – функция выполнения лабораторной работы, проведение измерений, фиксирование и анализ этих измерений, с помощью лабораторного оборудования, подготовка выводов.

Блока А5 – подготовка данных, полученных в процессе выполнения лабораторной работы А4, представленные в виде отчета.

Анализ модели IDEF0 бизнес-процесса «Выполнить лабораторную работу» и ее декомпозиции с детальным описанием контекстных стрелок позволяет на данном этапе сделать вывод, что в случае разработки, внедрения и применения программного обеспечения:

1. будут отсутствовать процессы, отображаемые блоком А3 «Подготовить лабораторное оборудование»;
2. изменится механизм блока А2 «Инструктаж по ТБ»;
3. изменится содержание и механизмы блока А4 «Выполнение измерений».

Из представленных функций будет подлежать автоматизации подготовка лабораторного оборудования, поскольку его заменит проектируемая информационная система.

Инструктаж по технике безопасности с росписью студента в журнале по ТБ при проведении лабораторных работ будет отсутствовать.

Все остальное останется без изменений, студент также будет получать задание, выполнять работу (только уже при помощи информационной системы), писать отчет и сдавать его преподавателю.

### **1.3 Виртуальные лаборатории в образовательном процессе**

*Виртуальная лаборатория* представляет собой программно-аппаратный комплекс, позволяющий проводить опыты без непосредственного контакта с реальной установкой или при полном отсутствии таковой.

Под виртуальными лабораториями понимается два типа программно-аппаратных комплексов:

- *лабораторная установка с удаленным доступом* – комплексные дистанционные лаборатории;
- программное обеспечение, позволяющее моделировать лабораторные опыты – *виртуальные лаборатории* (в узком смысле) [3].

В первом случае подразумевается так называемая лабораторная установка с удаленным доступом, в состав которой входит реальная лаборатория, программно-аппаратное обеспечение для управления установкой и оцифровки полученных данных, а также средства коммуникации. Во втором случае все

процессы моделируются при помощи компьютера.

В чём состоят преимущества виртуальных лабораторий перед реальными?

Виртуальные лабораторные работы – компьютерные программы, позволяющие выполнять эксперименты и получать результаты, без непосредственного использования реальных лабораторных установок и приборов. Для этого создается интерактивная модель лабораторной установки, включающая виртуальные приборы и инструменты. Компьютерная модель даёт возможность студенту последовательно выполнять этапы программы лабораторной работы. Работа с виртуальной лабораторной установкой напоминает своеобразную компьютерную игру. Можно нажимать кнопки виртуальных приборов, переключать тумблеры, поворачивать вентили, подключать провода, наблюдать за течением жидкости, горением и другими физическими и химическими явлениями и технологическими процессами.

На сайте Санкт-Петербургского (<http://distolymp2.spbu.ru/www/virtlab3/>) университета есть пример демонстрационной версии виртуальной лаборатории (рисунок 3).

Лабораторные работы этой лаборатории имеют ряд своих преимуществ, например, для их работы не нужен интернет, достаточно скачать лабораторию один раз, имеется наглядность, не нужен мощный компьютер.

К недостаткам можно отнести платность лаборатории, для полноценной работы с ней, нужно приобрести лицензионную версию программы.

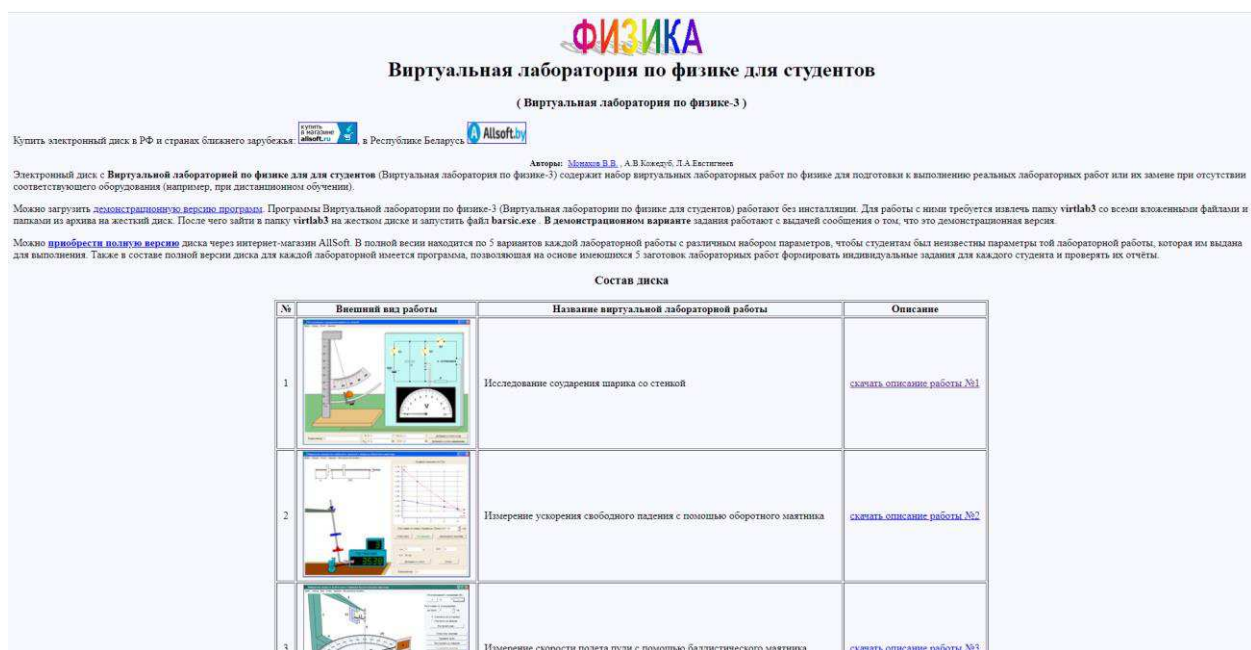


Рисунок 3 – Пример виртуальной лаборатории

Еще один сервис, предоставляющий возможность выполнить лабораторные работы по физике в режиме онлайн – это «Mediadidaktika» (<http://mediadidaktika.ru/>), на котором размещены виртуальные лабораторные работы по физике из сферы «механика», «термодинамика» и т.д.

Одним из преимуществ данного сервиса является то, что для его работы достаточно только браузера. Так же, к преимуществам можно отнести наглядность, оптимизированность.

Одним из важнейших недостатков сервиса можно считать то, что он работает на устаревшем плагине Flash, который большинство браузеров перестают поддерживать уже в этом году. К недостаткам так же можно отнести то, что не все работы функционируют, много работ представлено в виде изображений или видеороликов.

Кроме этих примеров есть еще несколько виртуальных лабораторий, в которых можно работать только после приобретения лицензии.

Основными преимуществами виртуальных лабораторий (А. В. Трухин, 2002 г.) являются:

1. Отсутствие необходимости приобретения дорогостоящего

оборудования и реактивов. Конечно, компьютерное оборудование и программное обеспечение также стоит недешево, однако универсальность компьютерной техники и ее широкая распространенность компенсируют этот недостаток.

2. Возможность моделирования процессов, протекание которых принципиально невозможно в лабораторных условиях. Современные компьютерные технологии позволят пронаблюдать процессы, трудноразличимые в реальных условиях без применения дополнительной техники, например, из-за малых размеров наблюдаемых частиц.

3. Возможность проникновения в тонкости процессов и наблюдения происходящего в другом масштабе времени, что актуально для процессов, протекающих за доли секунды или, напротив, длящихся в течение нескольких лет.

4. Безопасность. Безопасность является немаловажным плюсом использования виртуальных лабораторий в случаях, где идет работа, например, с высокими напряжениями или химическими веществами.

5. Возможность быстрого проведения серии опытов с различными значениями входных параметров, что часто необходимо для определения зависимостей выходных параметров от входных.

6. Экономия времени и ресурсов для ввода результатов в электронный формат. Данные могут заноситься в электронную таблицу результатов непосредственно при выполнении опытов экспериментатором или автоматически. Таким образом, экономится время и значительно уменьшается процент возможных ошибок.

7. Важное преимущество - использование виртуальной лаборатории в дистанционном обучении, когда в принципе отсутствует возможность работы в лабораториях вуза [3].

Перечисленные преимущества виртуальных лабораторных работ являются основой требований заказчика ХТИ – филиала СФУ:

1. *виртуальность* (лаборатория должна быть виртуальной и для выполнения опытов нужен будет только персональный компьютер) – данный принцип требует разработки программного обеспечения;

2. *оптимизированность* (лабораторные работы должны быть доступны вне вуза для любого студента) – соответствие данному принципу требует разработки веб-приложения, для действия которого не требуется специальное программное обеспечение, только браузер и наличие интернета;

3. *наглядность*. Информационная система должна полностью и наглядно отражать физический процесс лабораторной работы, которую выполняет студент.

#### **1.4 Обзор программных средств выполнения виртуальных лабораторных работ**

Прежде чем создавать полностью новое программное обеспечение виртуальных лабораторных работ, необходимо изучить существующие аналогичные программные продукты. Одна из статей информационной площадки [vs.ru](http://vs.ru) представляет 8 сервисов с виртуальными лабораторными работами, в том числе, по физике [4]. Далее рассмотрены некоторые из этих сервисов.

Сервис «Профессиональная группа» предоставляет пользователям набор виртуальных лабораторных работ из различных областей [5]. Страница сервиса показана на рисунке 4.



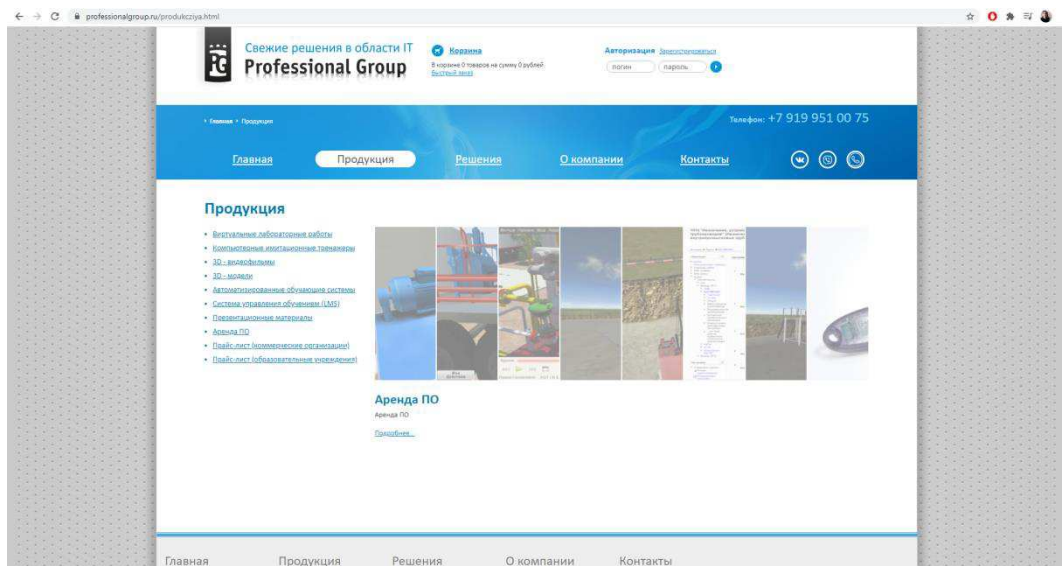


Рисунок 4 – Страница сервиса «Профессиональная группа»

Лабораторные работы по физике входят в список их работ, на сайте присутствует раздел выбора тем лабораторных работ (рисунок 5).

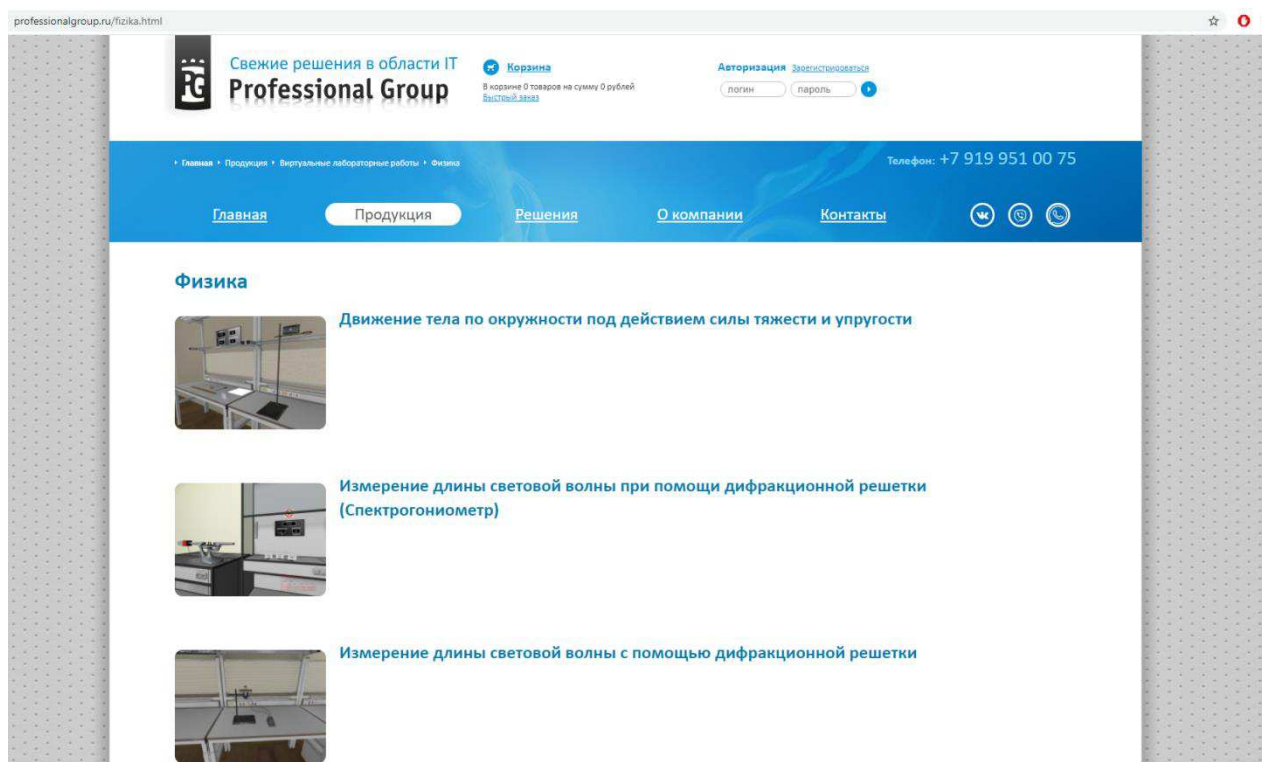


Рисунок 5 – Скриншот страницы выбора лабораторных работ сервиса «Профессиональная группа»

Стоимость одной лицензии на одну лабораторную работу 1680 рублей, демонстрационной версии программы сервисом не предоставляется.

Другой сервис, предоставляющий широкие возможности для дистанционного обучения это «ProgramLab» [6].

Интерфейс страницы сайта на рисунке 6.

Сервис предлагает более 80 виртуальных лабораторий (химия, физика, пищевая промышленность, нефть и газ, ветеринарное дело, медицина, энергетика и теплоснабжение, металлургия, строительство и т.д.).

Компания также занимается разработкой «цифровых двойников» – виртуальными копиями реальных объектов, которые выглядят и функционируют так же, как и их реальные двойники или прототипы.

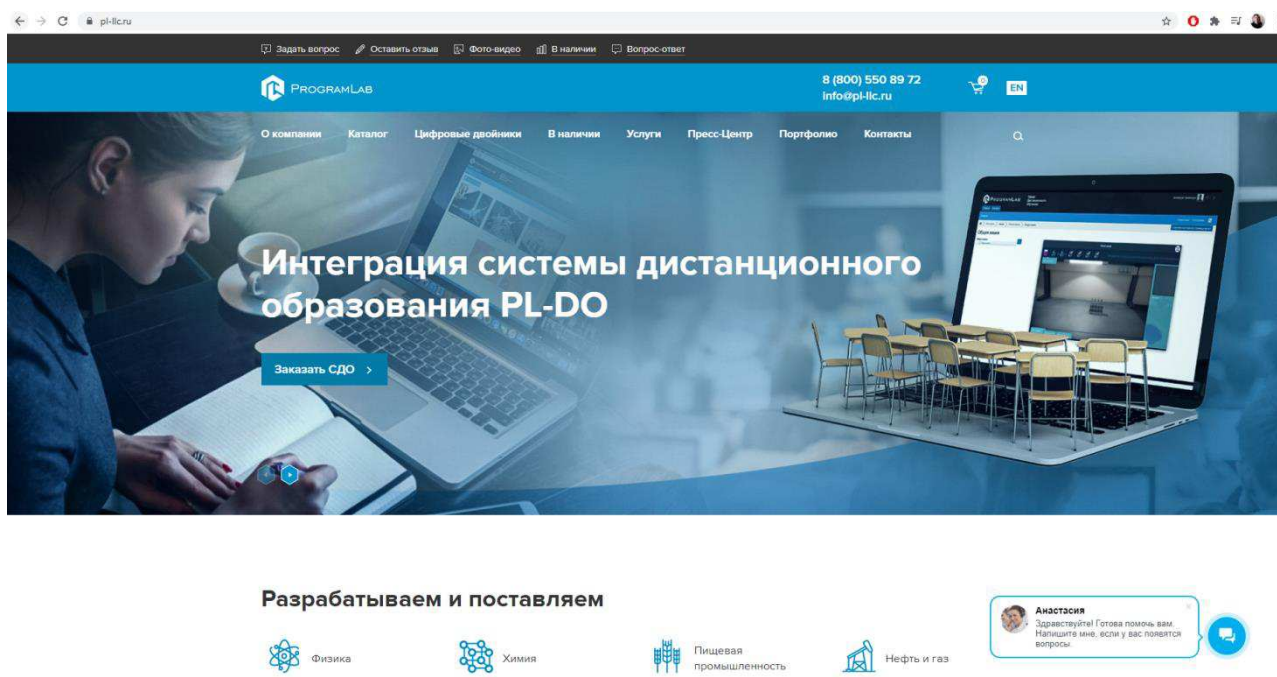


Рисунок 6 – Главная страница сервиса «ProgramLab»

Во вкладке «Физика» сервиса ProgramLab имеется возможность выбрать раздел физики, после чего выбирается тема лабораторной работы (рисунок 7). После выбора лабораторной работы, пользователю открывается страница покупки программного обеспечения с возможностью приобрести дополнительно персональный компьютер для этого ПО (рисунок 8).

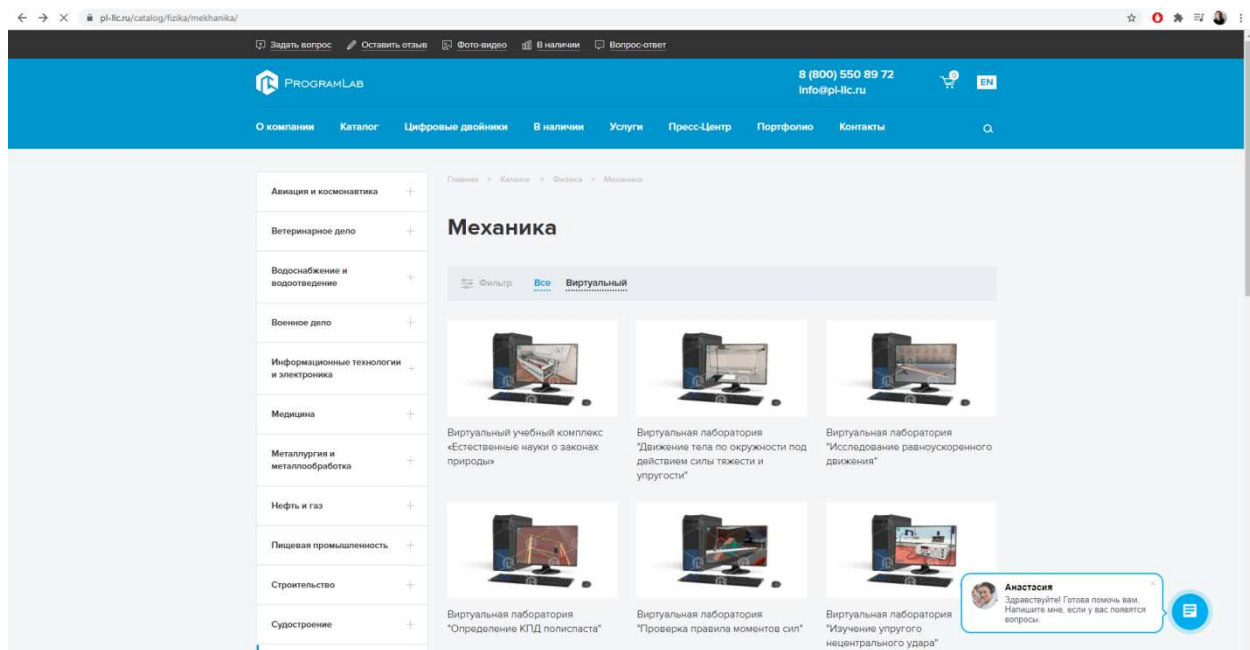


Рисунок 7 – Страница выбора лабораторных работ сервиса ProgramLab

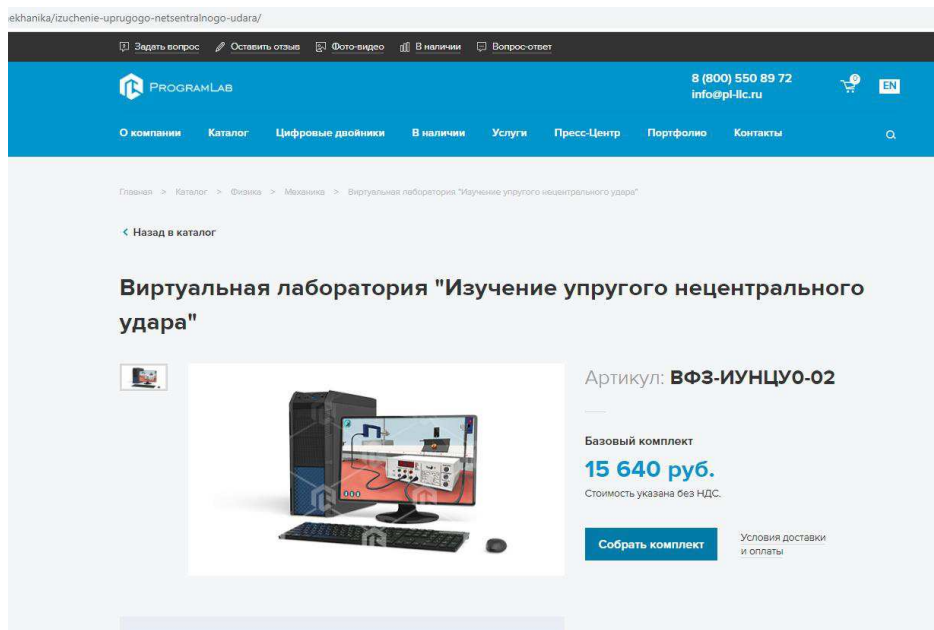


Рисунок 8 – Страница покупки программного обеспечения сервиса ProgramLab

Минимальное количество лицензий для приобретения – 10, их стоимость составляет 15 640 рублей. Версии программы для демонстрации возможностей не предоставляется.

Проанализировав данные сервисы, было принято решение о разработке своего программного обеспечения. Продемонстрированные лабораторные работы имеют ряд своих преимуществ, в том числе они сделаны в виде

виртуальных лабораторий, смоделированных с помощью 3D графики. Однако недостатком таких сервисов является отсутствие многих тем лабораторных работ, а также стоимость приобретения лицензий.

### **1.5 Выбор архитектуры и программных средств разработки**

Информационная система «Виртуальные лабораторные работы по физике» предполагает выполнение студентом работ на персональном компьютере, как в оборудованных компьютерных классах, так и на домашнем компьютере, причем информационная система будет содержать элементы 3D графики. Данные требования определяют выбор архитектуры и инструментальных средств разработки.

Архитектура информационной системы – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы. Выделяют типовые архитектуры ИС: файл-серверная архитектура; клиент-серверная архитектура; трехуровневая клиент-серверная архитектура.

Информационная система «Виртуальные лабораторные работы по физике» предполагает наличие базы данных, в которой будут храниться данные о студенте и лабораторных работах, исходя из этого, у создаваемой информационной системы будет трехуровневая клиент-серверная архитектура.

Трехуровневая клиент-серверная архитектура или так называемая 3-х слойная архитектура клиент-сервер. Основным ее отличием от других архитектур является физическое разделение программ, отвечающих за хранение данных (СУБД) от программ эти данные обрабатывающих (сервер приложения (СП), application server (AS)). Такое разделение программных компонент позволяет оптимизировать нагрузки как на сетевое, так и на вычислительное оборудование комплекса [7]. На рисунке 9 представлена модель этой архитектуры.



Рисунок 9 – Модель трехуровневой клиент-серверной архитектуры

Веб-приложение – клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами [8].

В серверной части приложения размещена база данных, в которой будет храниться информация о студенте, лабораторных работах и параметрах, необходимых для имитации физических процессов.

Программные средства, которые будут использованы для разработки системы:

1. SublimeText – для работы с кодом приложения.
2. OsPanel – для имитации сервера при разработке и тестировании элементов приложения.

3. PHPMyAdmin – для создания и работы с базой данных веб-приложения.

4. Браузер GoogleChrome – для разработки, тестирования и отладки веб-приложения.

5. Операционная система Windows 10.

SublimeText — текстовый редактор, поддерживает большое количество языков программирования и имеет возможность подсветки синтаксиса для C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL и XML. В дополнение к тем языкам программирования, которые включены по умолчанию, пользователи имеют возможность загружать плагины для поддержки других языков [8].

OSpanel (Open Server) – пакет серверного программного обеспечения для организации собственного портативного веб сервера для разработки сайтов. Помимо основного серверного ПО, сборка содержит в себе еще и дополнительные утилиты (Notepad++, Process Explorer, Sublime Text 2).

PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд [8].

Кроме перечисленных инструментов, при разработке системы использован язык программирования JavaScript, с использованием технологии WebGL для демонстрации элементов 3D графики.

WebGL (Web Graphics Library) программная библиотека для языка JavaScript, предназначенная для визуализации интерактивной трёхмерной графики и двухмерной графики в пределах совместимости веб-браузера без

использования плагинов. WebGL приносит в веб трёхмерную графику, вводя API, который построен на основе OpenGL ES 2.0, что позволяет его использовать в элементах canvas HTML5 [9].

Построение графики происходит на объекте canvas. Отрисовка в WebGL осуществляется с помощью так называемых вершинных и пиксельных шейдеров. Шейдеры – функции, написанные на специальном языке программирования GLSL, которые выполняются графической картой и обрабатывают данные вершин и пикселей, определяя окончательные параметры изображения объекта. Они могут включать в себя описание поглощения и рассеяния света, наложение текстуры, отражение и преломление, затенение, и т.д. Работа с шейдерами – трудоемкий процесс, нужно описать каждую вершину, ребро, грань, нормали к поверхности, их цвет, положение и пр. [10].

Для того чтобы не писать много кода и шейдерных процедур, существуют фреймворки и библиотеки, облегчающие работу с созданием интерактивной трехмерной графики в браузере и заметно сокращают программный код. Один из таких инструментов – библиотека Three.js.

Three.js – библиотека JavaScript, содержащая набор готовых классов для создания и отображения интерактивной компьютерной 3D графики в WebGL.

Библиотека Three.js облегчает работу с WebGL. При использовании Three.js отпадает необходимость в написании шейдерных процедур (но эта возможность остается), и появляется возможность оперировать с более привычными понятиями сцены, света и камеры, объектами и их материалами.

Библиотека Three.js также поддерживает отображение готовых трёхмерных моделей формата Collada (который обеспечивает совместимость моделей таких программ, как 3Ds Max, Blender, и т.д.).

Над библиотекой Three.js работает большое количество разработчиков. Главным идеологом и разработчиком является программист из Барселоны Рикардо Кабельо (Ricardo Cabello), творческий псевдоним Mr. Doob.

Для использования библиотеки нужно скачать с сайта [threejs.org](http://threejs.org) файл

three.min.js2 (собственно, сама библиотека, содержащая минимальный набор функций для работы с WebGL), и подключить к проекту.

Таким образом, архитектура информационной системы реализована в виде веб-приложения, которое предполагает меню выбора лабораторных работ, а также графическое отображения выполнения физических процессов с использованием элементов 3D-моделирования.

В качестве оборудования для работы системы будет являться персональный компьютер средней мощности с доступом в сеть Интернет.

## **1.6 Описание процессов выполнения виртуальных лабораторных работы по физике методами структурного проектирования**

В структурном анализе и проектировании используются модели, описывающие:

1. функциональную структуру системы;
2. последовательность выполняемых действий;
3. передачу информации между функциональными процессами;
4. отношения между данными.

Наиболее распространенными моделями первых трех групп являются модели в нотациях IDEF0, IDEF3, UseCase, DFD.

Для проектирования информационной системы нужно четко определить функции, которые будет выполнять эта система, и построить диаграммы, которые будут показывать, что будет в системе, как она будет работать (IDEF0), кто ей будет пользоваться (UseCase) и как (IDEF3). Так же, диаграмму, показывающую потоки данных системы (DFD).

### **1.6.1 Диаграмма вариантов использования (UseCase)**

Чтобы создать информационную систему, прежде всего, нужно



определить сущности, кто ей будет пользоваться и как. Разрабатываемой системой будут пользоваться студент и преподаватель. Диаграмма UseCase (рисунок 10) показывает, как они ею будут пользоваться.

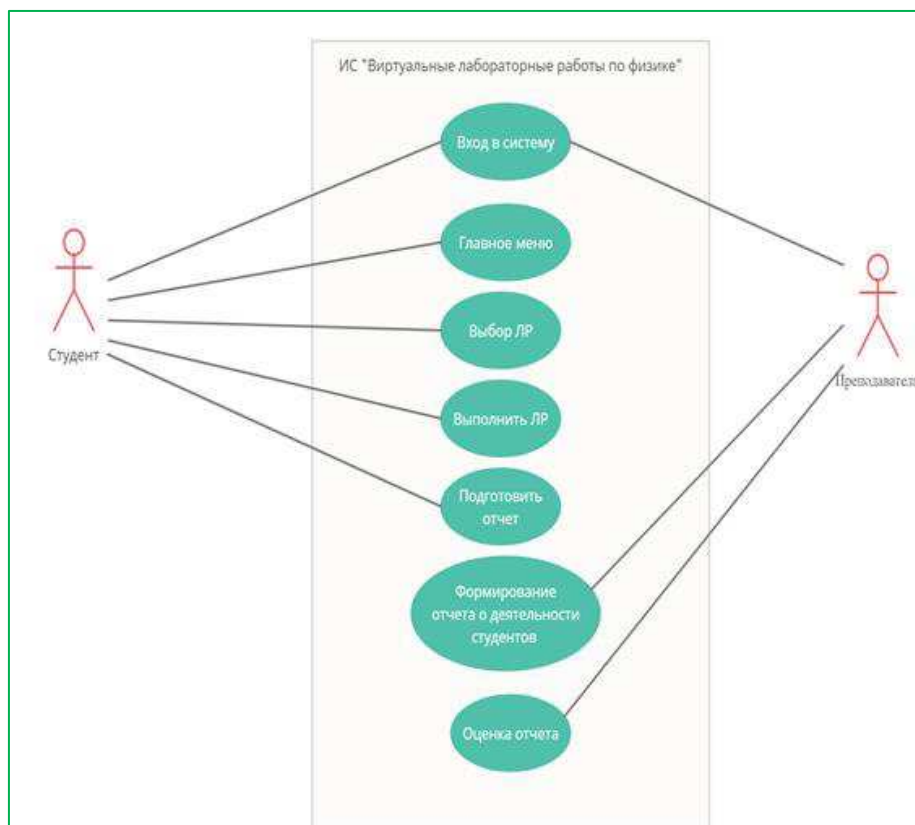


Рисунок 10 – Диаграмма UseCase ИС «Виртуальные лабораторные работы»

### 1.6.2 Модель организации процесса ТО-ВЕ

После внедрения информационной системы выполнять лабораторные работы студент будет с помощью компьютера.

Процесс выполнения останется прежним, студент также будет получать задание от преподавателя, анализировать его, выполнять измерения, писать отчет. Отличаться процесс будет тем, что лабораторное оборудование и лабораторию в целом заменит информационная система и все измерения будут выполняться на компьютере.

На рисунке 11 показана функциональная диаграмма в нотации IDEF0, показывающая, как будет производиться процесс выполнения лабораторной

работы после внедрения информационной системы «Виртуальные лабораторные работы по физике».

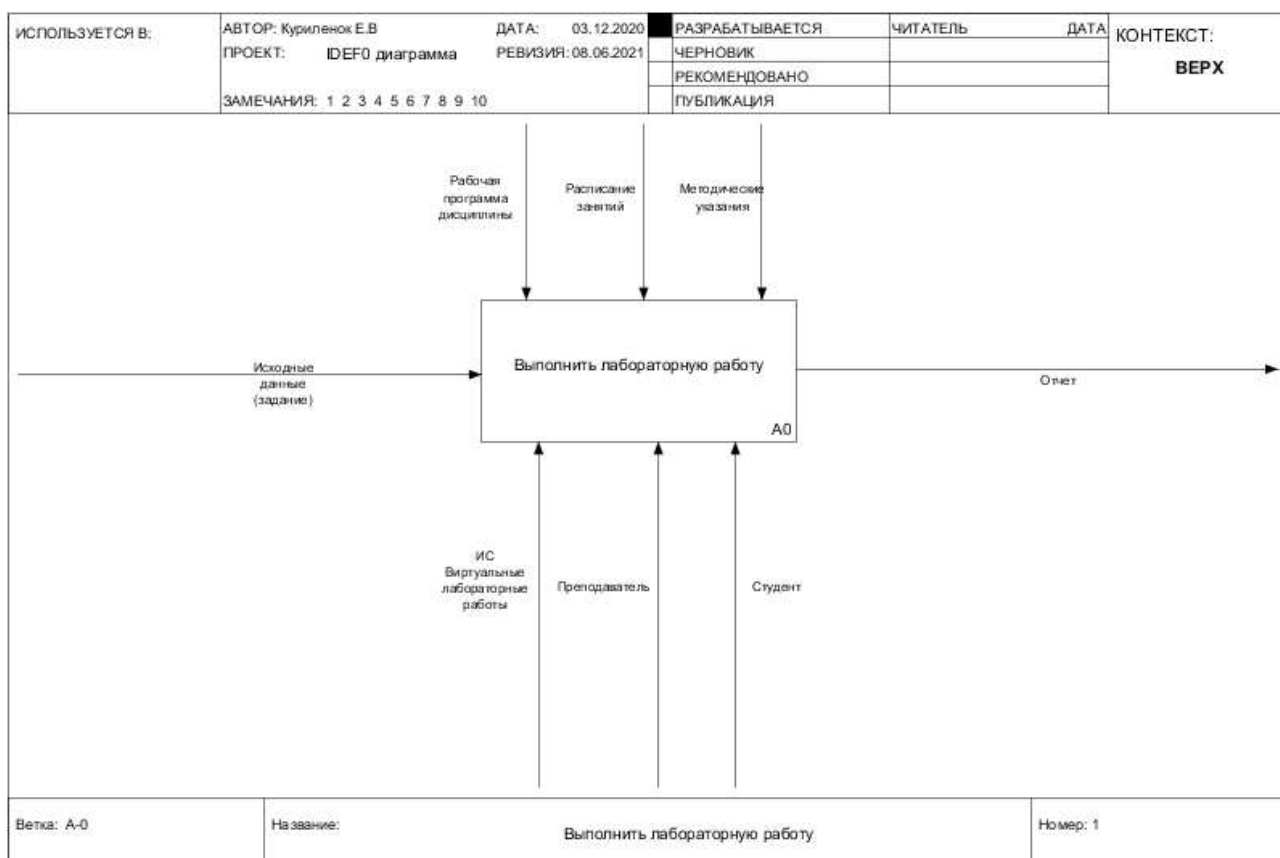


Рисунок 11 – Диаграмма TO-VE процесса «Выполнить лабораторную работу»

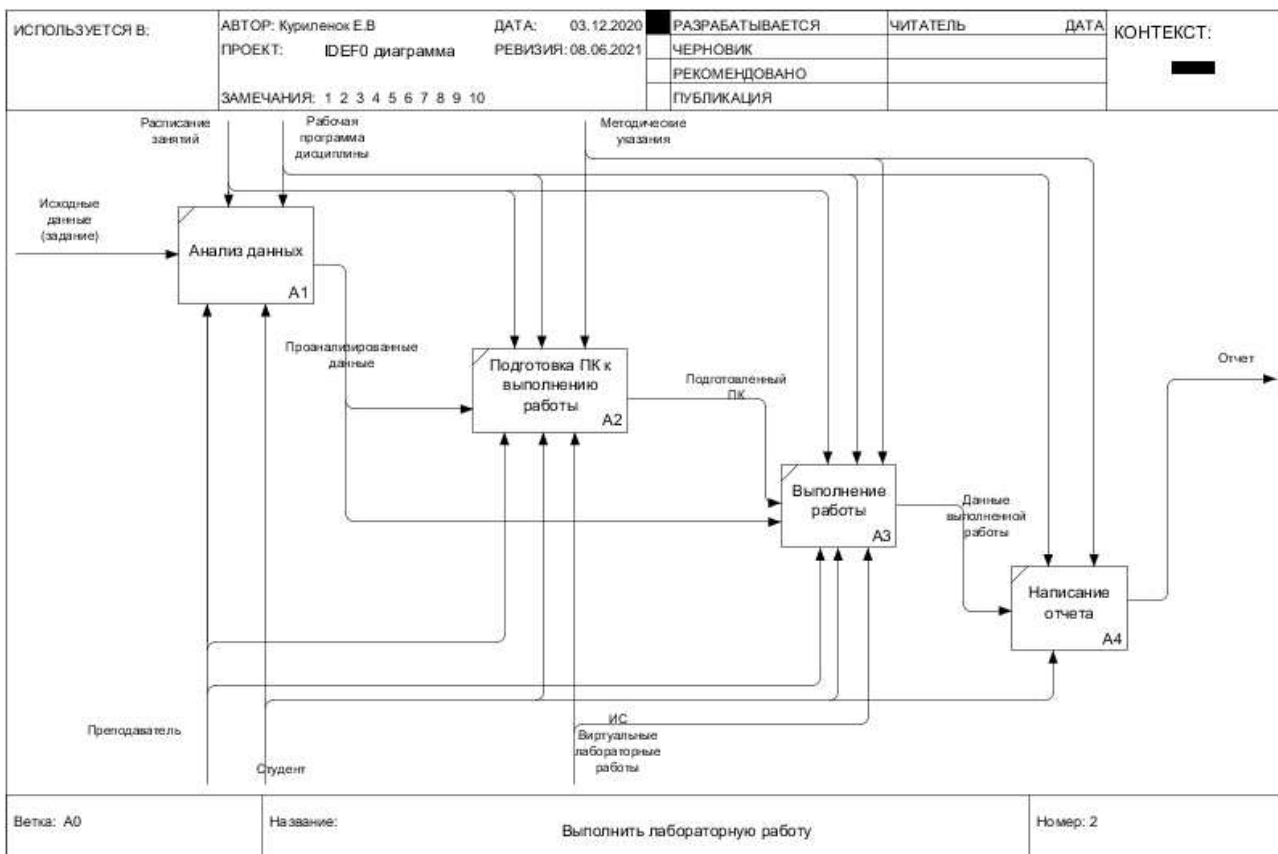


Рисунок 12 – Декомпозиция первого уровня диаграммы TO-VE процесса «Выполнить лабораторную работу»

Изменения после внедрения информационной системы «Виртуальные лабораторные работы по физике» в образовательный процесс института:

- подготовка лабораторного оборудования в реальной лаборатории заменяется подготовкой компьютера к выполнению лабораторной работы (студент должен открыть eКурсы, по ссылке запустить веб-приложение);
- выполнение работы остается, но студент производит измерения на виртуальном лабораторном оборудовании.

### 1.6.3 Диаграмма потоков данных (DFD)

Для правильного функционирования информационной системы нужно учесть, как будет эта система работать с информацией, поступающей в нее, а также, как в целом и какая информация будет использоваться для работы

информационной системы. На рисунке 13 показана диаграмма в нотации DFD, которая отображает потоки данных, необходимые для работы информационной системы. На рисунке 14 показана декомпозиция этой диаграммы.

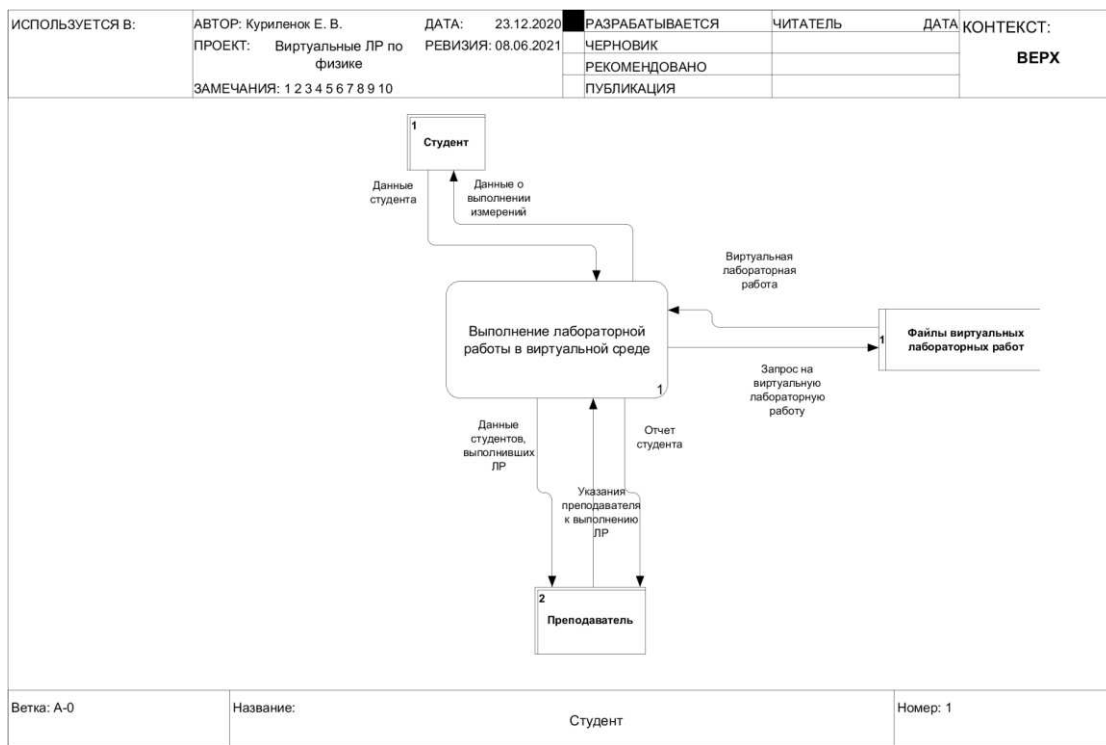


Рисунок 13 – Диаграмма DFD

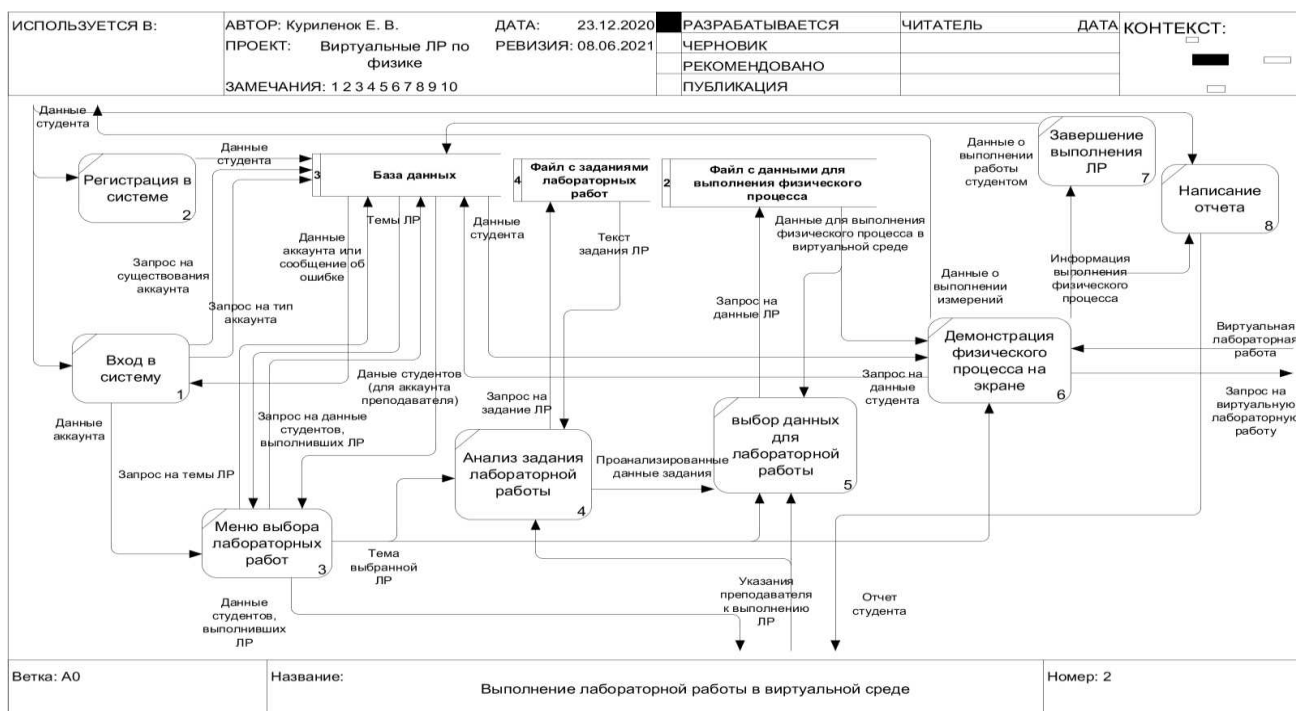


Рисунок 14 – Декомпозиция диаграммы DFD

### 1.6.4 Проектирование базы данных

Информационная система «Виртуальные лабораторные работы по физике» предполагает некоторое хранилище данных о лабораторной работе, о студенте, о физическом процессе, которые будут показываться на экране в процессе выполнения определенной работы.

Для реализации этого хранилища будет создана база данных, ER-диаграмма которой показана на рисунке 15, где таблицы обозначены прямоугольниками, в строках которых записаны имена полей. Связи между таблицами показаны стрелками, где выход-вход стрелки показывает связь «один ко многим» соответственно.

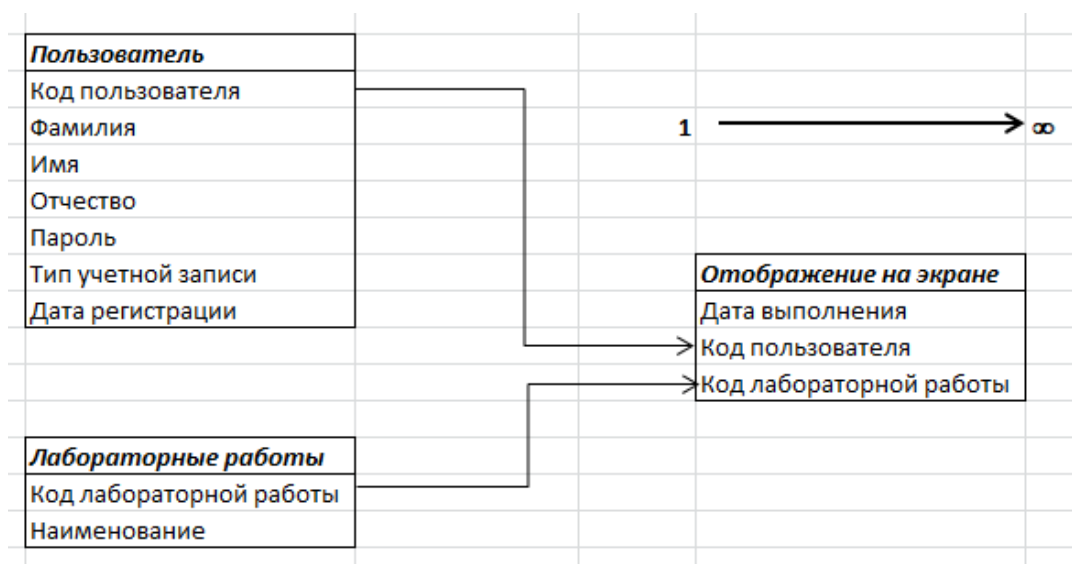


Рисунок 15 – ER-диаграмма


### 1.6.5 Построение модели действий пользователя в информационной системе в нотации IDEF3

Веб-приложение предполагает две роли пользователя:

1. Студент.
2. Преподаватель.

Начало работы пользователей не зависит от их роли, им нужно пройти по ссылке на веб-приложение. Студент или преподаватель проходит по ссылке в eКурсе дисциплины «Физика» на веб-приложение «Виртуальные лабораторные работы по физике».

Далее, перед пользователем открывается форма входа, чтобы войти или зарегистрироваться, на которой он вводит свои данные (ФИО, номер зачетной книжки). Макет формы представлен на рисунке 16.



Виртуальные лабораторные работы по физике

Введите свои данные:

Фамилия

Имя

Отчество

Номер зачетной книжки

Перейти на форму входа      Зарегистрироваться

Рисунок 16 – Макет формы регистрации

Далее для роли «Студент»: после успешной авторизации перед ним открывается меню с перечнем лабораторных работ (рисунок 17). Студент выбирает лабораторную работу, которую ему нужно выполнить. На данном этапе планируется разработать три лабораторные работы.

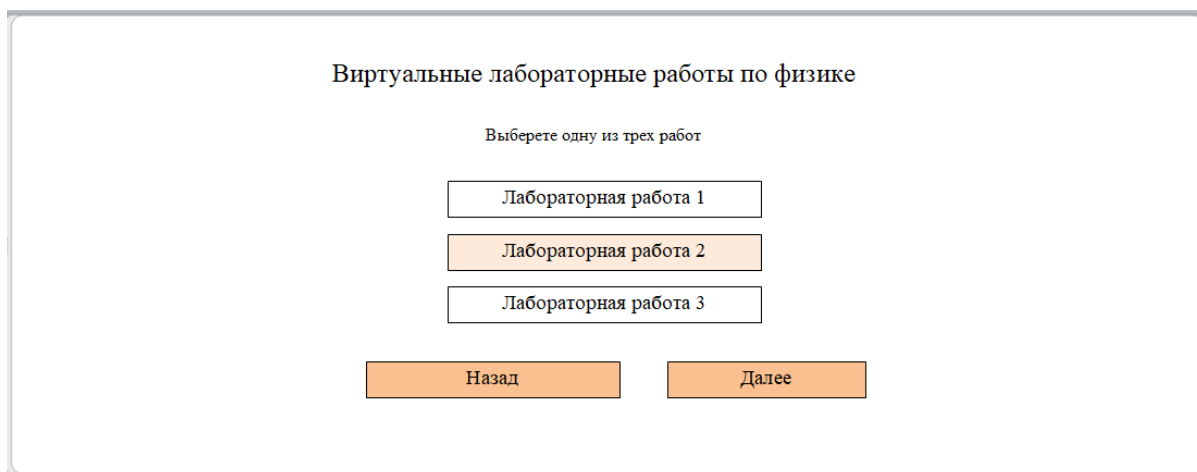


Рисунок 17 – Макет меню выбора лабораторных работ

После выбора лабораторной работы перед студентом открывается страница с пояснениями об используемом оборудовании краткими теоретическими сведениями, порядке выполнении лабораторной работы (макет представлен на рисунке 18).

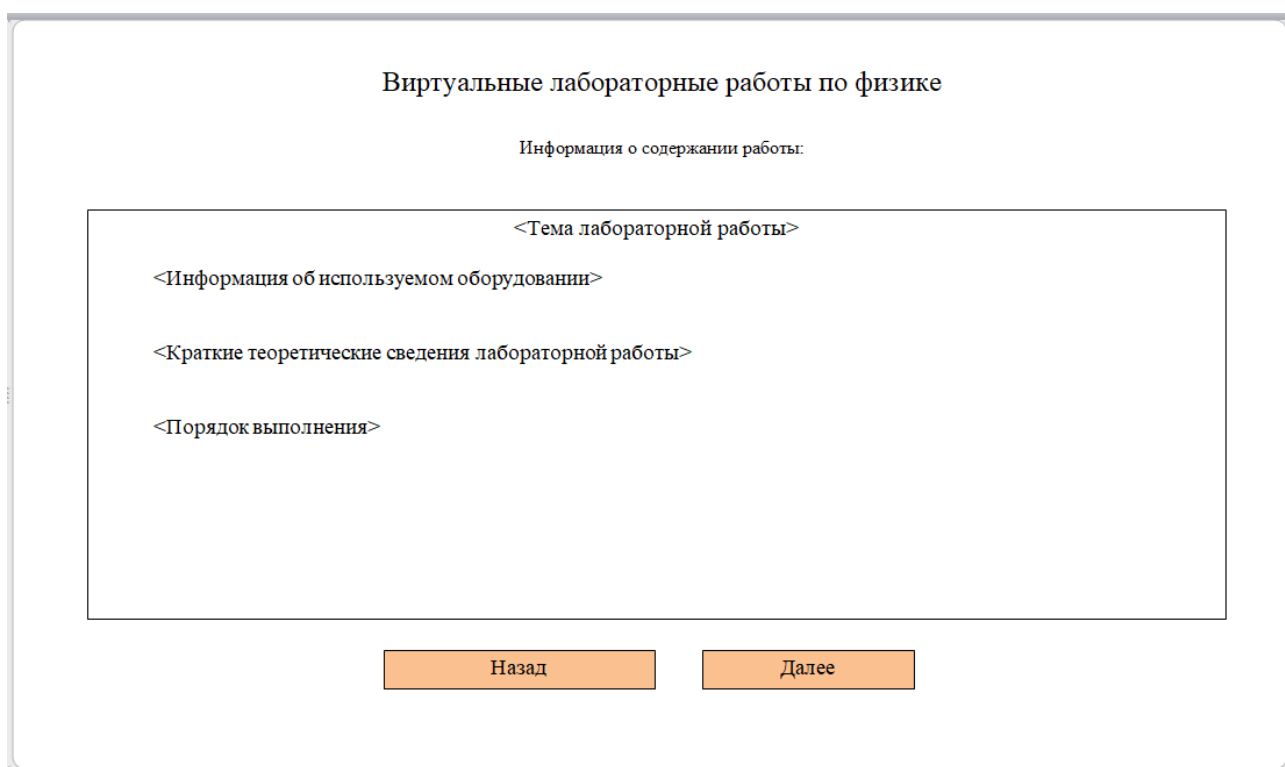


Рисунок 18 – Макет формы текста лабораторной работы

Далее осуществляется переход в графический интерфейс с виртуальным оборудованием и полями ввода необходимых данных.

Студент выбирает данные и ему демонстрируется выполнение физического процесса.

Для роли «Преподаватель» отличия в интерфейсе:

- после входа под своей учетной записью преподаватель будет видеть список студентов, выполнивших лабораторную работу (макет на рисунке 19),
- под которым будет кнопка для перехода в меню выбора лабораторных работ.

Виртуальные лабораторные работы по физике

Информация о студентах, выполнивших работы:

№	Фамилия	Имя	Отчество	Номер зачетной книжки	Дата выполнения	Тема лабораторной работы
1						
2						
3						
..						
n						

Рисунок 19 – Макет формы после входа преподавателя

На рисунке 20 представлен процесс работы с приложением в виде диаграммы в нотации IDEF3.



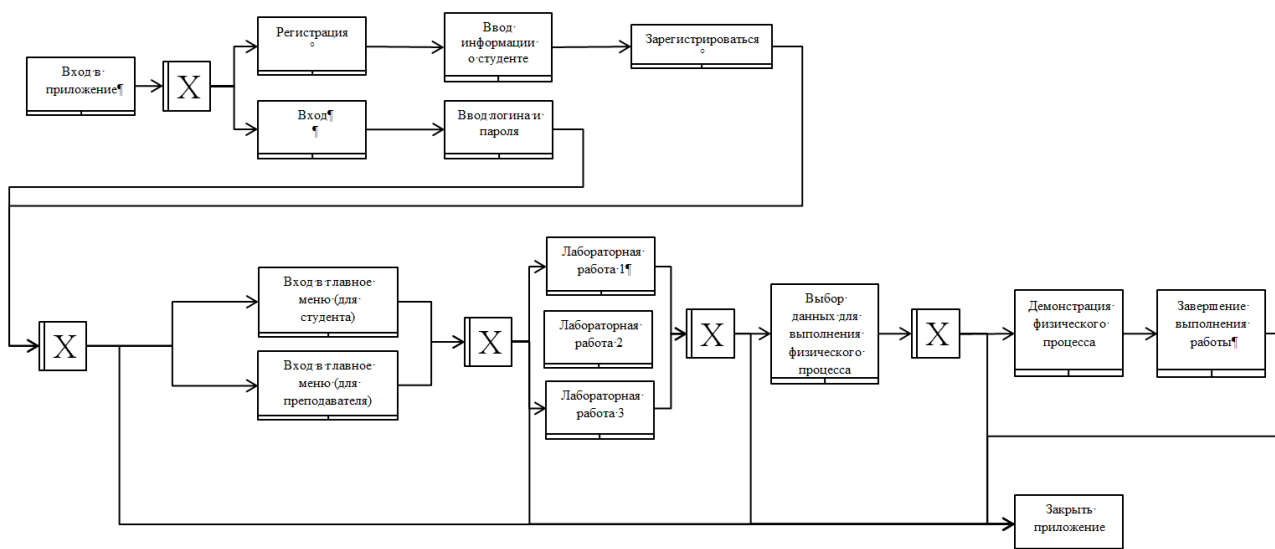


Рисунок 20 – Диаграмма в нотации IDEF3

### Выводы по разделу «Анализ процесса выполнения лабораторных работ по физике в ХТИ – филиале СФУ»

В первой части выпускной квалификационной работы представлены результаты исследования предметной области процесса выполнения лабораторных работ по физике студентами как есть (AS-IS) и как будет (TO-BE).

Дана краткая характеристика основной деятельности предприятия – образовательного учреждения, в котором возникла необходимость создания и применения в образовательном процессе системы для выполнения лабораторных работ удаленно, вне реальной лаборатории.

Проанализирован существующий процесс выполнения лабораторных работ (построена модель организации процесса AS-IS (как есть)) и описан с помощью диаграммы в нотации IDEF0.

Выполнен обзор аналогичных программных средств, для получения представления о том, как реализуются физические процессы программными средствами. Определены общие требования к системе.

Проанализированы архитектуры информационных систем и выбрана трехуровневая клиент-серверная архитектура, соответствующая требованиям виртуальности, оптимизированности, наглядности.

Представлено проектирование разработки информационной системы на языке моделирования, в частности построены диаграммы:

- функционального моделирования IDEF0, и ее декомпозиция для описания процессов верхнего уровня (как будет работать система);
- диаграмма взаимодействия UseCase (кто ею будет пользоваться);
- диаграмма IDEF3, для отображения последовательности выполнения работ (как ею пользоваться);
- диаграмма, описывающая потоки данных (информации) (DFD).

Построение диаграмм будущей информационной системы необходимо для ее разработки.

## **2 Описание разработки информационной системы «Виртуальные лабораторные работы по физике»**

Процесс выполнения лабораторной работы предполагает взаимодействие студента и преподавателя посредством электронной информационно-образовательной среды СФУ. Студент или преподаватель будут проходить по ссылке в eКурсе дисциплины «Физика» на веб-приложение «Виртуальные лабораторные работы по физике» (рисунок 21).

## Физика (часть 1)

Обзорная панель > Курсы > Каталог курсов > Филиалы > Хакасский технический институт > Кафедра прикладной информатики, математики и естественнонаучных дисциплин > Физика (часть 1 - механика)  
 Модуль 1. Механика > Лабораторная работа. Изучение неупругого удара

### Лабораторная работа. Изучение неупругого удара

**Задание:**

1. перейдите по ссылке для **выполнения виртуальной лабораторной работы**;
2. ознакомьтесь с целью, задачами и краткими теоретическими сведениями;
3. изучите порядок выполнения лабораторной работы;
4. выполните работу, фиксируя данные измерений в таблицу;
5. подготовьте отчет по лабораторной работе, заполнив его теоретическими данными и данными измерений в таблицах;
6. выполните необходимые вычисления и занесите их в таблицы;
7. рассчитайте погрешность;
8. сделайте вывод;
9. **ВАЖНО** – по завершении виртуальной лабораторной работы в системе кликните кнопку "Завершить работу" для фиксации ее выполнения вам в базе данных системы.

**Отчет прикрепите в ответ на задание для проверки преподавателем.**

Видимые группы: Все участники  
 Резюме оценивания

Рисунок 21 – Скриншот страницы еКурса по физике с заданием по выполнению виртуальной лабораторной работы

## 2.1 Интерфейс системы

Для создания интерфейса системы использовались SublimeText, Google Chrome. Средствами SublimeText созданы разметка и дизайн всех веб-страниц.

Веб-приложение предполагает две роли пользователя: студент, преподаватель.

Начало работы пользователей не зависит от их роли, им нужно пройти по ссылке на веб-приложение. Студент или преподаватель проходит по ссылке в еКурсе дисциплины «Физика» на веб-приложение «Виртуальные лабораторные работы по физике», далее, перед ним открывается форма входа, чтобы войти или зарегистрироваться, на которой он вводит свои данные (ФИО, номер зачетной книжки). На рисунках 22–25 представлен интерфейс страниц начала работы в системе. Для обеих ролей пользователя интерфейс, за исключением одной страницы, выглядит одинаково.

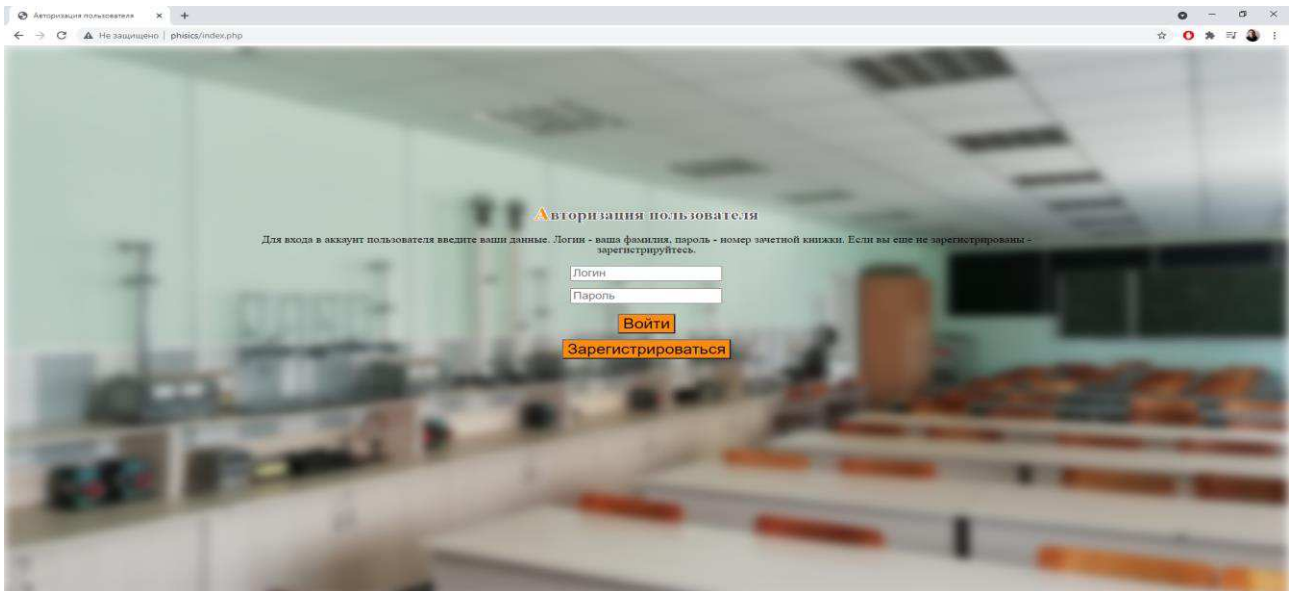


Рисунок 22 – Скриншот страницы авторизации пользователя

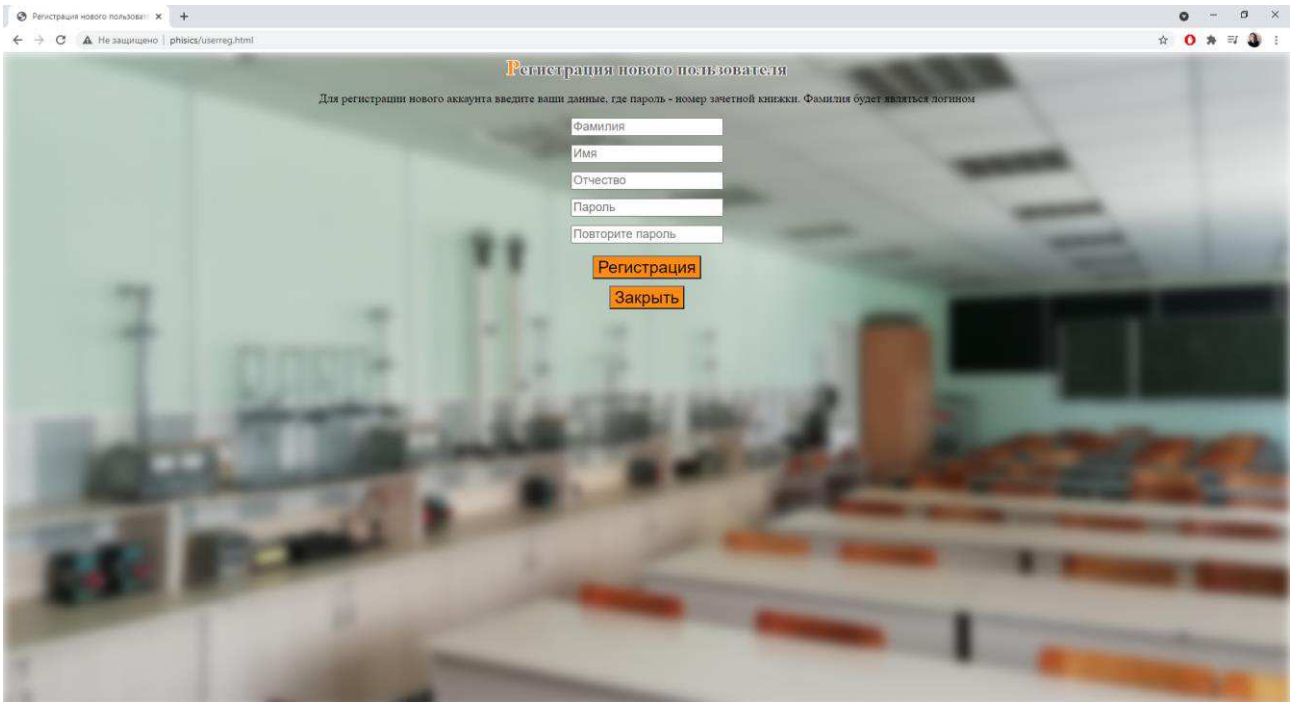


Рисунок 23 – Скриншот страницы регистрации пользователя

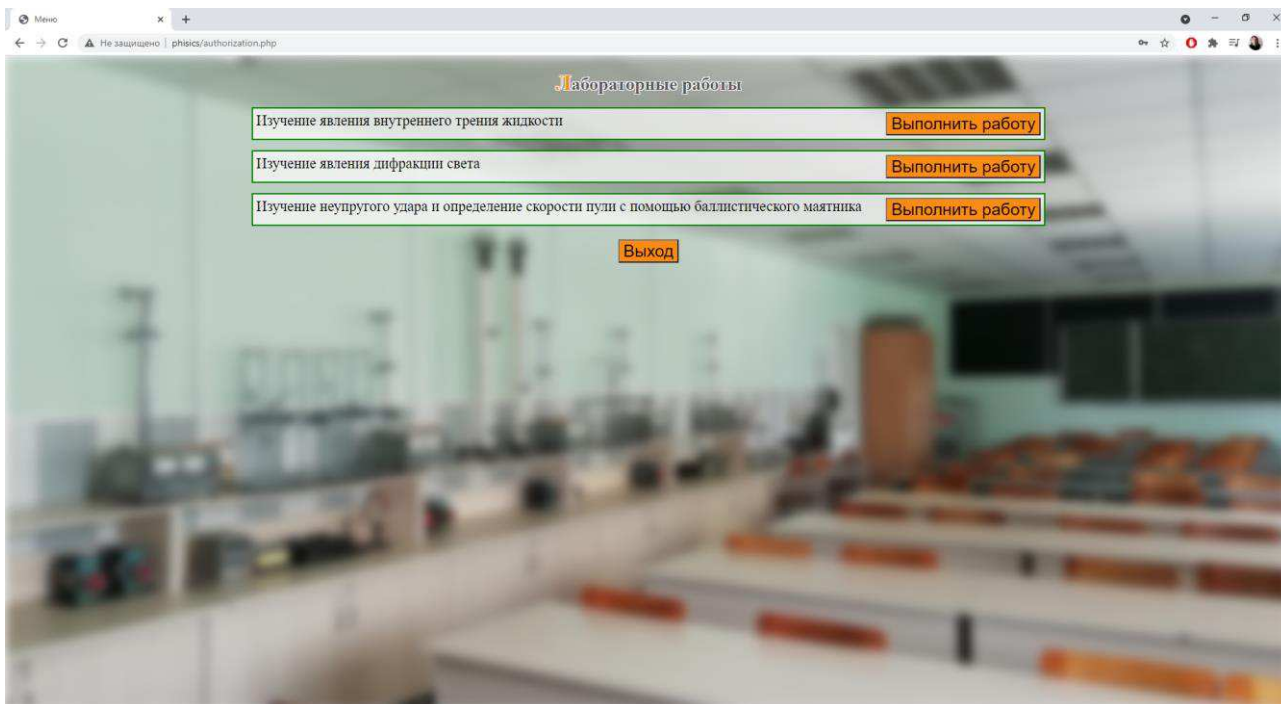


Рисунок 24 – Скриншот страницы меню выбора лабораторной работы для роли «Студент»

После входа под своей учетной записью преподаватель будет видеть список студентов, выполнивших лабораторную работу (рисунок 5), под которым меню выбора лабораторных работ.

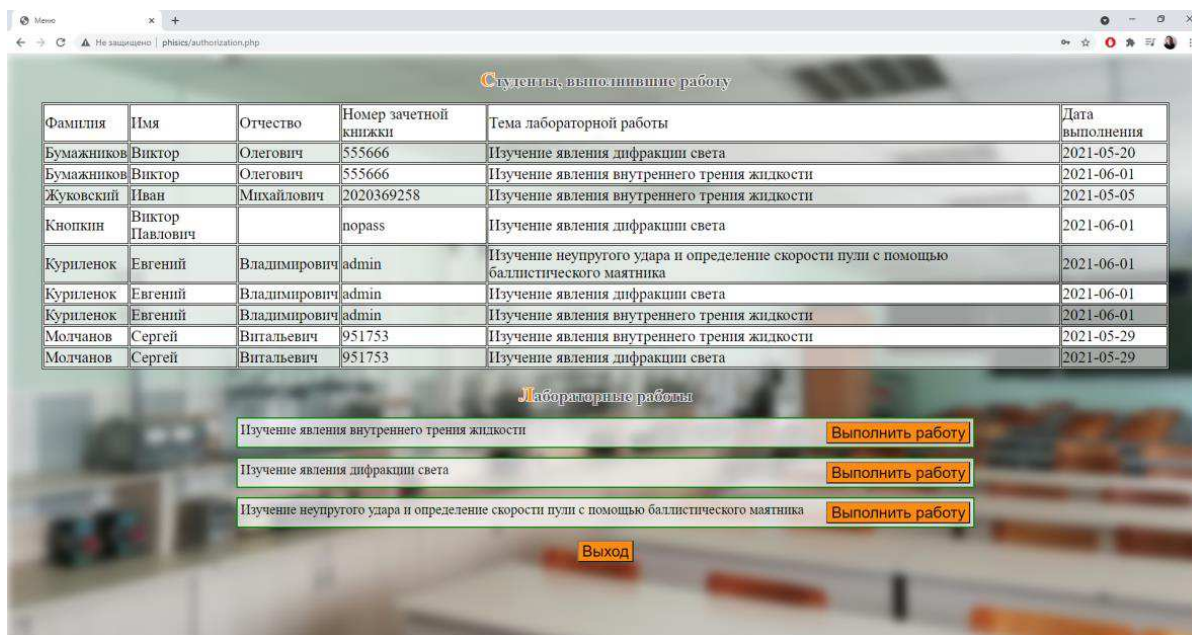


Рисунок 25 – Скриншот страницы меню выбора лабораторной работы для роли «Преподаватель»

На данном этапе разработаны три лабораторные работы:

1. изучение явления внутреннего трения в жидкости;
2. изучение явления дифракции света;
3. изучение неупругого удара [12].

Далее детально представлен интерфейс системы при выборе лабораторной работы «Изучение явления внутреннего трения в жидкости». После выбора лабораторной работы перед пользователем открываются последовательно страницы с пояснениями о цели работы, используемом оборудовании, краткими теоретическими сведениями и порядком выполнения работы (рисунки 26–28).

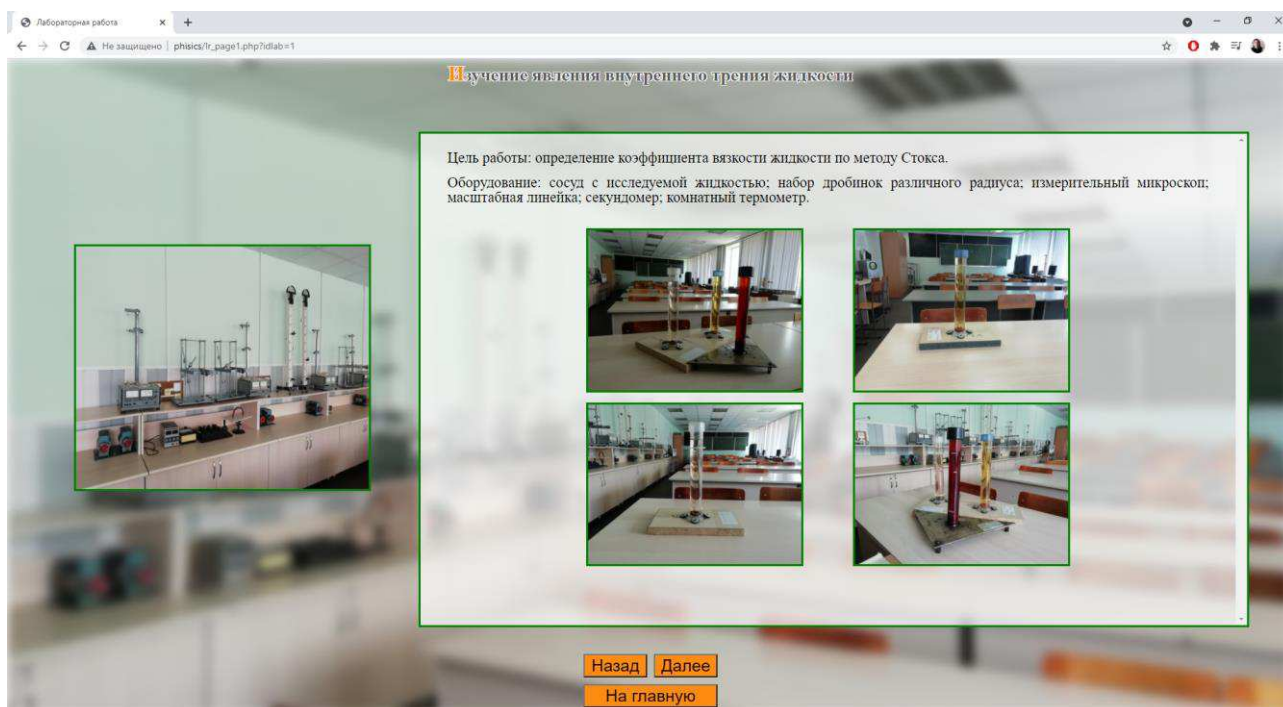


Рисунок 26 – Скриншот первой страницы выполнения лабораторной работы

При разработке интерфейса использованы фотографии, выполненные в физических лабораториях ХТИ – филиала СФУ:

– в лаборатории «Механика и молекулярная физика» для разработки интерфейса работы по изучению явления внутреннего трения в жидкости и по изучению неупругого удара,

– в лаборатории «Оптика» для разработки интерфейса работы по изучению явления дифракции света.

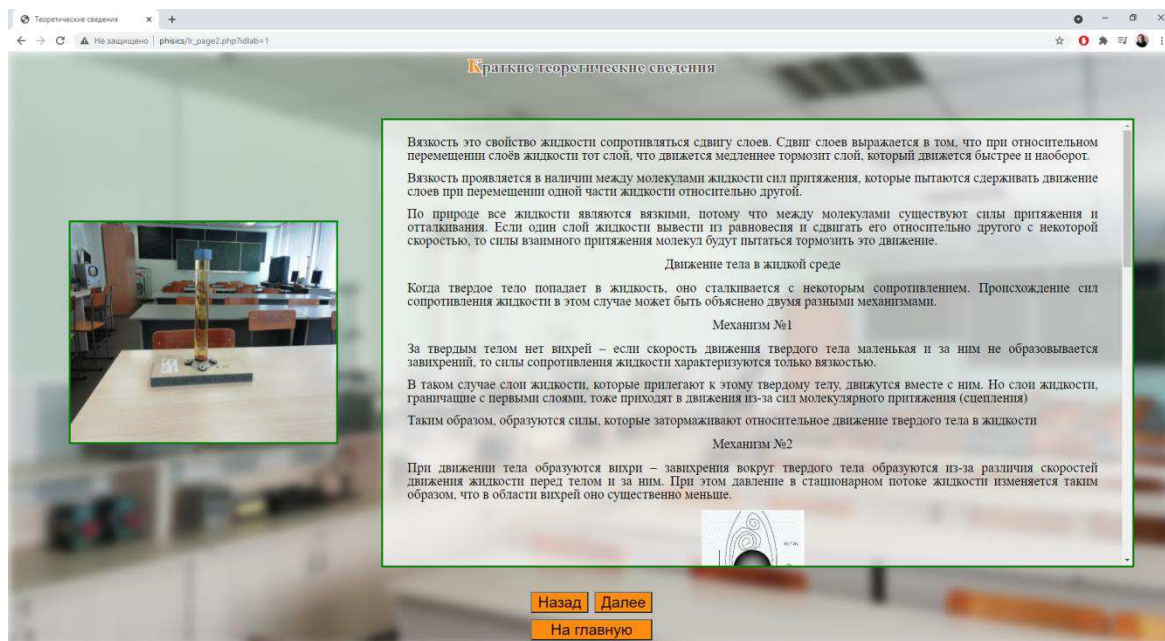


Рисунок 27 – Скриншот второй страницы выполнения лабораторной работы

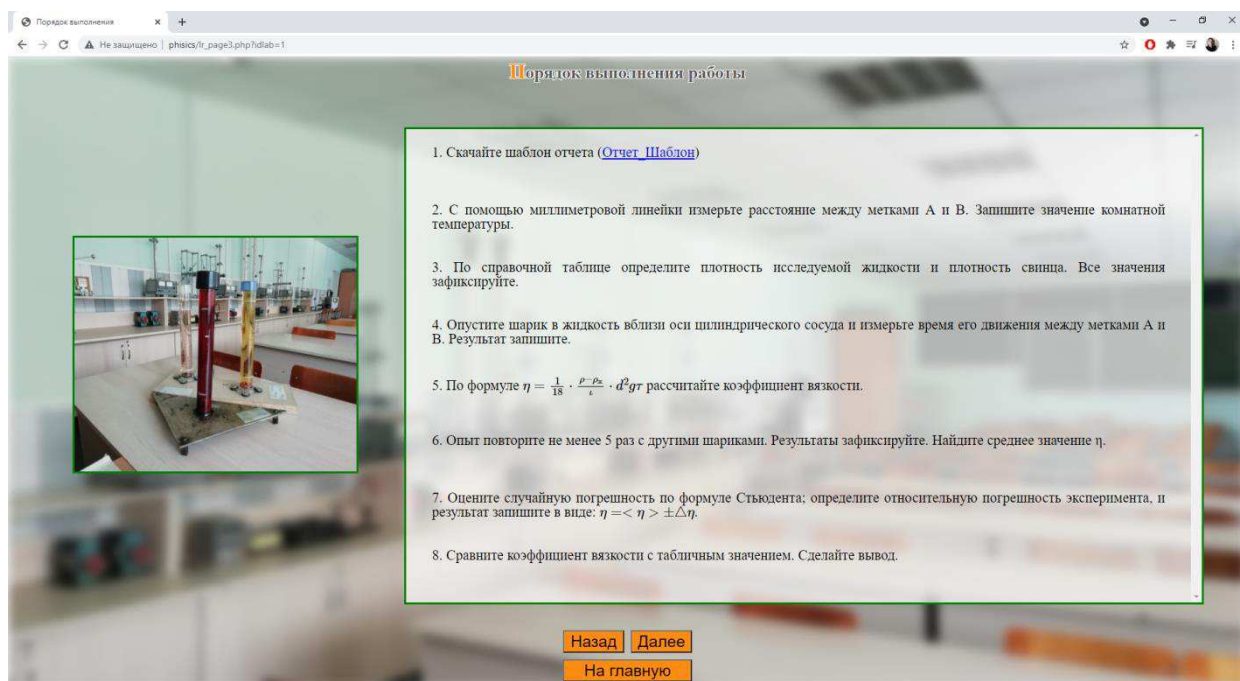


Рисунок 28 – Скриншот третьей страницы выполнения лабораторной работы

Далее осуществляется переход в графический интерфейс с виртуальным оборудованием и полями выбора необходимых данных. Студент в левом меню

выбирает начальные данные и управляет процессом выполнения работы с помощью кнопок правого меню, после чего ему демонстрируется выполнение физического процесса (рисунок 29).

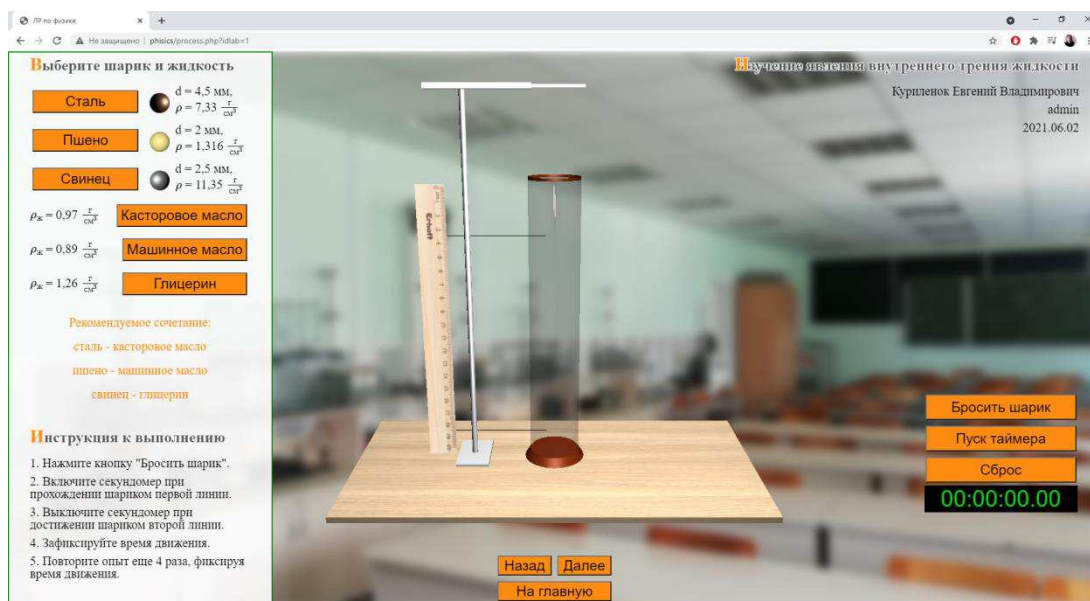


Рисунок 29 – Скриншот страницы демонстрации физического процесса

После выполнения измерений и фиксации результатов пользователь переходит на страницу с пояснениями о подготовке отчета и информации о том, каким образом отчет сдать преподавателю (рисунок 30). На этой же странице размещены кнопки для возврата на главную страницу, назад и кнопка завершения работы, при нажатии на которую записывается информация о пользователе, теме лабораторной работы и дате выполнения.



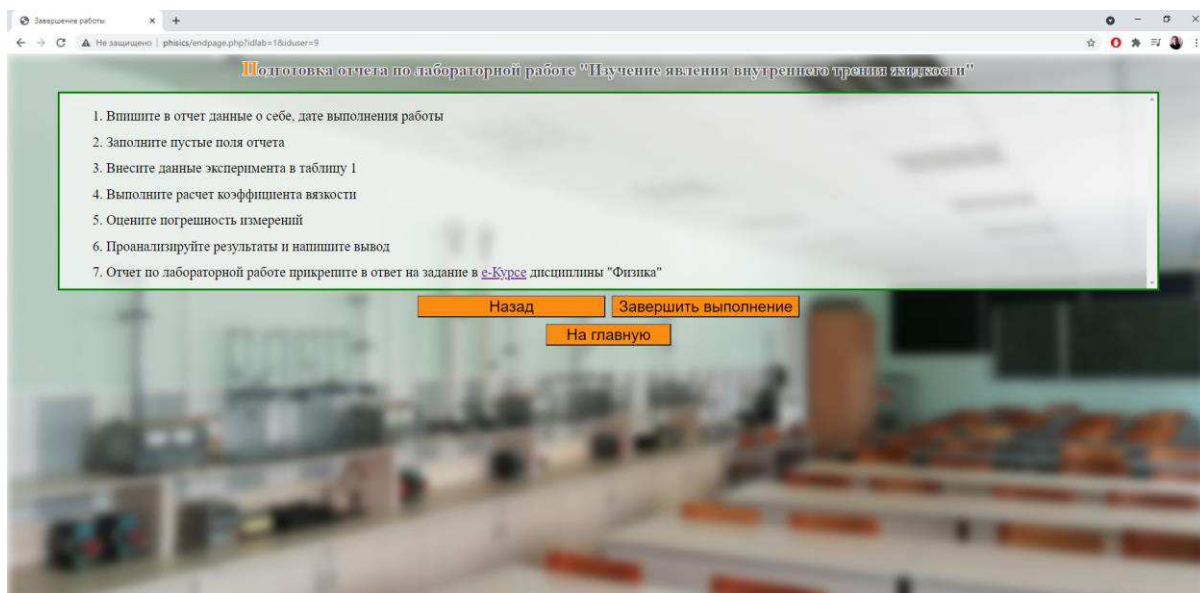


Рисунок 30 – Скриншот страницы завершения работы

Интерфейсы лабораторных работ 2 и 3 аналогичны интерфейсу рассмотренной работы, за исключением страниц на которых демонстрируется физический процесс.

На рисунках 31-35 представлены скриншоты интерфейса лабораторной работы 2.

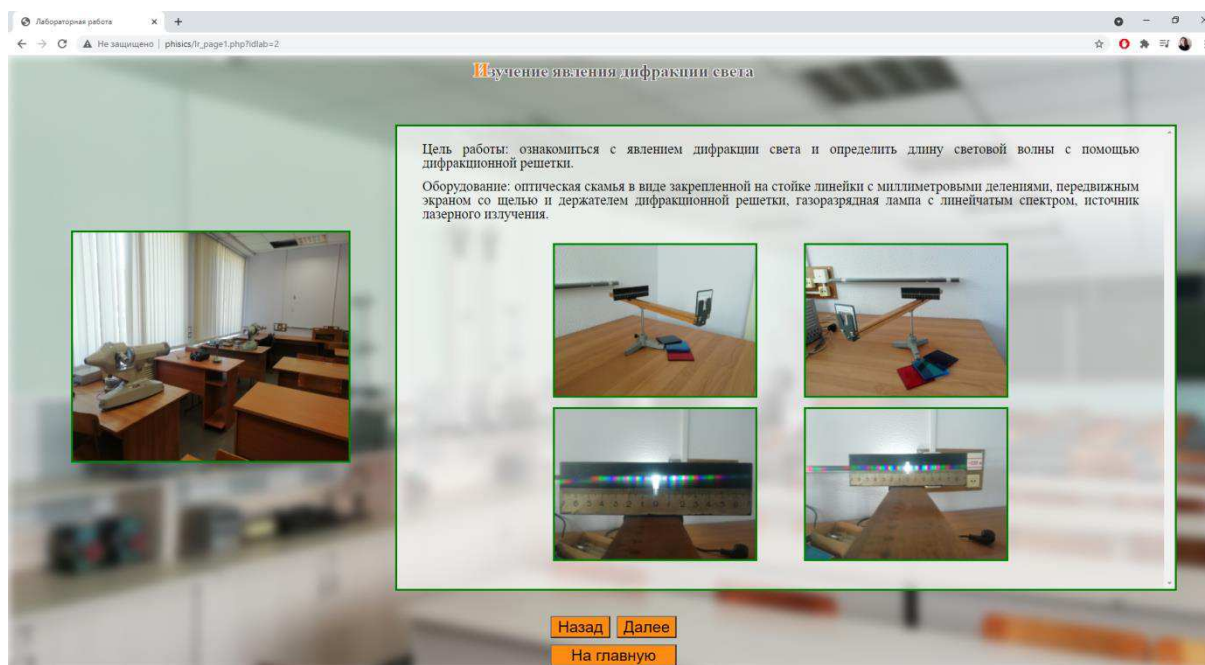


Рисунок 31 – Скриншот первой страницы выполнения лабораторной работы 2

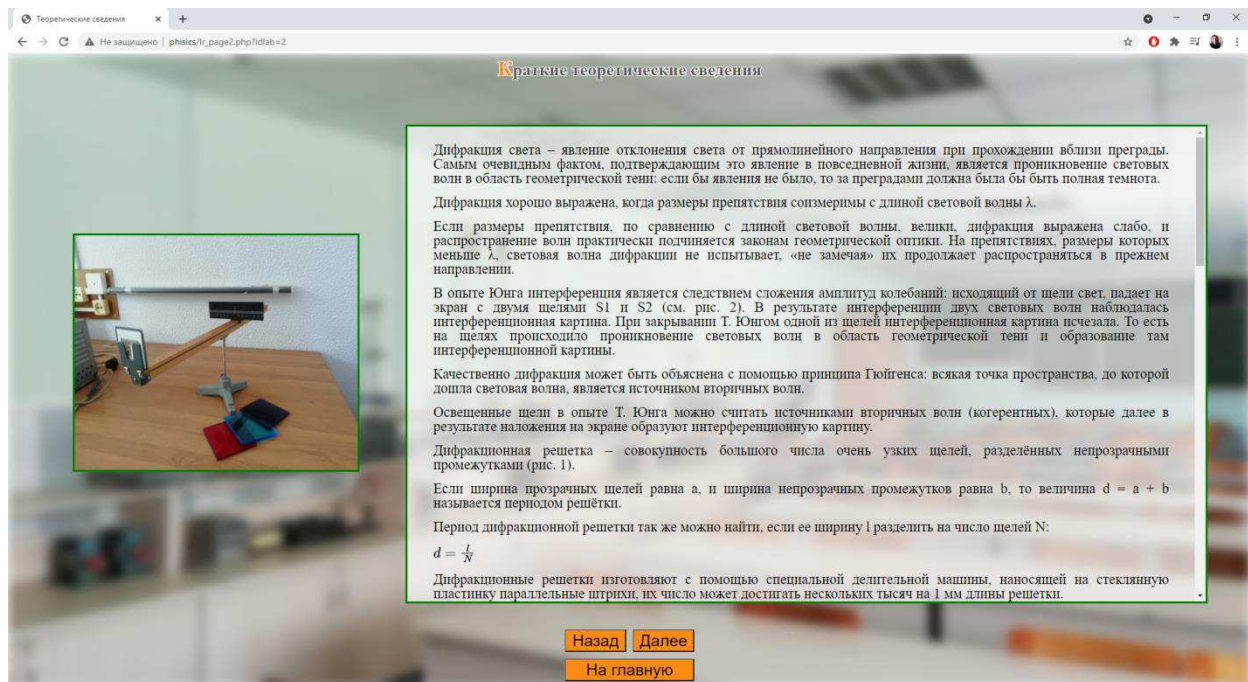


Рисунок 32 – Скриншот второй страницы выполнения лабораторной работы 2

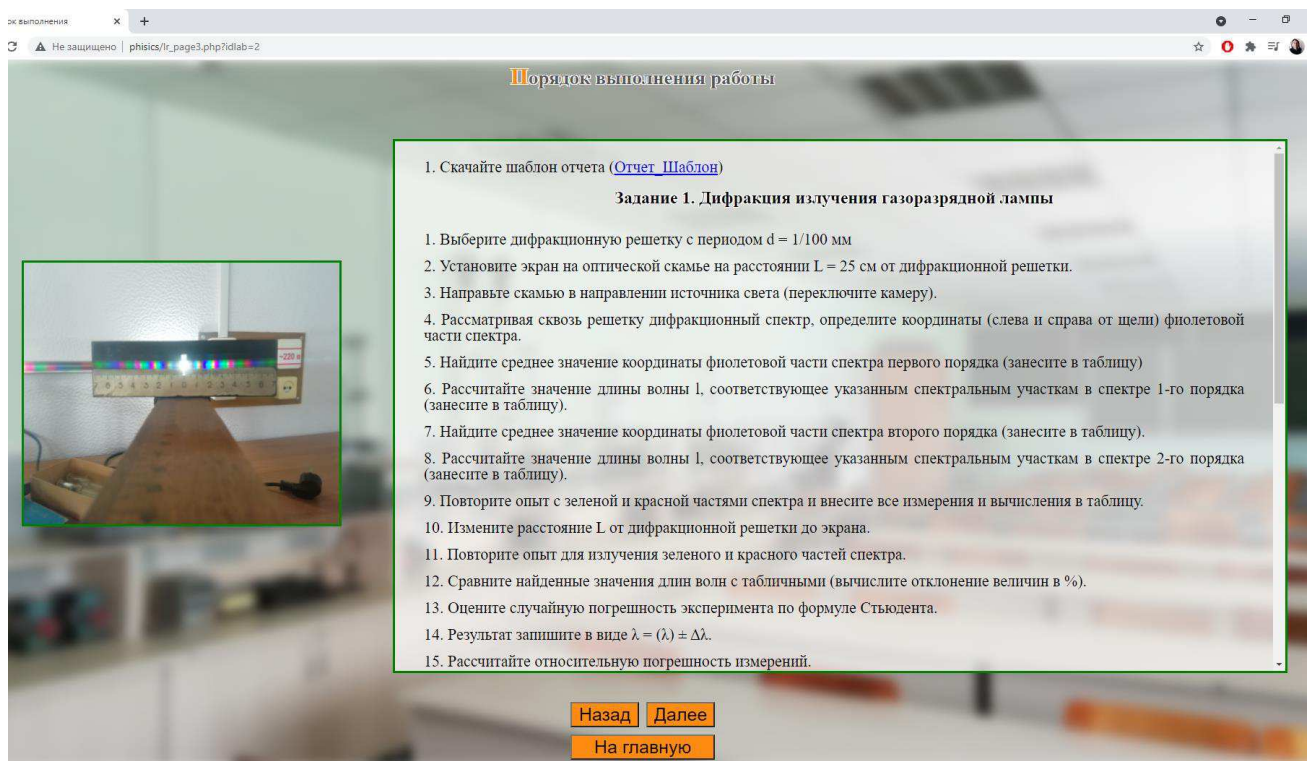


Рисунок 33 – Скриншот третьей страницы выполнения лабораторной работы 2

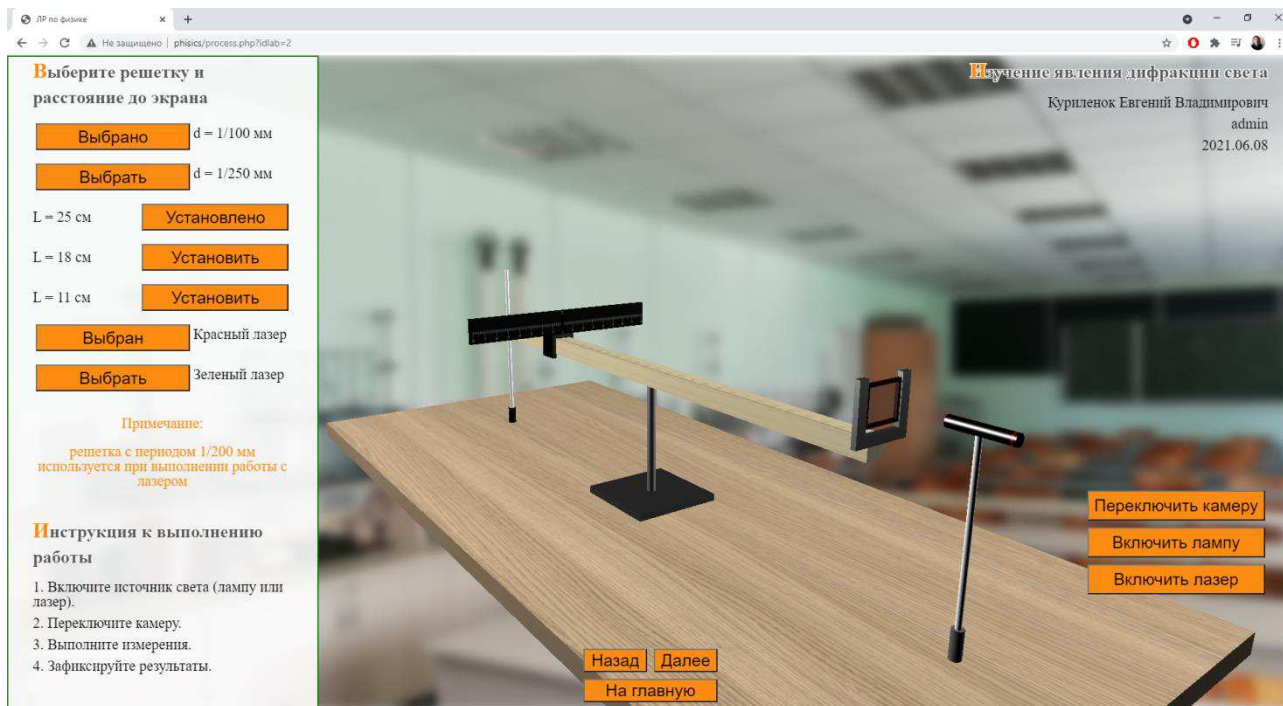


Рисунок 34 – Скриншот страницы демонстрации физического процесса лабораторной работы 2

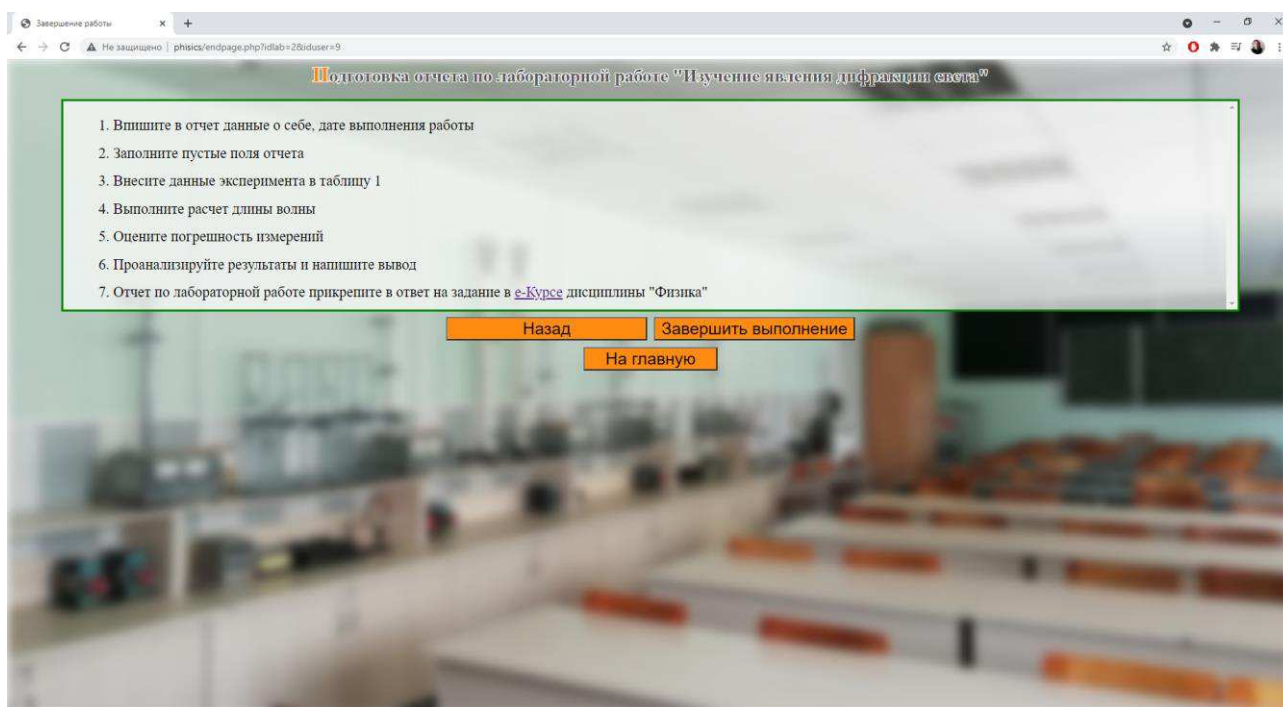


Рисунок 35 – Скриншот страницы завершения лабораторной работы 2

На рисунках 36-40 представлены скриншоты интерфейса лабораторной работы 3.

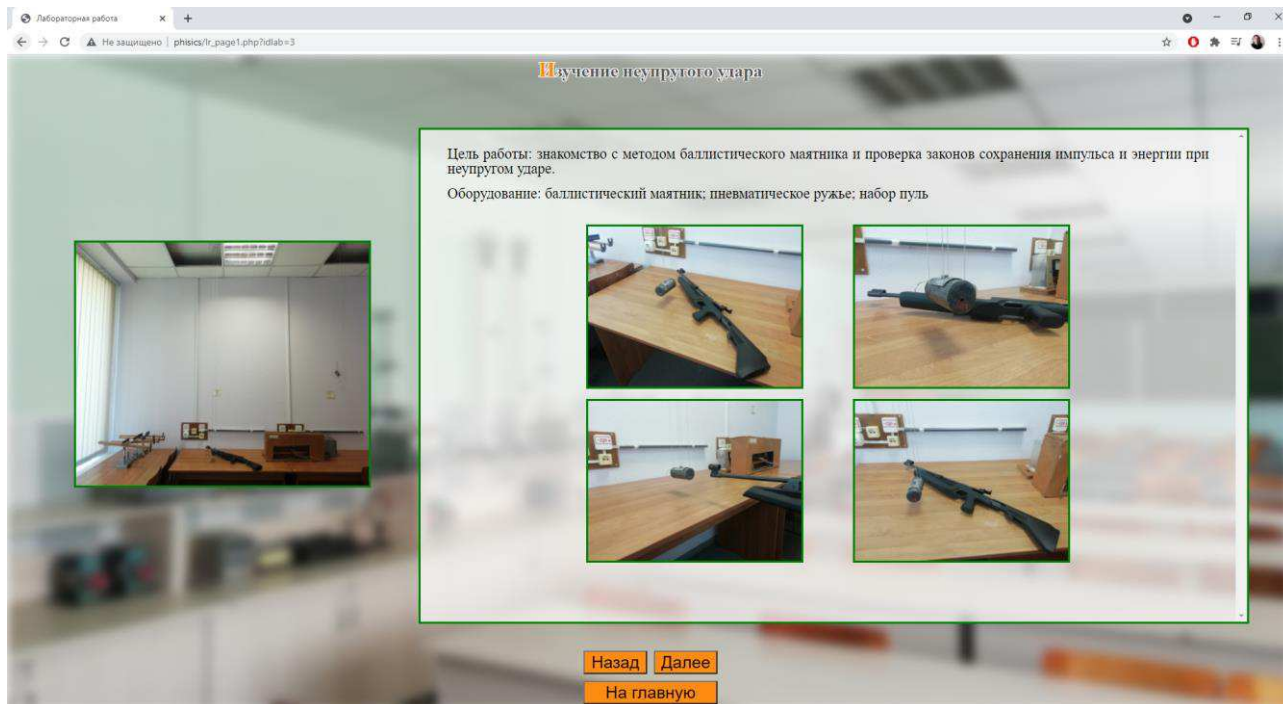


Рисунок 36 – Скриншот первой страницы выполнения лабораторной работы 3

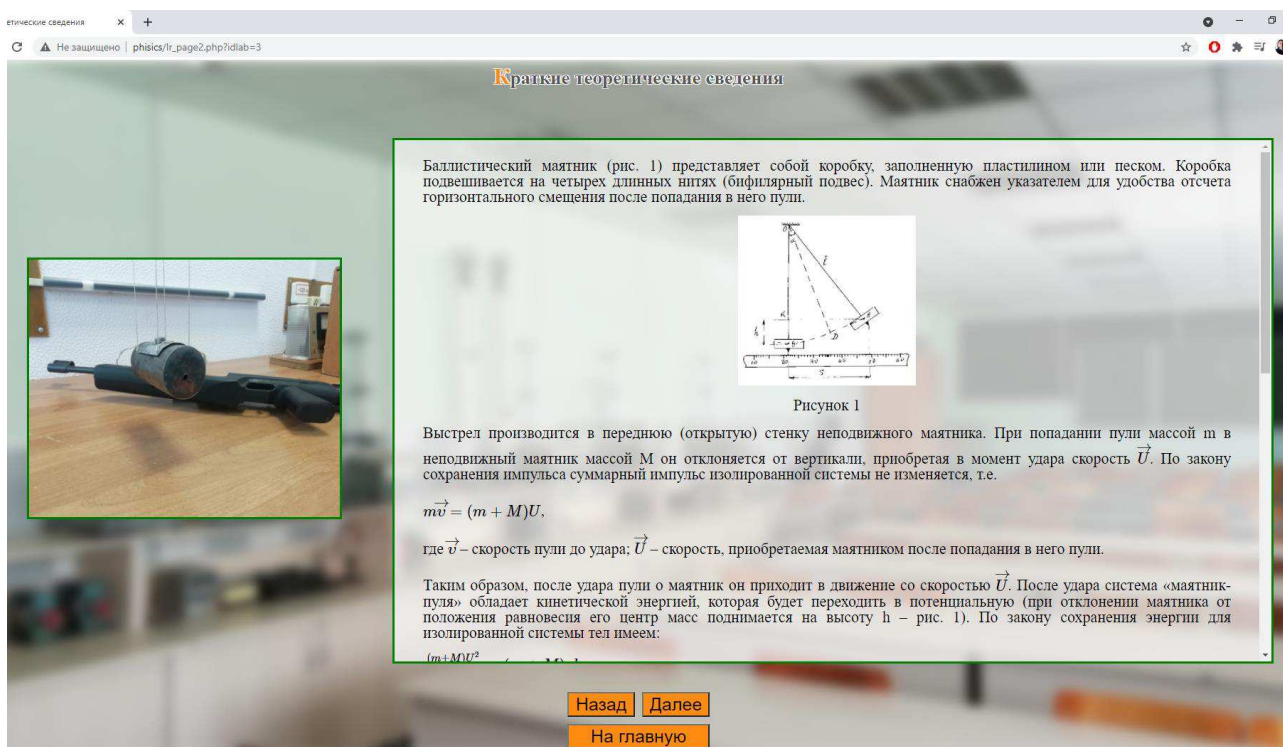


Рисунок 37 – Скриншот второй страницы выполнения лабораторной работы 3



Рисунок 38 – Скриншот третьей страницы выполнения лабораторной работы 3

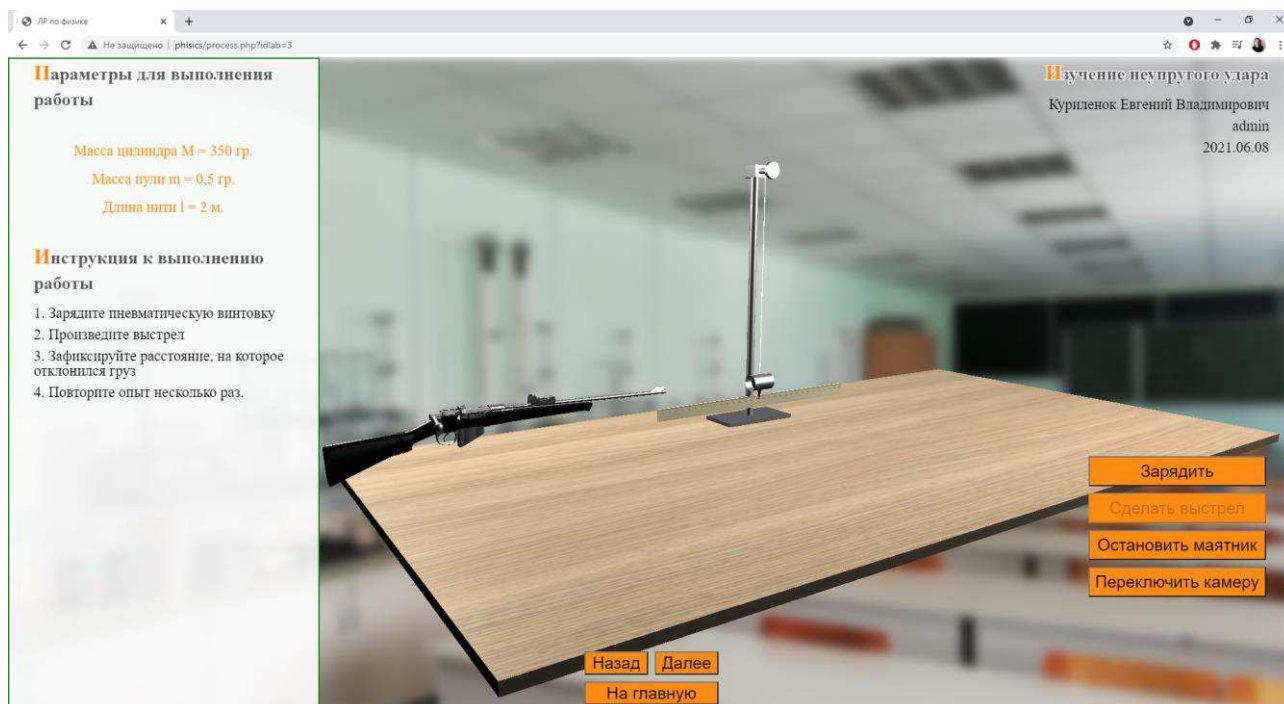


Рисунок 39 – Скриншот страницы демонстрации физического процесса лабораторной работы 3

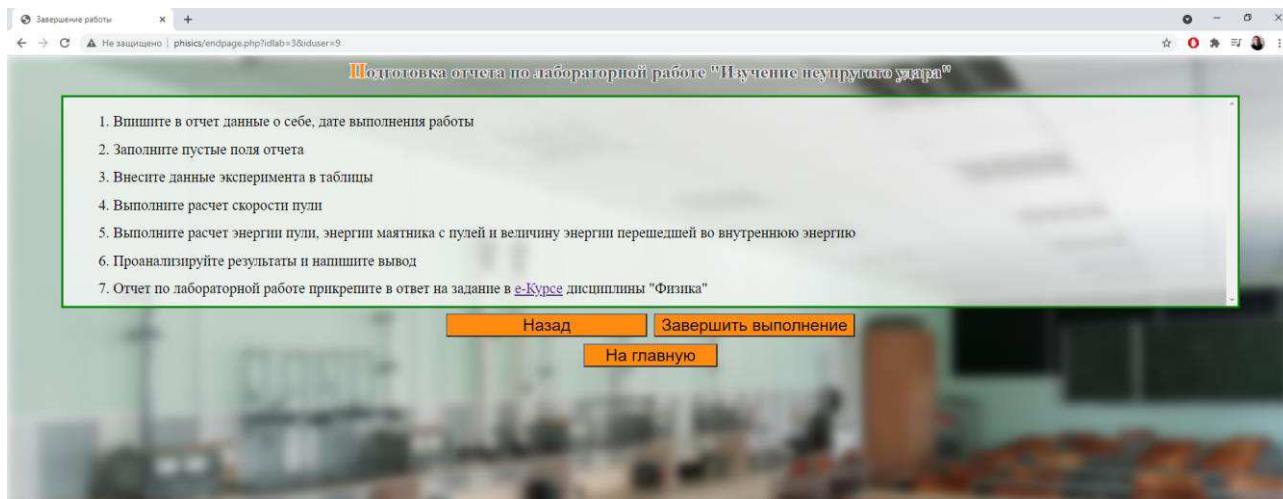


Рисунок 40 – Скриншот страницы завершения лабораторной работы 3

## 2.2 Программный код системы

### 2.2.1 База данных системы

Для полноценной работы веб-приложения с учетными записями пользователей была создана база данных `virtual_lab`, состоящая из трех таблиц (`userdata`, `lab`, `interfaceout`), содержащих информацию о пользователях, лабораторных работах и данных, выводимых на экран при выполнении лабораторной работы.

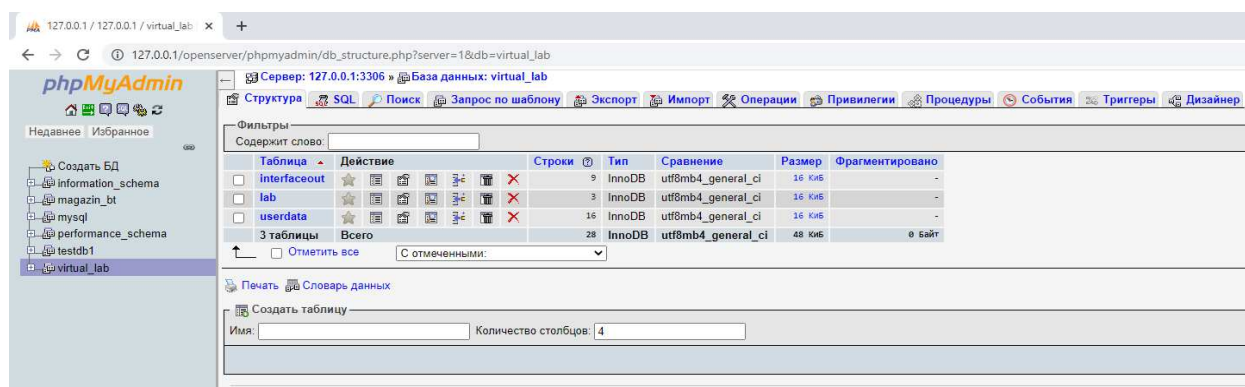


Рисунок 41 – Скриншот созданной базы данных

Сервер: 127.0.0.1:3306 » База данных: virtual\_lab » Таблица: userdata

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id_userdata			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	sname	utf8mb4_general_ci		Да	NULL			
<input type="checkbox"/>	3	name	utf8mb4_general_ci		Да	NULL			
<input type="checkbox"/>	4	tname	utf8mb4_general_ci		Нет	Нет			
<input type="checkbox"/>	5	pass	utf8mb4_general_ci		Да	NULL			
<input type="checkbox"/>	6	date			Да	current_timestamp()			
<input type="checkbox"/>	7	account_name	utf8mb4_general_ci		Нет	Нет			

Отметить все С отмеченными:

Рисунок 42 – Скриншот структуры таблицы userdata

Сервер: 127.0.0.1:3306 » База данных: virtual\_lab » Таблица: lab

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id_lab			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	name_lab	utf8mb4_general_ci		Нет	Нет			
<input type="checkbox"/>	3	theme_lab	utf8mb4_general_ci		Нет	Нет			

Отметить все С отмеченными:

Рисунок 43 – Скриншот структуры таблицы lab

Сервер: 127.0.0.1:3306 » База данных: virtual\_lab » Таблица: interfaceout

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id_interfaceout			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	id_userdata			Нет	Нет			
<input type="checkbox"/>	3	id_lab			Нет	Нет			
<input type="checkbox"/>	4	date			Да	NULL			

Отметить все С отмеченными:

Рисунок 44 – Скриншот структуры таблицы interfaceout

Созданная база данных позволяет хранить повторяющуюся информацию и в случае необходимости быстро и упорядоченно ее выводить с помощью sql-запросов внедренных в веб-страницы с помощью языка программирования php.

## 2.2.2 Разметка и функционал веб-страниц

Для начальной страницы веб-приложения «Виртуальные лабораторные работы по физике» или страницы авторизации пользователя был создан файл

index.php, с расширением php для работы функционала очистки текущей сессии. С помощью html разметки и css стилей был создан дизайн этой страницы, а также отправка данных и проверка заполнения полей с помощью javascript-функций. На странице размещены две формы с кнопками, для перенаправления пользователя на страницу регистрации и страницу входа в систему, если пользователь уже зарегистрирован. На рисунке 45 представлен скриншот кода файла index.php.

Разметку страницы регистрации содержит файл userreg.html, в котором содержится код (рисунок 46), добавляющий на страницу форму с input-полями для ввода данных о новом пользователе, кнопку, отправляющие введенные данные в php-файл для дальнейшей обработки, а также, кнопку выхода на страницу авторизации.

Кроме того, на этой странице тоже есть javascript функция, проверяющая пустые поля в форме, в случае обнаружения которых выдает сообщение о том, что пользователю нужно заполнить все необходимые поля.

После ввода данных на странице авторизации и нажатия на кнопку «вход», пользователь перенаправляется на страницу входа, она же – страница меню выбора лабораторных работ. Код этой страницы содержит файл authorization.php (рисунки 47, 48). Код этого файла производит обработку данных, полученных из формы страницы авторизации, и выводит соответствующую информацию на веб-страницу, в зависимости от того, какие данные получены.



```
index.php x
1 <?php session_start();
2 session_destroy();?>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Авторизация пользователя</title>
8 <link rel="stylesheet" type="text/css" href="reset.css">
9 <link rel="stylesheet" type="text/css" href="styles.css">
10 <link rel="stylesheet" type="text/css" href="mobstyles.css">
11 <meta name="viewport" content="width=device-width, initial-scale=1">
12 </head>
13 <script type="text/javascript">
14     function CheckFields(){
15         met = 0;
16         for (i=0; i<6; i++) {
17             if (document.forms[0].elements[i].value=="") {
18                 alert("Введите данные");
19                 document.forms[0].elements[i].focus();
20                 met = 1;
21                 break;
22             }
23         }
24         if (met!=1) {
25             return true;
26         }
27         else {
28             return false;
29         }
30     }
31     function CheckFields2(){
32         return true;
33     }
34 </script>
35 <body>
36 <div class="box">
37 <h1 class="h1">Авторизация пользователя</h1>
38 <div class="form">
39 <p>Для входа в аккаунт пользователя введите ваши данные. Логин - ваша фамилия, пароль - номер зачетной книжки. Если вы
еще не зарегистрированы - зарегистрируйтесь.</p>
40 <form name="myform" action="authorization.php" method="post" onsubmit="return CheckFields()" >
41     <div class="form">
42         <input id='i1' type="text" name="surname" placeholder="Логин" class="input1">
43         <input id='i2' type="password" name="password" placeholder="Пароль" class="input1">
44         <input type="submit" value="Войти" name="button" class="input2" >
45     </div>
46 </form>
47 <form name="myform" action="userreg.html" method="post" onsubmit="return CheckFields2()" >
48     <input type="submit" value="Зарегистрироваться" name="button" class="input2" >
49 </form>
50 </div>
51 </div>
52 </body>
53 </html>
```

Рисунок 45 – Скриншот кода index.php

```

userreg.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Регистрация нового пользователя</title>
6 <link rel="stylesheet" type="text/css" href="reset.css">
7 <link rel="stylesheet" type="text/css" href="styles.css">
8 <link rel="stylesheet" type="text/css" href="mobstyles.css">
9 <meta name="viewport" content="width=device-width, initial-scale=1">
10 </head>
11 <script type="text/javascript">
12 function CheckFields(){
13     met = 0;
14     for (i=0; i<6; i++) {
15         if (i==2) {
16             if (document.forms[0].elements[i].value=="") {
17                 alert("Заполните все поля!");
18                 document.forms[0].elements[i].focus();
19                 met = 1;
20                 break;
21             }
22         }
23     }
24     if (document.forms[0].elements[3].value != document.forms[0].elements[4].value) {
25         alert("Полн ПАРОЛЬ не совпадает!");
26         document.forms[0].elements[4].focus();
27         met = 1;
28     }
29     if (met==1) {
30         return true;
31     }
32     else {
33         return false;
34     }
35 }
36 function CheckFields2(){
37     return true;
38 }
39 </script>
40 <body>
41 <h1 class="h1">Регистрация нового пользователя</h1>
42 <div class="form">
43 <div class="form">
44 <div class="form">
45 <div class="form">
46 <div class="form">
47 <div class="form">
48 <div class="form">
49 <div class="form">
50 <div class="form">
51 <div class="form">
52 <div class="form">
53 <div class="form">
54 <div class="form">
55 <div class="form">
56 <div class="form">
57 <div class="form">
58 <div class="form">
59 <div class="form">

```

Рисунок 46 – Скриншот кода userreg.html

```

authorization.php
1 <?php session_start(); ?>
2 <!DOCTYPE html>
3 <html >
4 <head>
5 <meta charset="UTF-8">
6 <title>Меню</title>
7 <link rel="stylesheet" type="text/css" href="styles.css">
8 <link rel="stylesheet" type="text/css" href="mobstyles.css">
9 </head>
10 <script type="text/javascript">
11 function CheckFields(){
12     return true;
13 }
14 </script>
15 <?php
16 $p1 = $_SESSION['sname'];
17 $p2 = $_SESSION['pass'];
18 if ($p1==null || $p2==null) {
19     $p1 = $_POST['surname'];
20     $p2 = $_POST['password'];
21     $_SESSION['sname'] = $p1;
22     $_SESSION['pass'] = $p2;
23 }
24 $link = mysqli_connect('localhost', 'root', '', 'virtual_lab');
25 $query0 = mysqli_query($link, "SET NAMES \utf8mb4"") or die("Ошибка " . mysqli_error($link));
26 $query1 = mysqli_query($link, "SELECT * FROM userdata WHERE sname = \"\$p1\" AND pass = \"\$p2\"");
27 $query3 = mysqli_query($link, "SELECT * FROM lab");
28 if ( mysqli_num_rows($query1) != 0 ) {
29     $query2 = mysqli_query($link, "SELECT * FROM userdata WHERE sname = \"\$p1\" AND pass = \"\$p2\" AND
30     account_name = \"admin\" ");
31 $query1 = mysqli_query($link, "SELECT `userdata`.`id_userdata`, `userdata`.`sname`, `userdata`.`name`, `userdata`.`
32 tname`, `userdata`.`pass`, `userdata`.`account_name`, `interfaceout`.`date`, `lab`.`name_lab` FROM `userdata`,
33 `interfaceout`, `lab` WHERE (`userdata`.`id_userdata` = `interfaceout`.`id_userdata`) AND (`interfaceout`.`id_lab`
34 = `lab`.`id_lab`) ORDER BY `sname`");
35 }
36 <?>
37 <body>
38 <h1 class="h1">Студенты, выполнившие работу</h1>
39 <table class="table1">
40 <tr bgcolor="white">
41 <td >Фамилия</td>
42 <td >Имя</td>
43 <td >Отчество</td>
44 <td >Номер зачетной книжки</td>
45 <td >Тема лабораторной работы</td>
46 <td >Дата выполнения</td>
47 </tr>
48 </table>
49 <?php
50 $num = 1;
51 while ($row=mysqli_fetch_array($query1)){
52     echo "<tr><td>";

```

Рисунок 47 – Скриншот кода authorization.php

```

authorization.php x
45 <?php
46 $num = 1;
47 while ($row=mysqli_fetch_array($query1)){
48     $pole1=$row['id_userdata'];
49     $pole2=$row['sname'];
50     $pole3=$row['name'];
51     $pole4=$row['tname'];
52     $pole5=$row['date'];
53     $pole6=$row['pass'];
54     $pole7=$row['name_lab'];
55     $pole8=$row['account_name'];
56     if ($pole8=="admin") {
57         $pole6=$pole8;
58     }
59     if (($num % 2) == 0) {
60         $bgcolor = "white";
61     } else {
62         $bgcolor = "";
63     }
64     ?>
65     <tr bgcolor="<?php echo ($bgcolor)?>">
66         <td><?php echo "$pole2"?></td>
67         <td><?php echo "$pole3"?></td>
68         <td><?php echo "$pole4"?></td>
69         <td><?php echo "$pole6"?></td>
70         <td><?php echo "$pole7"?></td>
71         <td><?php echo "$pole5"?></td>
72     </tr>
73     <?php
74     $num++;
75 }
76 ?>
77 </table>
78 <?php
79 }
80 ?>
81 <h1 class="h1">Лабораторные работы</h1>
82 <div class="form">
83 <?php while ($row=mysqli_fetch_array($query3)){
84     $id_lab=$row['id_lab'];
85     $name_lab=$row['name_lab']; ?>
86     <div class="menu_lr">
87         <p><?php echo($name_lab)?></p>
88         <form name="myform" action="lr_page1.php?idlab=<?php echo($id_lab) ?>" method="post" onsubmit="return
89             CheckFields()" >
90             <input type="submit" value="Выполнить работу" name="button" class="input2" >
91         </form>
92     </div>
93 <?php }
94 else {
95     ?>
96     <div class="form">
97         <p>Пользователя с такой фамилией или паролем не обнаружено</p>
98     <?php
99     }
100     ?>
101     <form name="myform" action="index.php" method="post" onsubmit="return CheckFields()" >
102         <input type="submit" value="Выход" name="button" class="input2" >
103     </form>
104 </div>
105 <?php
106 mysqli_close($link);
107 ?>
108 </body>
109 </html>

```

Рисунок 48 – Скриншот продолжения кода authorization.php

При открытии этой страницы запускается функция `session_start`, которая создает новую сессию, для хранения данных, используемых на всех страницах веб-приложения (логин и пароль). Далее php код записывает в новые переменные данные из файла сессии и проверяет эти переменные, на наличие информации. Если в них ничего не записалось, то в эти переменные записываются данные, полученные методом post со страницы авторизации, после чего, информация из этих переменных записывается в файл сессии. Это

сделано для того, чтобы была возможность перехода на эту страницу не только со страницы авторизации (в веб-приложении реализована кнопка «Назад»).

После этого, логин и пароль, полученные одним из перечисленных способов, проверяются в базе данных. Создается подключение к базе данных, производится запрос, получающий все записи из таблицы лабораторных работ (таблица lab), и запрос в таблицу пользователей (таблица userdata), на наличие записи с таким логином и паролем и, если результат не отрицательный, производится запрос на тип учетной записи (преподавательский) в эту же таблицу. Если результат и этого запроса не отрицательный, то производится запрос, на получение информации, о студентах, выполнивших лабораторные работы, результат которого выводится в виде таблицы. После вывода таблицы, на странице создается список лабораторных работ, полученный из запроса, с кнопками перехода к их выполнению, после которого отображается кнопка выхода на страницу авторизации. В случае отрицательного результата на запрос о типе учетной записи вывод таблицы пропускается. В случае отрицательного запроса на наличие в базе данных пользователя с полученными логином и паролем, на страницу выводится соответствующее сообщение и кнопка выхода.

Кнопки перехода к выполнению генерируются для всех лабораторных работ одинаково в виде формы с полем input (тип button), которая, при нажатии на кнопку, перенаправляет пользователя на первую страницу с теорией, передавая методом get номер лабораторной работы.

После выбора лабораторной работы пользователь переходит к теории об этой работе, содержащейся на трех страницах, код которых представлен в трех файлах lr\_page1, lr\_page2, lr\_page3 с расширением .php для генерации разной выводимой информации, в зависимости от номера работы.

Теория разделена по страницам на цель лабораторной работы и оборудование, краткие теоретические сведения, порядок выполнения работы. Разметка и дизайн всех трех страниц полностью одинаковый, изменяется

только контент, выводимый на этих страницах. Код одной из таких страниц представлен на рисунке 49.

```
lr_page1.php
1 <?php session_start(); ?>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="UTF-8">
6 <title>лабораторная работа</title>
7 <link rel="stylesheet" type="text/css" href="reset.css">
8 <link rel="stylesheet" type="text/css" href="styles.css">
9 <link rel="stylesheet" type="text/css" href="mobstyles.css">
10 <meta name="viewport" content="width=device-width, initial-scale=1">
11 </head>
12 <?php
13 $id_lr = $_GET["idlab"];
14 $link = mysqli_connect('localhost', 'root', '', 'virtual_lab');
15 $query0 = mysqli_query($link, "SET NAMES \utf8mb4\"); or die("Ошибка " . mysqli_error($link));
16 $query1 = mysqli_query($link, "SELECT * FROM lab WHERE id_lab = \"{$id_lr}\" ");
17 while ($row = mysqli_fetch_array($query1)){
18     $name_lab = $row['name_lab'];
19 } ?>
20 <script type="text/javascript">
21     function ChackFields2(){
22         return true;
23     }
24 </script>
25 <body>
26 <h1 class="h1">?php echo($name_lab) ?</h1>
27 <div class="content_box">
28     <div class="info_container">
29         <div class="image_box">
30             
31         </div>
32         <div class="info_box">
33             <div class="background"></div>
34             <div class="info1">
35                 <div class="info1_page1">
36                     <?php if ($id_lr==1) {
37                         ?>
38                         <p><>Цель работы: определение коэффициента вязкости жидкости по методу Стокса.</p>
39                         <p><>Оборудование: сосуд с исследуемой жидкостью; набор дробинки различного радиуса; измерительный микроскоп; масштабная линейка; секундомер; комнатный термометр.</p>
40                     <?php }
41                     if ($id_lr==2) { ?>
42                         <p><>Цель работы: ознакомиться с явлением дифракции света и определить длину световой волны с помощью дифракционной решетки.</p>
43                         <p><>Оборудование: оптическая скамья в виде закрепленной на стойке линейки с миллиметровыми делениями, передвижной экран со щелью и держателем дифракционной решетки, газоразрядная лампа с линейчатым спектром, источник лазерного излучения.</p>
44                     <?php }
45                     if ($id_lr==3) { ?>
46                         <p><>Цель работы: знакомство с методом баллистического маятника и проверка законов сохранения импульса и энергии при неупругом ударе.</p>
47                         <p><>Оборудование: баллистический маятник; пневматическое ружье; набор пуль.</p>
48                     <?php } ?>
49                 </div>
50             </div>
51         </div>
52     </div>
53 </div>
54 <div class="downcontainer">
55     <div class="downcontainer_upbutton">
56         <form name="myform" action="authorization.php" method="post" onsubmit="return ChackFields()" >
57             <input type="submit" value="Назад" name="button" class="input2" >
58         </form>
59         <form name="myform" action="lr_page2.php?idlab=?php echo($id_lr) ?" method="post" onsubmit="return ChackFields()" >
60             <input type="submit" value="Далее" name="button" class="input2" >
61         </form>
62     </div>
63     <div class="downcontainer_downbutton">
64         <form name="myform" action="authorization.php" method="post" onsubmit="return ChackFields()" >
65             <input type="submit" value="На главную" name="button" class="input2" >
66         </form>
67     </div>
68 </div>
69 </body>
70 </html>
```

Рисунок 49 – Скриншот кода lr\_page1.php

Методом getфайл получает номер лабораторной работы, после чего подключается к базе данных и делает запрос на название лабораторной работы из таблицы lab. Разметка страницы представлены двумя блоками – текстовым и блоком изображения. С помощью html-тега img на страницу загружается изображение для номера текущей лабораторной работы (см. строку 30, рисунок 49), в текстовый блок, в зависимости от текущей лабораторной работы, выводится разная информация о цели и оборудовании. После этих двух блоков на страницу выводится блок, содержащий три формы с кнопками, для перехода

на предыдущую страницу, следующую страницу и главную страницу. В случае страницы lr\_page1 переход пользователя назад и на главную перенаправляет его на одну и ту же страницу – authorization, работа которой без получения логина и пароля описана ранее.

После нажатия на кнопку далее, пользователь последовательно перенаправляется на следующие страницы.

Разметка страниц lr\_page2 и lr\_page3 такая же, как и у страницы lr\_page1. В блоке изображения отображается тематическая фотография для выбранной лабораторной работы, в текстовом блоке выводится информация о кратких теоретических сведениях (lr\_page2) и порядке выполнения работы (lr\_page3).

Кнопка «Далее» перенаправляет пользователя на страницу выполнения лабораторной работы и демонстрации физического процесса process, код которой содержится в файле process.php (рисунки 50–55). С помощью html разметки и стилей css сделан дизайн этой страницы, включающий левое меню выбора параметров для выполнения работы, краткую инструкцию управления процессом, правое меню управления виртуальной лабораторной установкой и нижнее меню для перехода между страницами.

После запуска страница обращается к текущей сессии, для получения данных пользователя, подключается к базе данных и производит запрос на наличие пользователя с таким логином и паролем (рисунок 50). Если сессия по каким-то причинам была прервана, запрос вернет отрицательный результат и пользователю будет предложено войти в систему заново.

```
process.php
1 <?php session_start(); ?>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="UTF-8">
6 <title>JP no физике</title>
7 <link rel="stylesheet" type="text/css" href="reset.css">
8 <link rel="stylesheet" type="text/css" href="styles.css">
9 <meta name="viewport" content="width=device-width, initial-scale=1">
10 </head>
11 <body class="body_process">
12 <?php
13 $p1 = $_SESSION['sname'];
14 $p2 = $_SESSION['pass'];
15 $id_lr = $_GET['idlab'];
16 $link = mysqli_connect('localhost', 'root', '', 'virtual_lab');
17 $query0 = mysqli_query($link, "SET NAMES 'utf8mb4'" or die("Ошибка " . mysqli_error($link));
18 $query1 = mysqli_query($link, "SELECT * FROM userdata WHERE sname = \"\$p1\" AND pass = \"\$p2\"");
19 IF ( mysqli_num_rows($query1) != 0) {
20 $query1 = mysqli_query($link, "SELECT `userdata`.`id_userdata`, `userdata`.`sname`, `userdata`.`name`, `userdata`.`tname`, `userdata`.`pass`, `userdata`.`account_name`, `lab`.`
    id_lab`, `lab`.`name_lab` FROM `userdata`, `lab` WHERE (`userdata`.`sname` = \"\$p1\") AND (`userdata`.`pass` = \"\$p2\") AND (`lab`.`id_lab` = \"\$id_lr\")");
21 while ($row=mysqli_fetch_array($query1)){
22     $id=$row['id_userdata'];
23     $sname=$row['sname'];
24     $name=$row['name'];
25     $phathername=$row['tname'];
26     $password=$row['pass'];
27     $idlab=$row['id_lab'];
28     $labname=$row['name_lab'];
29     $account_name=$row['account_name'];
30     IF ($account_name=="admin") {
31         $password=$account_name;
32     }
33 } ?>
34 <div class="background leftcontainer"></div>
```

Рисунок 50 – Скриншот кода process.php. Запрос в БД

В случае положительного результата запроса, файл производит еще один запрос на информацию о пользователе (ФИО, номер зачетной книжки) и о лабораторной работе.

Далее, файл генерирует разметку для трех разных работ с разными кнопкам и скриптами (функционал процесса демонстрации лабораторной работы «Изучение явления внутреннего трения жидкости» описан в пункте 2.2.3).

После подключения внешних js-файлов и создания блока для отрисовки сцены визуального отображения физического процесса (рисунок 51), генерируются последовательно левое (рисунок 52) и правое меню (рисунок 53), которые отличаются в зависимости от номера лабораторной работы.

Перед созданием правого меню, на странице размещаются пустые поля для заполнения информацией о студенте, лабораторной работе и дате выполнения (рисунок 53, строки 189-194).

Затем размещается нижний блок с кнопками перехода между страницами (рисунок 54).

После создания нижнего контейнера с кнопками перехода, в файле process.php подключаются дополнительные js-файлы с таймером и датой, а также запускается javascript-функция, заполняющая подготовленные пустые поля информацией о студенте и лабораторной работе (рисунок 55).

```
33     } ?>
34     <div class="background_leftcontainer"></div>
35     <div class="container">
36         <div id="MyWebGLApp" class="canv"></div>
37         <script type="text/javascript" src="js/startedFunctionLr3.js"></script>
38         <script src="js/threejs/build/three.js"></script>
39         <script src="https://polyfill.io/v3/polyfill.min.js?features=es6"></script>
40         <script id="MathJax-script" async src="https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-mml-html.js"></script>
41     <div class="interface">
42         <?php if ($id lr==1) {
```

Рисунок 51 – Скриншот кода process.php. Подключение к внешним js-файлам



```

41 <div class="interface">
42 <?php if ($id_lr==1) {
43     ?>
44     <script type="text/javascript" src="js/mainapp<?php echo($id_lr) ?>.js"></script>
45 <div class="leftcontainer">
46     <div class="leftupcontainer">
47         <h1>Выберите шарик и жидкость</h1>
48     <form name="myform" method="post">
49         <input type="button" value="Сталь" name="button" class="input2" onclick="Create1()">
50         
51     <div class="class1">
52         <p>d = 4,5 мм,</p>
53         <p>\(\rho\)= 7,33 \(\frac{r}{cm^3}\)</p>
54     </div>
55 </form>
56 <form name="myform" method="post">
57     <input type="button" value="Пшено" name="button" class="input2" onclick="Create2()">
58     
59     <div class="class1">
60         <p>d = 2 мм,</p>
61         <p>\(\rho\)= 1,316 \(\frac{r}{cm^3}\)</p>
62     </div>
63 </form>
64 <form name="myform" method="post">
65     <input type="button" value="Свинец" name="button" class="input2" onclick="Create3()">
66     
67     <div class="class1">
68         <p>d = 2,5 мм,</p>
69         <p>\(\rho\)= 11,35 \(\frac{r}{cm^3}\)</p>
70     </div>
71 </form>
72 <form name="water1form" method="post">
73     <div class="class1">
74         <p>\(\rho_{x}\)= 0,97 \(\frac{r}{cm^3}\)</p>
75     </div>
76     <input type="button" value="Касторовое масло" name="water1" class="input2" onclick="Createwater1()">
77 </form>
78 <form name="water2form" method="post">
79     <div class="class1">
80         <p>\(\rho_{x}\)= 0,89 \(\frac{r}{cm^3}\)</p>
81     </div>
82     <input type="button" value="Машинное масло" name="water2" class="input2" onclick="Createwater2()">
83 </form>
84 <form name="water3form" method="post">
85     <div class="class1">
86         <p>\(\rho_{x}\)= 1,26 \(\frac{r}{cm^3}\)</p>
87     </div>
88     <input type="button" value="Глицерин" name="water3" class="input2" onclick="Createwater3()">
89 </form>
90 <div class="class1">
91
92 </div>
93
94 </div>
95 <div class="instruction">
96 <div class="recom">
97     <p>Рекомендуемое сочетание:</p>
98     <p>сталь - касторовое масло</p>
99     <p>пшено - машинное масло</p>
100    <p>свинец - глицерин</p>
101 </div>
102 <h1>Инструкция к выполнению</h1>
103 <p>1. Нажмите кнопку "Бросить шарик".</p>
104 <p>2. Включите секундомер при прохождении шариком первой линии.</p>
105 <p>3. Выключите секундомер при достижении шариком второй линии.</p>
106 <p>4. Зафиксируйте время движения.</p>
107 <p>5. Повторите опыт еще 4 раза, фиксируя время движения.</p>
108 </div>
109 </div>
110
111 <?php }
112     if ($id_lr==2) { ?>

```

Рисунок 52 – Скриншот кода process.php. Создание левого меню

```

187 <div id="rtcont" class="rghitcontainer">
188
189 <div class="rghitupcontainer">
190 <h1 id="labname">labname</h1>
191 <p id="studname">studname</p>
192 <p id="studnumber">studnumber</p>
193 <p id="date">date</p>
194 </div>
195
196 <div class="rghitdowncontainer">
197 <?php if ($id_lr==1) { ?>
198 <div class="buttoncontainer">
199 <div class="button">
200 <form name="startProcessForm" method="post">
201 <input name="processstarter" type="button" value="Бросить шарик" class="input2" onclick="clickt()">
202 </form>
203 </div>
204 </div>
205 <div class="timer">
206 <form name=clockform>
207 <div class="timer_button">
208 <input name=starter type=button value="пуск таймера"
209 onclick="findTIME()" class="input2">
210 <input name=clearer type=button value="Сброс"
211 onclick="clearALL()" class="input2">
212 </div>
213 <input name=clock size=10 value="00:00:00.00" class="timerout">
214 </form>
215 </div>
216 </div>
217 <?php }
218 if ($id_lr==2) { ?>
219 <div class="buttoncontainer">

```

Рисунок 53 – Скриншот кода process.php. Создание правого меню

```

254 </div>
255 <div class="downcontainer">
256 <div class="downcontainer_upbutton">
257 <form name="myform" action="lr_page3.php?idlab=<?php echo($id_lr) ?>" method="post" onsubmit="return ChackFields()" >
258 <input type="submit" value="Назад" name="button" class="input2" >
259 </form>
260 <form name="myform" action="endpage.php?idlab=<?php echo($id_lr) ?>&iduser=<?php echo($id) ?>" method="post" onsubmit="return ChackFields()" >
261 <input type="submit" value="Далее" name="button" class="input2" >
262 </form>
263 </div>
264 <div class="downcontainer_downbutton">
265 <form name="myform" action="authorization.php" method="post" onsubmit="return ChackFields()" >
266 <input type="submit" value="На главную" name="button" class="input2" >
267 </form>
268 </div>
269 </div>
270 </div>
271 <script type="text/javascript" src="js/date.js"></script>

```

Рисунок 54 – Скриншот кода process.php. Создание нижнего меню

Произведя все измерения, пользователь нажимает на кнопку «Далее», которая перенаправляет его на страницу подготовки отчета о выполнении лабораторной работы.

Код страницы (рисунок 56) содержится в файле endpage.php, который представляет собой html-разметку в стиле всех предыдущих страниц и состоит из текстового блока с порядком подготовки отчета, зависящим от номера лабораторной работы и нижним блоком кнопок, для перехода между страницами.

```

270 </script>
271 <script type="text/javascript" src="js/date.js"></script>
272 <script type="text/javascript">
273 function ChackFields() {
274     return true;
275 }
276 document.getElementById("labname").innerHTML = "<?php echo ($labname)?>";
277 document.getElementById("studname").innerHTML = "<?php echo ($surname)?> "+"<?php echo ($name)?> "+"<?php echo ($phathername)?>";
278 document.getElementById("date").innerHTML = date_time();
279 document.getElementById("studnumber").innerHTML = "<?php echo ($password)?>";
280 </script>
281
282 <script type="text/javascript" src="js/timer.js"></script>
283 <?php }
284 else {
285     ?>
286     <p>Ошибка авторизации, попробуйте <a href="index.php">войти</a> заново</p>
287 <?php } ?>
288
289 </body>
290 </html>
291

```

Рисунок 55 – Скриншот кода process.php. Заполнение полей о студенте

```

endpage.php
1 <?php session_start(); ?>
2 <!doctype html>
3 <html>
4 <head>
5 <meta charset="UTF-8">
6 <title>Завершение работ</title>
7 <link rel="stylesheet" type="text/css" href="reset.css">
8 <link rel="stylesheet" type="text/css" href="styles.css">
9 </head>
10 <script type="text/javascript" src="js/date.js"></script>
11 <script type="text/javascript">
12     function ChackFields2(){ return true; }
13 </script>
14 <?php $p1 = $_SESSION['sname'];
15 $p2 = $_SESSION['pass'];
16 $id_lr = $_GET['idlab'];
17 $iduser = $_GET['iduser'];
18 $link = mysqli_connect('localhost', 'root', '', 'virtual_lab');
19 $query0 = mysqli_query($link, "SET NAMES 'utf8mb4'" ) or die("Ошибка " . mysqli_error($link));
20 $query1 = mysqli_query($link, "SELECT * FROM lab WHERE id_lab = '$id_lr'" );
21 while ($row = mysqli_fetch_array($query1)){ $name_lab = $row['name_lab']; } ?>
22 <body>
23 <h1 class="h1">Подготовка отчета по лабораторной работе "<?php echo ($name_lab) ?>"</h1>
24 <div class="content_box">
25     <div class="info_container">
26         <div class="info">
27             <p><?>1. Внимате в отчет данные о себе, дате выполнения работ</p>
28             <p><?>2. Заполните пустые поля отчета</p>
29             <?php if ($id_lr==1) { ?>
30                 <p><?>3. Внесите данные эксперимента в таблицу 1</p>
31                 <p><?>4. Выполните расчет коэффициента вязкости</p>
32                 <p><?>5. Оцените погрешность измерений</p>
33                 <p><?>6. Проанализируйте результаты и напишите вывод</p>
34                 <p><?>7. Отчет по лабораторной работе прикрепите в ответ на задание в <a href="https://e.sfu-kras.ru/mod/assign/view.php?id=394895">е-Курсе</a> дисциплины "физика"</p>
35             <?php }
36             if ($id_lr==2) { ?>
37                 <p><?>3. Внесите данные эксперимента в таблицу 1</p>
38                 <p><?>4. Выполните расчет длины волны</p>
39                 <p><?>5. Оцените погрешность измерений</p>
40                 <p><?>6. Проанализируйте результаты и напишите вывод</p>
41                 <p><?>7. Отчет по лабораторной работе прикрепите в ответ на задание в <a href="https://e.sfu-kras.ru/mod/assign/view.php?id=673865">е-Курсе</a> дисциплины "физика"</p>
42             <?php }
43             if ($id_lr==3) { ?>
44                 <p><?>3. Внесите данные эксперимента в таблицы</p>
45                 <p><?>4. Выполните расчет скорости пулек</p>
46                 <p><?>5. Выполните расчет энергии пули, энергии ядзника с пулей и величину энергии перешедшей во внутреннюю энергию</p>
47                 <p><?>6. Проанализируйте результаты и напишите вывод</p>
48                 <p><?>7. Отчет по лабораторной работе прикрепите в ответ на задание в <a href="https://e.sfu-kras.ru/mod/assign/view.php?id=394549">е-Курсе</a> дисциплины "физика"</p>
49             <?php } ?>
50         </div>
51     </div>
52 </div>
53 <div class="downcontainer">
54     <div class="downcontainer_upbutton_endpage">
55         <form name="myform" action="process.php?idlab=<?php echo ($id_lr) ?>" method="post" onsubmit="return ChackFields()" >
56             <input type="submit" value="Назад" name="button" class="input2" >
57         </form>
58         <form name="myform" action="add_data.php?action-end&idlab=<?php echo ($id_lr) ?>&iduser=<?php echo ($iduser) ?>" method="post" onsubmit="return ChackFields()" >
59             <input id="i5" type="text" name="date" placeholder="номер" class="input3" value="" >
60             <input type="submit" value="Завершить выполнение" name="button" class="input2" >
61         </form>
62     </div>
63     <div class="downcontainer_downbutton_endpage">
64         <form name="myform" action="authorization.php" method="post" onsubmit="return ChackFields()" >
65             <input type="submit" value="На главную" name="button" class="input2" >
66         </form>
67     </div>
68 </div>
69 <script type="text/javascript">
70     document.getElementById("i5").value = date_time();
71 </script>
72 </body>
73 </html>

```

Рисунок 56 – Скриншот кода endpage.php

Страница получает данные о номере лабораторной работы и номере студента методом get от предыдущей страницы, производит запрос в базу данных на наименование лабораторной работы для вывода в заголовок h1,

после которого генерирует текстовый блок, с необходимым для данной работы порядком выполнения. Нижний блок кнопок перехода перенаправляет пользователя на предыдущую страницу, на главную страницу, а также, при нажатии на кнопку «Завершить выполнение» отправляются данные о номере лабораторной работы и номере студента методом get и дате выполнения (скрипт заполнения поля – строка 70, рисунок 56) методом post в php-файл add\_data, для дальнейшей обработки.

Файл add\_data представляет собой php-код (рисунок 57) обработки полученных данных из двух файлов – userreg (форма заполнения данных регистрации) и endpage (страница подготовки отчета о выполнении лабораторной работы), каждый из которых методом get заполняет переменную action отличающимся значением.

```

add_data.php
1 <?php session_start(); ?>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="UTF-8">
6 <title>Сообщение о регистрации</title>
7 <link rel="stylesheet" type="text/css" href="reset.css">
8 <link rel="stylesheet" type="text/css" href="styles.css">
9 </head>
10 <script type="text/javascript">
11     function checkFields2(){ return true; }
12 </script>
13 <div class="form">
14 <?php
15 $action = $_GET["action"];
16 $p1 = $_SESSION['sname'];
17 $p4 = $_SESSION['pass'];
18 if ($p1==null || $p4==null) {
19     $p1 = $_POST['surname'];
20     $p4 = $_POST['password'];
21     $_SESSION['sname'] = $p1;
22     $_SESSION['pass'] = $p4;
23 }
24 $p5 = "student";
25 $link = mysqli_connect('localhost', 'root', '', 'virtual_lab');
26 $query0 = mysqli_query($link, "SET NAMES 'utf8mb4'" or die("Ошибка " . mysqli_error($link));
27 $query1 = mysqli_query($link, "SELECT * FROM userdata WHERE (sname=\"$p1\" AND (pass=\"$p4\"));");
28 if (mysqli_num_rows($query1) != 0) {
29     if ($action=="reg") {
30         ?> <p>Ошибка регистрации. Пользователь с такой фамилией и паролем уже существует</p><?php
31     }
32     else {
33         $idlab = $_GET["idlab"];
34         $iduser = $_GET["iduser"];
35         $date = $_POST["date"];
36         $query2 = mysqli_query($link, "SELECT * FROM interfaceout WHERE (id_userdata=\"$iduser\" AND id_lab=\"$idlab\"");");
37         if (mysqli_num_rows($query2) != 0) {
38             $query3 = mysqli_query($link, "UPDATE `interfaceout` SET `date`=\"$date\" WHERE id_userdata=\"$iduser\" AND id_lab=\"$idlab\"");");
39         }
40         else {
41             $query3 = mysqli_query($link, "INSERT INTO interfaceout (`id_userdata`, `id_lab`, `date`)
42             VALUES(\"$iduser\", \"$idlab\", \"$date\")");");
43         }
44         echo "данные о выполнении записаны". mysqli_error($link);
45     }
46 }
47 else {
48     if ($action=="end") {
49         ?> <p>Ошибка авторизации, попробуйте <a href="index.php">войти</a> заново</p><?php
50     }
51     else {
52         $p2 = $_POST["name"];
53         $p3 = $_POST["fathername"];
54         $query3 = mysqli_query($link, "INSERT INTO userdata (`sname`, `name`, `tname`, `pass`, `account_name`)
55         VALUES(\"$p1\", \"$p2\", \"$p3\", \"$p4\", \"$p5\")");");
56         echo "Регистрация прошла успешно". mysqli_error($link);
57     }
58 }
59 ?>
60 <form name="myform" action="authorization.php" method="post" onsubmit="return checkFields2()" >
61     <input type="submit" value="На главную" name="button" class="input2" >
62 </form>
63 </div>
64 </html>

```

Рисунок 57 – Скриншот кода add\_data.php

Файл подключается к текущей сессии, заполняет переменные о логине и пароле либо из сессии (если страница открылась из `endpage`, то сессия хранит эти данные), либо методом `post` (если страница открылась из `userreg`, то в сессии нет логина и пароля). После этого создается подключение к базе данных и производится запрос в таблицу `userdata` на наличие пользователя с таким логином и паролем. Если результат запроса не отрицательный, то дальнейший процесс зависит от того, какая страница обратилась к этому файлу страница регистрации или страница завершения работы.

В первом случае, файл выводит на страницу сообщение об ошибке регистрации, так как такой пользователь уже есть в бд. Во втором случае, заполняются переменные о номере пользователя, номере лабораторной работе и дате выполнения (данными полученными методом `get`), и производится запрос в таблицу `interfaceout` на наличие записи с такими данными. Если результат запроса не отрицательный, то производится запрос на редактирование этой записи с изменением даты, в противном случае – добавляется новая запись в таблицу `interfaceout`, после чего выводится сообщение об успешном добавлении записи выполнения лабораторной работы.

Если результат запрос в таблицу `userdata` на наличие пользователя с текущим логином и паролем отрицательный, то дальнейший процесс так же зависит от обратившейся страницы к этому файлу.

В случае, если это была страница завершения работы, то выводится сообщение об ошибке авторизации, так как неавторизованный пользователь, не может выполнять лабораторные работы. В случае страницы регистрации – в переменные записываются данные об имени и отчестве (полученные методом `post` от `userreg`) и производится запрос на добавление записи в таблицу `userdata`, в которую записываются ФИО, номер зачетной книжки и тип аккаунта, по умолчанию это студент.

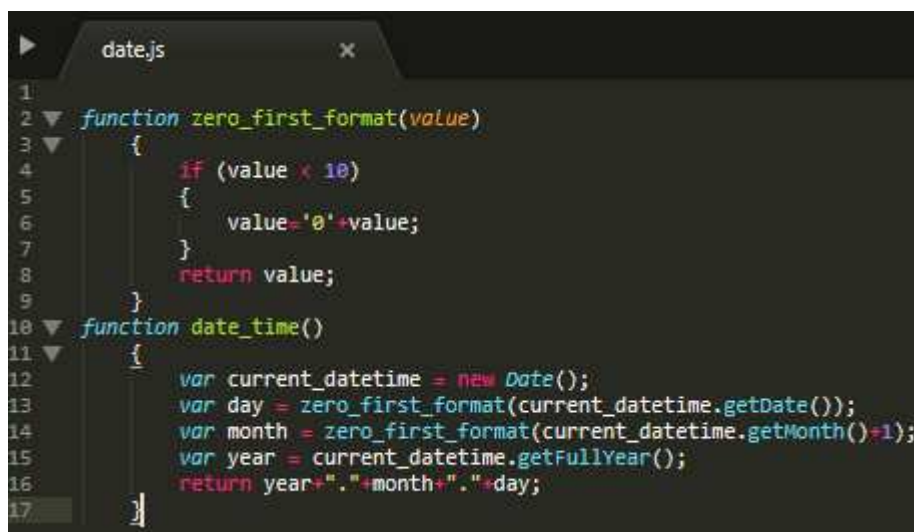
После этого на страницу добавляется кнопка перехода в меню выбора лабораторных «На главную».

### 2.2.3 Создание имитационной модели физического процесса лабораторной работы «Изучение внутреннего трения жидкости»

Для работы веб-приложения были созданы javascript-функции, отвечающие за разные элементы функционала, расположенные в разных файлах с расширением .js, для удобства проектирования.

При выполнении пользователем первой лабораторной работы, файл process.php подключает следующие js-файлы: date, timer, mainapp1. Кроме этого, для выполнения всех работ необходимо подключить библиотеку three.js, файл которой расположен в предварительно скаченной папке threejs. Код подключения этих файлов показан ранее.

Код date.js показан на рисунке 58 и представлен в виде двух функций, преобразующих текущие дату и время компьютера пользователя в дату нужного формата.



```
1
2 function zero_first_format(value)
3 {
4     if (value < 10)
5     {
6         value = '0'+value;
7     }
8     return value;
9 }
10 function date_time()
11 {
12     var current_datetime = new Date();
13     var day = zero_first_format(current_datetime.getDate());
14     var month = zero_first_format(current_datetime.getMonth()+1);
15     var year = current_datetime.getFullYear();
16     return year+"."+month+"."+day;
17 }
```

Рисунок 58 – Скриншот кода файла date.js

Код timer.js показан на рисунке 59 и представлен в виде функций, которые запускаются нажатием на кнопки пользователем и преобразующие текущее время компьютера в формат таймера, отсчитывающего миллисекунды,

секунды, минуты от нуля.

```
timer.js
1 var base = 60;
2 var clocktimer, dateObj, dh, dm, ds, ms;
3 var readout = '';
4 var h = 1;
5 var m = 1;
6 var tm = 1;
7 var s = 0;
8 var ts = 0;
9 var ms = 0;
10 var show = true;
11 var init = 0;
12 var ii = 0;
13 function clearALL() {
14   clearTimeout(clocktimer);
15   h = 1; m = 1; tm = 1; s = 0; ts = 0; ms = 0;
16   init = 0; show = true;
17   readout = '00:00:00.00';
18   document.clockform.clock.value = readout;
19   document.clockform.starter.value = "Пуск таймера";
20   var CF = document.clockform;
21   ii = 0;
22   function startTIME() {
23     var cdateObj = new Date();
24     var t = (cdateObj.getTime() - dateObj.getTime()) / (s * 1000);
25     if (t > 999) { s++; }
26     if (s >= (m * base)) { ts = 0;
27       m++; } else {
28       ts = parseInt((ms / 100) + s);
29       if (ts >= base) { ts = ts - ((m - 1) * base); }
30       if (m >= (h * base)) { tm = 1;
31         h++; } else {
32         tm = parseInt((ms / 100) + m);
33         if (tm >= base) { tm = tm - ((h - 1) * base); }
34         ms = Math.round(t * 100);
35         if (ms > 99) { ms = 0; }
36         if (ms == 0) { ms = '00'; }
37         if (ms > 0 && ms < 9) { ms = '0' + ms; }
38         if (ts > 0) { ds = ts; if (ts < 10) { ds = '0' + ts; } } else { ds = '00'; }
39         dm = tm - 1;
40         if (dm > 0) { if (dm < 10) { dm = '0' + dm; } } else { dm = '00'; }
41         dh = h - 1;
42         if (dh > 0) { if (dh < 10) { dh = '0' + dh; } } else { dh = '00'; }
43         readout = dh + ':' + dm + ':' + ds + '.' + ms;
44         if (show == true) { document.clockform.clock.value = readout; }
45         clocktimer = setTimeout("startTIME()", 1);
46         function findTIME() {
47           if (init == 0) { dateObj = new Date();
48             startTIME();
49             init = 1;
50             document.clockform.starter.value = "Стоп";
51           } else { if (show == true) {
52             show = false;
53             document.clockform.starter.value = "Пуск таймера";
54           } else { show = true;
55             document.clockform.starter.value = "Стоп";
56           } } }

```

Рисунок 59 – Скриншот кода файла timer.js

Сам таймер, расположенный на странице, представляет собой поле input стилизованное под часовое табло и кнопки старта и сброса, запускающие функции startTIME, findTIME, clearALL (рисунок 60).

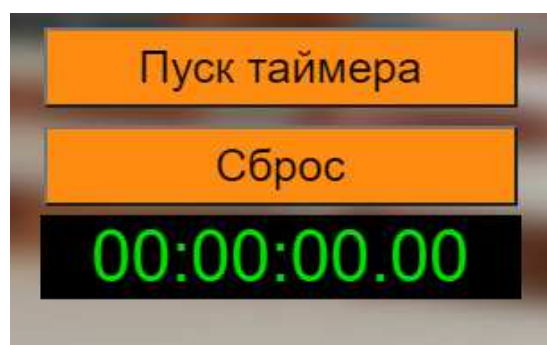


Рисунок 60 – Скриншот части страницы с секундомером и кнопками управления

Код файла mainapp.js (рисунки 63-68) состоит из набора функций, которые отрисовывают сцену с 3D объектами, в зависимости от условий, действующих на эти объекты. Условия определяются пользователем или самими объектами, если запущен какой-то процесс.

Для виртуальной демонстрации физических процессов лабораторных работ была выбрана технология отображения и анимации в виде 3D объектов со статичной камерой. Для реализации этого была использована библиотека threejs, упрощающая сложные элементы обработки 3D графики.

Физический процесс лабораторной работы «Изучение внутреннего трения жидкости» представляет собой падающий шарик в сосуд с жидкостью. У пользователя есть возможность выбрать материал, из которого сделан шарик (сталь, пшено, свинец), и жидкость, в которой он будет падать (касторовое масло, машинное масло, глицерин). На рисунке 61 показано меню выбора исходных параметров лабораторной работы.

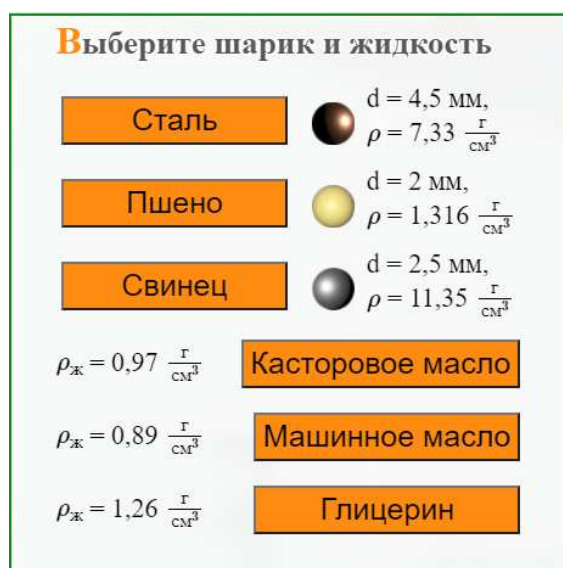


Рисунок 61 – Скриншот фрагмента страницы с меню выбора исходных физических данных



Виртуальная установка представлена в виде некоторой поверхности (стола), с размещенными на ней стеклянным сосудом, линейкой и штативом, для сброса шарика (рисунок 62).

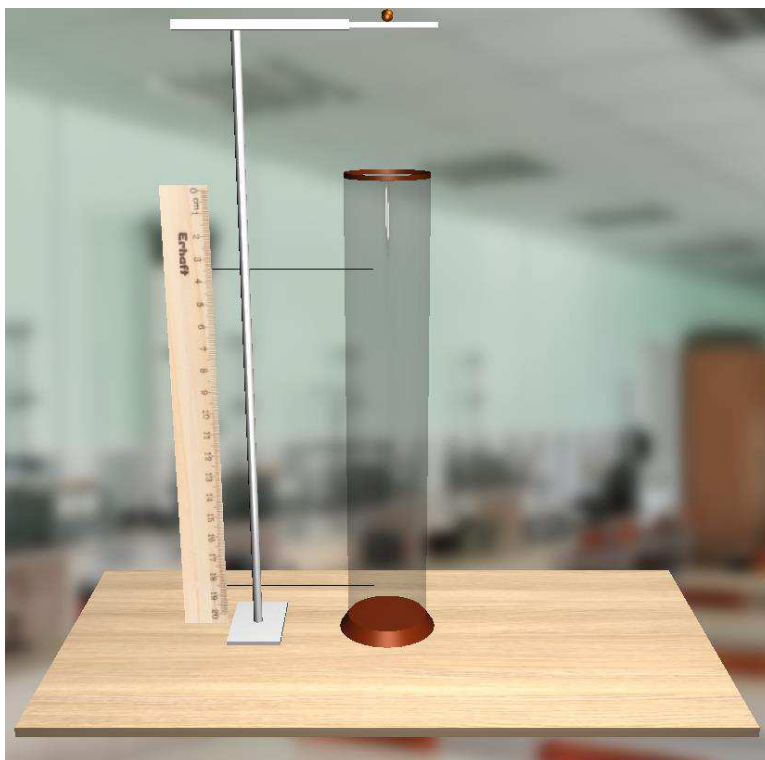


Рисунок 62 – Установка лабораторной работы 1

Для реализации этой установки в файле `mainapp1.js` с помощью функций `three.js`, а также специально разработанных `javascript`-функций была создана виртуальная сцена, на которую добавлены соответствующие объекты. Перед созданием сцены были объявлены необходимые для реализации переменные, код объявления которых показан на рисунке 63.

```

1  var scene;
2  var camera;
3  var controls;
4  var light;
5  var renderer, container;
6  var colorWater;
7
8  var Plane, Sphere, Cylinder, Line, positionSphere, positionSphere2;
9  var V=0;
10 var a;
11 var b;
12 var Uconst;
13 var rhoWat, rhoSph;
14 var Fa;
15 var Ft;
16 var Fc;
17 var flag=0;
18 var flag_water=0;
19 var flag_add_water=0;
20 var flag_remove_water=1;
21 var flag_create_sphere=0;
22 var flag_sht=0;
23
24  init();

```

Рисунок 63 – Код объявления глобальных переменных

Кроме переменных, была запущена функция `init`, которая состоит из функций создания сцены, объектов и анимации (рисунок 64).

```

26  function init(){
27
28      CreateScene();
29      CreateObjects();
30      update();
31  }
32

```

Рисунок 64 Код функции `init`

Функция `CreateScene`, код которой показан на рисунке 65, создает новую сцену, а также добавляет на нее свет и камеру.

```

33  function CreateScene(){
34      scene = new THREE.Scene();
35
36      camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 1, 10000 );
37      camera.position.set(0, 40, 130);
38      camera.rotation.set(-0.3, 0, 0);
39
40      light = new THREE.PointLight( 0xffffff);
41      lightup = new THREE.PointLight( 0xffffff, 0.8);
42      light.position.set( 5, 0, 25);
43      lightup.position.set( 53, 130, -80);
44
45      scene.add(camera);
46      scene.add(light);
47      scene.add(lightup);
48
49
50      renderer = new THREE.WebGLRenderer({ alpha: true });
51      renderer.setClearColor(0x000000, 0);
52      DivOut = document.getElementById("rtcont")
53
54      renderer.setSize(window.innerWidth, window.innerHeight);
55
56      container = document.getElementById("MyWebGLApp");
57      container.appendChild( renderer.domElement );
58      window.addEventListener('resize', onWindowResize, false );
59  }
60

```

Рисунок 65 – Код функции CreateScene

С помощью объектов библиотеки three.js THREE.Scene(), THREE.PerspectiveCamera(), THREE.PointLight(), с необходимыми параметрами были созданы новая сцена, перспективная камера и два точечных источника света. С помощью функций position.set и rotation.set в виде координат установлено положение камеры, света на сцене, а с помощью функции add, они были добавлены на сцену. Для отображения сцены на веб странице был использован объект THREE.WebGLRenderer(), который добавил объект визуализации сцены и ее объектов и создал полотно canvas в подготовленном блоке (блок div с идентификатором rtcont) веб-страницы process. Далее указаны размеры объекта (во все окно браузера) и запущена функция onWindowResize (код на рисунке 66), позволяющая изменять размеры объекта визуализации, если размеры окна браузера изменились.

```
61 function onWindowResize()  
62 {  
63     camera.aspect = window.innerWidth / window.innerHeight;  
64     camera.updateProjectionMatrix();  
65     renderer.setSize(window.innerWidth, window.innerHeight);  
66 }
```

Рисунок 66 – Код функции onWindowResize

После создания сцены с камерой и светом с помощью функции CreateObjects были созданы объекты, из которых состоят элементы установки виртуальной лабораторной работы. Для создания объекта в threejs необходимо задать для него геометрию и материал используя специальные объекты THREE.[тип геометрии]Geometry() и THREE.Mesh[тип материала]Material({}). В скобках указываются параметры такие как, размеры, цвет и т.д. На рисунке 37 показан код создания геометрии и материалов объектов, используемых в этой лабораторной работе.

Некоторым объектам вместо цвета была добавлена текстура, предварительно добавленная в папку image и загруженная специальным загрузчиком THREE.ImageUtils.loadTexture() библиотеки three.js. Материалу, который используется для создания стеклянного сосуда было присвоено свойство отображения объекта с внешней и внутренней стороны (строки 104, 105 на рисунке 67).

```

69 var geometryPlane = new THREE.PlaneGeometry( 20, 1);
70 var geometryLineAB = new THREE.PlaneGeometry( 34, 0.2);
71 var geometryLine = new THREE.PlaneGeometry( 10, 100, 0);
72 var geometrySphere = new THREE.SphereGeometry(1.5);
73 var geometryCylinder = new THREE.CylinderGeometry(9, 9, 100, 24, 1, true);
74 var geometryCylinderDwn = new THREE.CylinderGeometry(9, 11, 3, 24);
75 var geometryCylinderUp = new THREE.CylinderGeometry(9, 9, 1, 24, 1, true);
76
77 var geometryBox = new THREE.BoxGeometry(150, 2, 75);
78 var geometryShtPlan = new THREE.BoxGeometry(12, 1, 20);
79 var geometryShtCylinder = new THREE.CylinderGeometry(1, 1, 130, 16);
80 var geometryShtBoxOut = new THREE.BoxGeometry(35, 2, 4);
81 var geometryShtBoxIn = new THREE.BoxGeometry(20, 1, 3);
82
83
84 var TextureLine = new THREE.ImageUtils.loadTexture( 'image/1.jpg' );
85 var TextureBox = new THREE.ImageUtils.loadTexture( 'image/textureTableLr1.jpeg' );
86 var materialLine = new THREE.MeshBasicMaterial( { map: TextureLine } );
87 var materialBox = new THREE.MeshPhongMaterial( { map: TextureBox } );
88 var materialLineAB = new THREE.MeshBasicMaterial({color: 0x000000});
89 var materialSphere = new THREE.MeshPhongMaterial({color: 0x90ee90, specular: 0xcbcbcb});
90 var materialCylinderBorder = new THREE.MeshPhongMaterial({color: 0x61210b});
91
92 var materialCylinder = new THREE.MeshPhysicalMaterial({
93   color: 0x2e2e2e,
94   metalness: 0,
95   roughness: 0,
96   envMapIntensity: 1,
97   alphaTest: 0.5,
98   depthWrite: false,
99   transmission: 0.85, // use material.transmission for glass materials
100  opacity: 1, // set material.opacity to 1 when material.transmission is non-zero
101  transparent: true
102  });
103 var materialSht = new THREE.MeshPhongMaterial({color: 0xb5b5b5, specular: 0xcbcbcb});
104 materialCylinderBorder.side = THREE.DoubleSide;
105 materialCylinder.side = THREE.DoubleSide;

```

Рисунок 67 – Код создания геометрии и материалов объектов

Используя созданные геометрию объектов и материалы с помощью объекта THREE.Mesh() создаются новые 3D объекты, которые использованы в лабораторной работе. Код показан на рисунке 68.

```

107 Plane = new THREE.Mesh( geometryPlane, materialSphere );
108 Sphere = new THREE.Mesh( geometrySphere, materialSphere );
109 Line = new THREE.Mesh( geometryLine, materialLine );
110 Cylinder = new THREE.Mesh( geometryCylinder, materialCylinder );
111 CylinderUp = new THREE.Mesh( geometryCylinderUp, materialCylinderBorder );
112 CylinderDwn = new THREE.Mesh( geometryCylinderDwn, materialCylinderBorder );
113 LineA = new THREE.Mesh( geometryLineAB, materialLineAB );
114 LineB = new THREE.Mesh( geometryLineAB, materialLineAB );
115 Box = new THREE.Mesh( geometryBox, materialBox );
116
117 ShtPlan = new THREE.Mesh( geometryShtPlan, materialSht );
118 ShtCylinder = new THREE.Mesh( geometryShtCylinder, materialSht );
119 ShtBoxOut = new THREE.Mesh( geometryShtBoxOut, materialSht );
120 ShtBoxIn = new THREE.Mesh( geometryShtBoxIn, materialSht );

```

Рисунок 68 – Код создания 3D объектов

Далее, с помощью функции `position.set` объектам назначаются координаты, на которых они будут находиться, а функцией `add` они добавляются на созданную сцену (рисунок 69).

```
122 Line.position.set(-42, -30, -100);
123 LineA.position.set(-20, 2.5, -100);
124 LineB.position.set(-20, -70.5, -100);
125 Plane.position.set(0, 50, -100);
126 Sphere.position.set(0, 53, -100);
127 Cylinder.position.set(0, -28, -100);
128 CylinderUp.position.set(0, 22, -100);
129 CylinderDwn.position.set(0, -79, -100);
130 Box.position.set(0, -85, -100);
131
132 ShtPlan.position.set(-30, -80, -100);
133 ShtCylinder.position.set(-30, -14.5, -100);
134 ShtBoxOut.position.set(-25, 51, -100);
135 ShtBoxIn.position.set(0, 51, -100);
136
137 scene.add(Line);
138 scene.add(LineA);
139 scene.add(LineB);
140
141 scene.add(Cylinder);
142 scene.add(CylinderUp);
143 scene.add(CylinderDwn);
144 scene.add(Box);
145
146 scene.add(ShtPlan);
147 scene.add(ShtCylinder);
148 scene.add(ShtBoxOut);
149 scene.add(ShtBoxIn);
```

Рисунок 69 – Код добавления созданных 3Dобъектов на сцену

Все размеры 3D объектов и координаты их расположения на сцене вычислены заранее, также как и подобраны материалы, максимально приближенные к материалам реальных объектов.

Функция `CreateObjects` добавила на сцену стол, штатив и стеклянный цилиндр (сосуд). Поскольку тип шарика и жидкость выбираются пользователем, то были добавлены функции (`Create1`, `Create2` и `Create3` для шариков, `CreateWater1`, `CreateWater2` и `CreateWater3` для жидкости), создающие эти 3Dобъекты в зависимости от выбора пользователя.

Скриншоты кода функций `Create1` и `CreateWater1` показаны на рисунках 70 и 71, у других функций меняются только параметры объектов (цвет, размеры, коэффициенты для физического процесса).

```

150 }
151 function create1(){
152     if (flag==0) {
153         if (flag_create_sphere!=1) {
154             flag_create_sphere=1;
155             scene.remove(Sphere);
156             var geometrySphere1 = new THREE.SphereGeometry(1.2);
157             var materialSphere1 = new THREE.MeshStandardMaterial({
158                 color: 0xff8000,
159                 roughness: 0.45,
160                 metalness: 0.8});
161             Sphere = new THREE.Mesh( geometrySphere1, materialSphere1 );
162             Sphere.position.set(0, 52.7, -100);
163             positionSphere=Sphere.position.y;
164             scene.add(Sphere);
165             Ft=6.27;
166             rhoSph=0.5;
167         } else{
168             flag_create_sphere=0;
169             scene.remove(Sphere);
170         }
171     }
172 }
173

```

Рисунок 70 – Код создания одного шарика

```

219 function Createwater1(){
220
221     if (flag_water==0) {
222         flag_water=1;
223         Createwater();
224         Fc=5.72;
225         rhoWat=0.9;
226         document.water1form.water1.value="Опустошить";
227         document.water2form.water2.value="Опустошить";
228         document.water3form.water3.value="Опустошить";
229     } else {
230         if (Sphere.position.y!=positionSphere){
231             flag_create_sphere=0;
232             scene.remove(Sphere);
233         }
234         flag_water=0;
235         document.water1form.water1.value="Касторовое масло";
236         document.water2form.water2.value="Машинное масло";
237         document.water3form.water3.value="Глицерин";
238     }
239 }

```

Рисунок 71 – Код создания одной жидкости

Процесс создания шарика достаточно простой, функция проверяет, чтобы была не нажата кнопка, запускающая опускание шарика в жидкость, проверяет, чтобы выбранный шарик не был уже создан (если он уже был создан, он удаляется), удаляет со сцены старый шарик, используя функцию `remove`, и создает новый шарик с геометрией и материалом, предназначенными для этого типа шарика (на рисунке 70 представлена функция создания стального шарика), после чего, объект добавляется на сцену.

Процесс создания жидкости сложнее процесса создания шарика, поскольку здесь реализована анимация наполнения сосуда, а также, меняются наименования кнопок с названиями жидкостей, после их нажатия. Для

реализации этого был использован механизм из нескольких функций, одна из которых постоянно вызывается (примерно 60 раз в секунду) функцией анимации update.

После вызова функции CreateWater1 кнопкой проверяется, чтобы функция уже не была вызвана ранее, потому что, если функция вызывалась, то процесс наполнения заменяется процессом опустошения (за это отвечает переменная flag\_water, которая принимает значения от 0 до 3, где 0 – это отсутствие жидкости в сосуде, а 1, 2, 3 – номера жидкостей). На рисунке 72 виртуальная лабораторная установка с жидкостью и без нее.

Если эта функция CreateWater1 не вызывалась, то устанавливается текущий номер жидкости в переменную, параметры анимации физического процесса и вызывается функция CreateWater (код показан на рисунке 73).

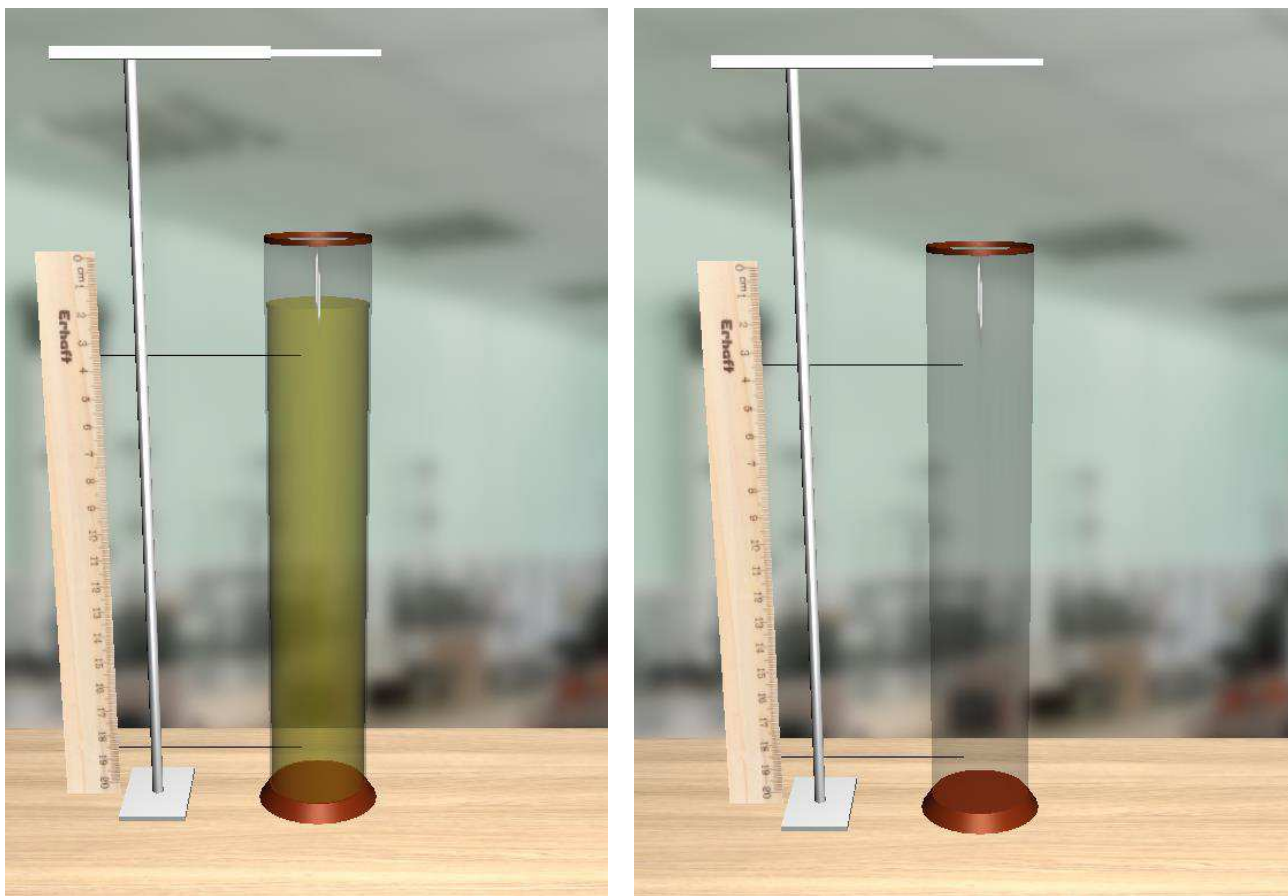


Рисунок 72 – Скриншот виртуальной лабораторной установки с разным состоянием наполнения сосуда



```

287 function CreateWater(){
288     if (flag_add_water==0) {
289         if (flag_water==1) {
290             colorWater=0xb3a80f;
291         }
292         if (flag_water==2) {
293             colorWater=0xec0707;
294         }
295         if (flag_water==3) {
296             colorWater=0xee5b0;
297         }
298         var geometryWater = new THREE.CylinderGeometry(8.6, 8.6, 1, 24);
299         var materialWater = new THREE.MeshPhysicalMaterial({
300             color: colorWater,
301             metalness: 0,
302             roughness: 0,
303             envMapIntensity: 1,
304             alphaTest: 0.5,
305             depthWrite: false,
306             transmission: 0.65, // use material.transmission for glass materials
307             opacity: 1, // set material.opacity to 1 when material.transmission is used
308             transparent: true});
309         materialWater.side = THREE.DoubleSide;
310         water = new THREE.Mesh(geometryWater, materialWater);
311         water.position.set(0, -77.5, -100);
312         scene.add(water);
313         flag_add_water=1;
314     }
315 }
316
317

```

Рисунок 73 – Код функции CreateWater

Данная функция создает новый 3D объект с геометрией и материалом, определенными номером жидкости (в зависимости от жидкости меняется ее цвет). Для имитации наполнения сосуда, используется увеличение высоты объекта (жидкости), поэтому функция CreateWater создала объект с высотой равной 0. Для отображения процесса наполнения сосуда была создана функция addWater (рисунок 74), которая будет регулярно вызываться функцией анимации update и при каждом вызове изменять высоту цилиндра (жидкости), параллельно проверяя текущее состояние жидкости.

```

319 function addwater(){
320     if (flag_water!=0) {
321         if (water.scale.y < 90) {
322             water.scale.y += 1;
323             water.position.y += 0.5;
324         } else {
325             water.scale.y = 90;
326             water.position.y = -33.5;
327         }
328     } else {
329         if (water.scale.y > 0.1) {
330             water.scale.y -= 1;
331             water.position.y += -0.5;
332         } else {
333             scene.remove(water);
334             flag_add_water=0;
335         }
336     }
337 }
338

```

Рисунок 74 – Код функции addWater

Если была вызвана функция создания какой-либо жидкости (например CreateWater1), то она увеличивает высоту цилиндра до определенного значения, после чего фиксирует это значение. Если эта функция была вызвана, но жидкость уже добавлена (в этом случае flag\_water принимает значение 0), то функция addWater уменьшает высоту цилиндра жидкости до минимально возможного, после чего удаляет цилиндр со сцены.

Для запуска процесса падения шарика используется кнопка «Бросить шарик», которая запускает функцию clickt (рисунок 75).

```
339 function clickt(){
340
341     if (flag==0) {
342         flag=1;
343         Uconst=Ft-Fc;
344         document.startProcessForm.processStarter.value="Повторить";
345     } else {
346         flag=0;
347         document.startProcessForm.processStarter.value="Бросить шарик";
348     }
349 }
```

Рисунок 75 – Код функции clickt

Эта функция меняет значение глобальной переменной flagи значение наименования кнопки. Так же, функция записывает в переменную Uconst параметр скорости, вычисляемый из параметров жидкости и шарика, падающего в эту жидкость.

Эта переменная используется в имитации падения шарика в жидкость, которую показывает постоянно вызываемая анимацией функция izm. Код этой функции показан на рисунке 76.

Так как эта функция регулярно вызывается, то основной ее код начинает работать только после нажатия на кнопку «Бросить шарик». В случае нажатия на эту кнопку переменная flag принимает значение 1, после чего начинается движение элемент штатива. После достижения этим элементом значения, когда шарик может падать, элемент штатива фиксируется и начинается процесс

падения шарика вниз с ускорением (свободное падение шарика), который зависит от следующих факторов: достиг ли шарик жидкости (условие строки 357 рисунок 76); есть ли в сосуде жидкость (условие строки 360 рисунок 76); замедление шарика в жидкости до постоянной скорости  $U_{const}$ , вычисленной ранее из параметров падающего шарика и жидкости, содержащейся в сосуде; достиг ли шарик дна сосуда (условие на строках 368-374 рисунка 76).

```

351 function izm(){
352     if (flag==1){
353         ShtBoxIn.position.x += -0.7;
354     }
355     if (ShtBoxIn.position.x < -12) {
356         ShtBoxIn.position.x = -13;
357         if (Sphere.position.y > 10) {
358             V += a;
359         }
360         else if (flag_water!=0){
361             if (V>Uconst){
362                 V += -Fc;
363                 positionSphere2=Sphere.position.y;
364             } else {
365                 V=Uconst;
366             }
367         }
368         if (Sphere.position.y > -748&&V=0) {
369             Sphere.position.y += -V;
370             positionSphere2=Sphere.position.y;
371         }
372         else{
373             Sphere.position.y = positionSphere2;
374         }
375     }
376     if (flag==0) {
377         ShtBoxIn.position.x = 0;
378         Sphere.position.y = positionSphere;
379         V=0;
380         a=0.16;
381         b=1;
382     }
383 }
384

```

Рисунок 76 – Код функции izm

В случае повторного нажатия на кнопку «Бросить шарик» (ставшая кнопкой «Повторить» после первого нажатия), переменная flag принимает значение 0, элемент штатива и шарик устанавливаются на начальные позиции, все измененные динамические параметры принимают начальные значения (строки 376-383 рисунка 76), после чего пользователь может повторить опыт и запустить падение шарика снова.

На рисунке 77 представлен вид установки с различными положениями

шарика.

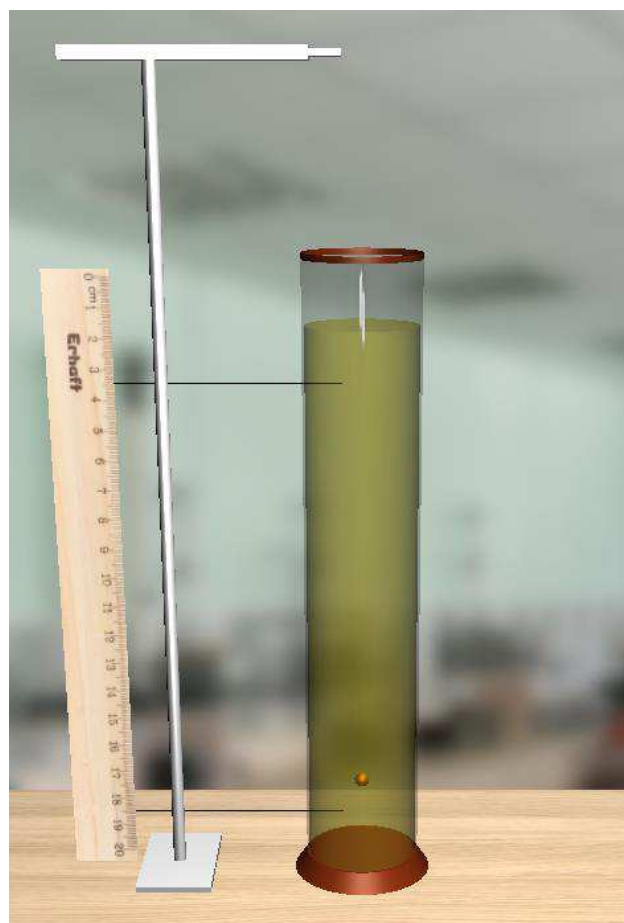
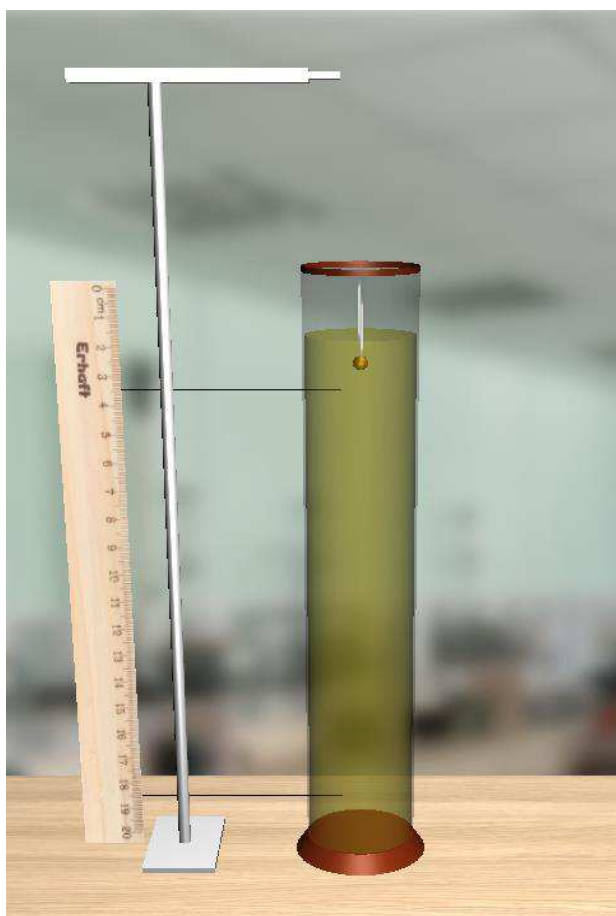
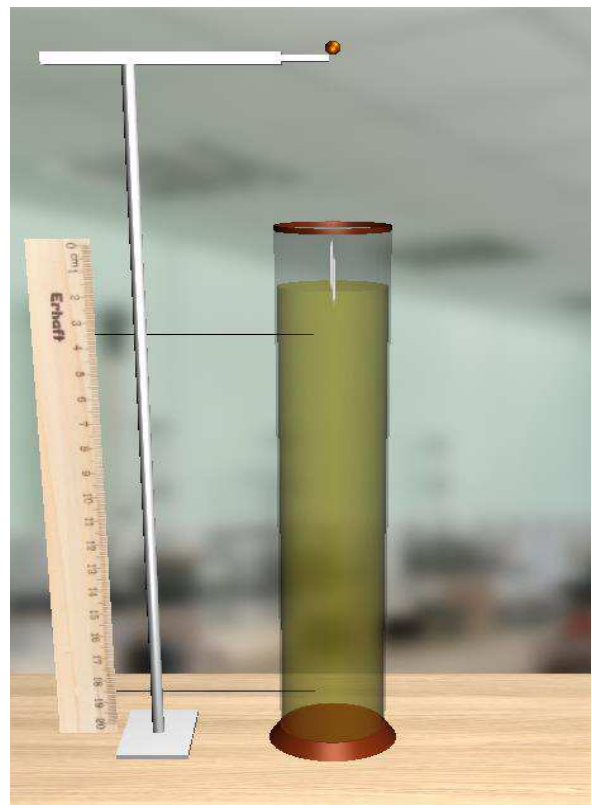
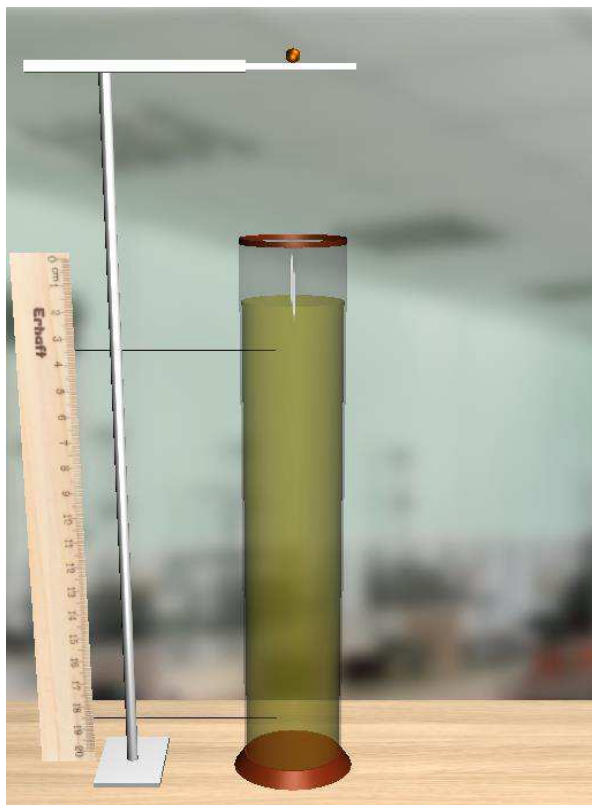


Рисунок 77 – Скриншот виртуальной лабораторной установки с разным положением падающего шарика

Все параметры рассчитаны ранее и максимально приближают физический процесс к реальному.

Все функции, производящие динамические изменения параметров 3D объектов, вызываются функцией `update`, код которой показан на рисунке 78.

```
388 function update(){
389     if (flag_add_water!=0) {
390         addwater();
391     }
392     izm();
393     render();
394     requestAnimationFrame(update);
395 }
```

Рисунок 78 – Код функции `update`

Она вызывает себя функцией `requestAnimationFrame()`, созданной для обработки 3D объектов в анимации, а также функции `izm`, `addWater` (вызывается, если процесс наполнения жидкости не завершился) и `render`.

Функция `render`(код на рисунке 79) создана для рендеринга сцены и камеры, а поскольку в приложении используется анимация, то и рендеринг должен быть постоянным, поэтому он вызывается в функции `update`.

```
398 function render()
399 {
400     renderer.render(scene, camera);
401 }
```

Рисунок 79 – Код функции `render`

На рисунке 80 представлены три рекомендуемые для выполнения измерений состояния виртуальной лабораторной установки.

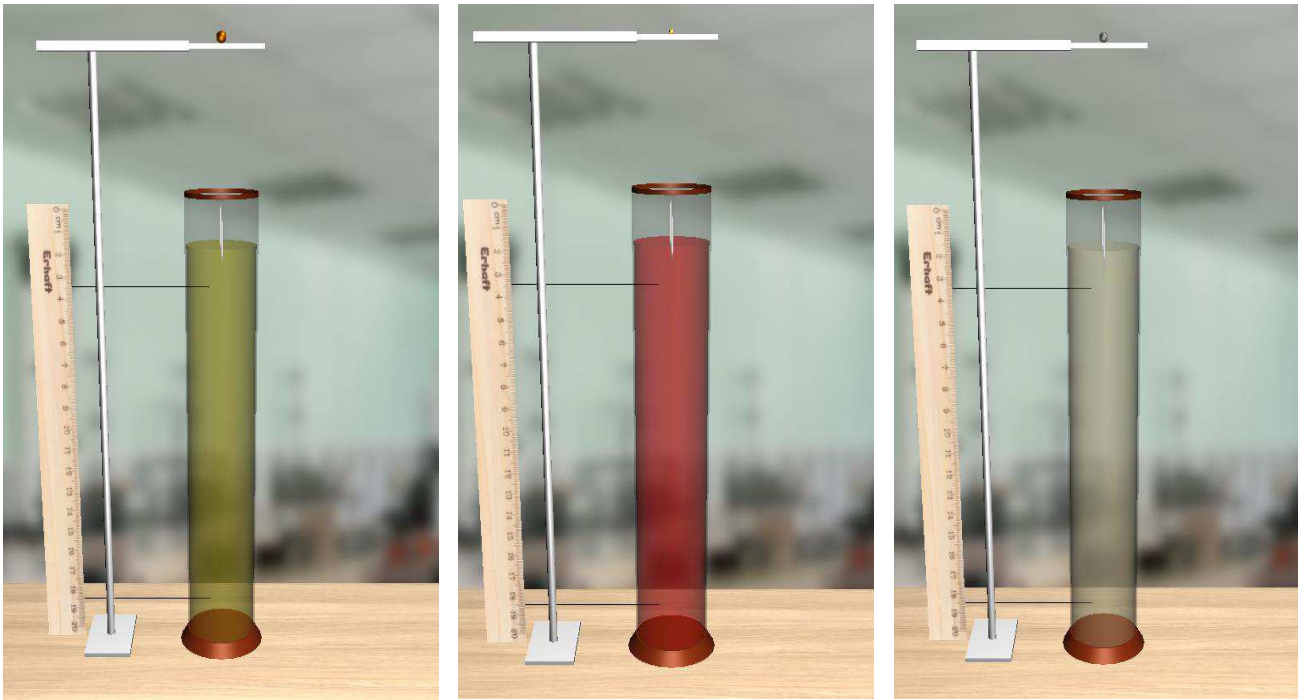


Рисунок 80 – Скриншот виртуальной лабораторной установки с рекомендуемыми сочетаниями шарика и жидкости

При разработке двух других лабораторных работ использованы несколько иные возможности программирования для демонстрации физических явлений.

#### **2.2.4 Создание имитационной модели физического процесса лабораторной работы «Изучение явления дифракции света»**

Начальные этапы разработки этой лабораторной работы реализуются так же, как и в случае разработки лабораторной работы «Изучение явления внутреннего трения жидкости». В случае выбора лабораторной работы 2, файл разметки process.php генерирует для этой лабораторной работы левое и правое меню, с похожим дизайном первой работы, но с отличающимися от нее кнопками (рисунки 81, 82).

```

111 <?php }
112     if ($id_lr==2) { ?>
113     <div class="leftcontainer">
114         <script type="text/javascript" src="js/mainapp<?php echo($id_lr) ?>.js"></script>
115     <div class="leftupcontainer">
116         <h1>Выберите решетку и расстояние до экрана</h1>
117         <form name="grating1form" method="post">
118             <input type="button" value="Выбрано" name="grating1" class="input2" onclick="SelectGrating1()">
119             <p>d = 1/100 мм</p>
120         </form>
121         <form id="5" name="grating2form" method="post">
122             <input id="1" type="button" value="Выбрать" name="grating2" class="input2" onclick="SelectGrating2()">
123             <p>d = 1/250 мм</p>
124         </form>
125         <form name="dist1form" method="post">
126             <div class="class1">
127                 <p>L = 25 см</p>
128             </div>
129             <input type="button" value="Установлено" name="dist1" class="input2" onclick="SelectDist1()">
130         </form>
131         <form name="dist2form" method="post">
132             <div class="class1">
133                 <p>L = 18 см</p>
134             </div>
135             <input type="button" value="Установить" name="dist2" class="input2" onclick="SelectDist2()">
136         </form>
137         <form name="dist3form" method="post">
138             <div class="class1">
139                 <p>L = 11 см</p>
140             </div>
141             <input type="button" value="Установить" name="dist3" class="input2" onclick="SelectDist3()">
142         </form>
143         <form name="laserColor1form" method="post">
144             <input type="button" value="Выбран" name="color1" class="input2" onclick="SelectLaserColor1()">
145             <p>Красный лазер</p>
146         </form>
147         <form name="laserColor2form" method="post">
148             <input type="button" value="Выбрать" name="color2" class="input2" onclick="SelectLaserColor2()">
149             <p>Зеленый лазер</p>
150         </form>
151     </div>
152
153     <div class="instruction">
154         <div class="recom">
155             <p>Примечание:</p>
156             <p>решетка с периодом 1/200 мм используется при выполнении работы с лазером</p>
157         </div>
158         <h1>Инструкция к выполнению работы</h1>
159         <p>1. Включите источник света (лампу или лазер).</p>
160         <p>2. Переключите камеру.</p>
161         <p>3. Выполните измерения.</p>
162         <p>4. Зафиксируйте результаты.</p>
163     </div>
164 </div>
165 </div>
166 <?php }

```

Рисунок 81 – Код левого меню 2 лабораторной работы

```

218     if ($id_lr==2) { ?>
219     <div class="buttoncontainer">
220         <div class="button">
221             <form name="startCameraForm" method="post">
222                 <input name="processCameraStarter" type="button" value="Переключить камеру" class="input2" onclick="clicktCamera()">
223             </form>
224             <form name="startProcessForm" method="post">
225                 <input name="processStarter" type="button" value="Включить лампу" class="input2" onclick="clickt()">
226             </form>
227             <form name="startLaserForm" method="post">
228                 <input name="laserStarter" type="button" value="Включить лазер" class="input2" onclick="clicktLaser()">
229             </form>
230         </div>
231     </div>
232 </div>
233 <?php }

```

Рисунок 82 – Код правого меню 2 лабораторной работы

Эта лабораторная работа предполагает наличие кнопок выбора периода дифракционной решетки (2 разных решетки), а также установки экрана на 3 расстояния. Кроме того, в задании к работе есть изучение монохромного источника света двух цветов, поэтому на страницу добавлены кнопки выбора

цвета лазера. Правое меню состоит из кнопок включения источников света (лазера для монохромного света или газоразрядной лампы – для полихромного) и кнопка переключения положения камеры. Скриншоты этих меню на веб-странице показаны на рисунке 83.

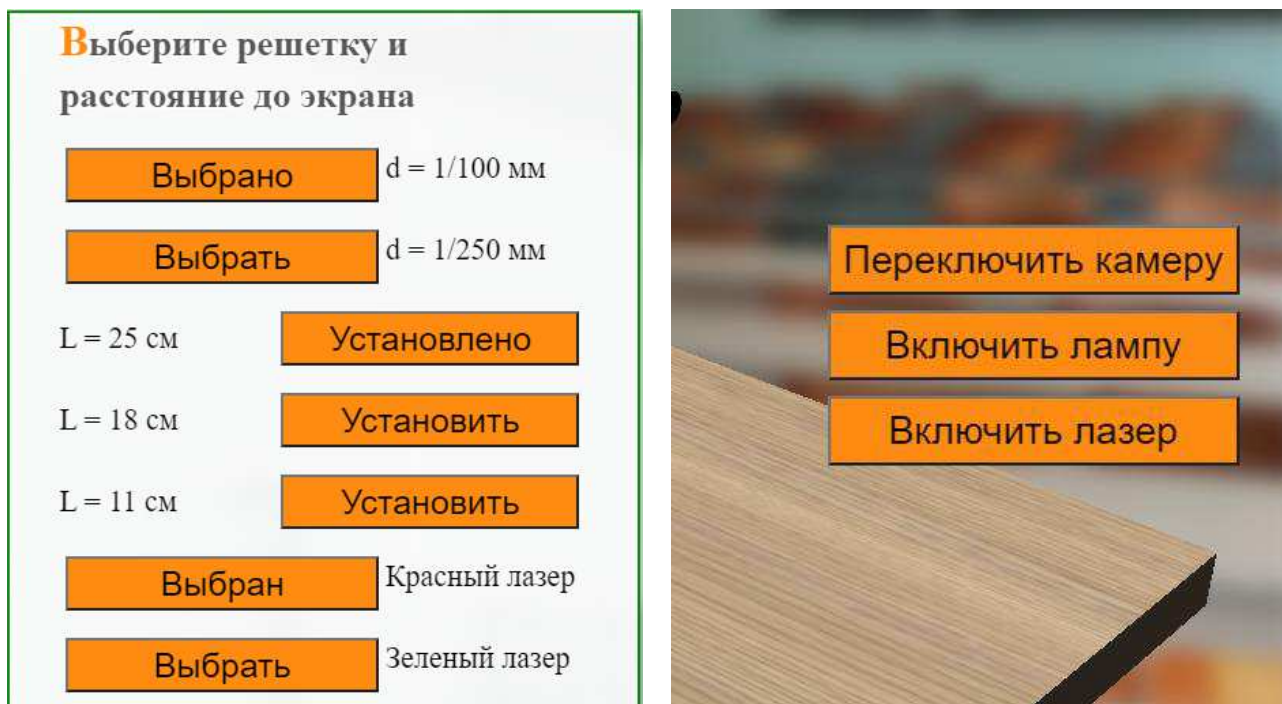


Рисунок 83 – Скриншот левого и правого меню 2 лабораторной работы

Физический процесс лабораторной работы «Изучение явления дифракции света» представляет собой отображение полихромного или монохромного светового спектра на экране установки (передвижной линейки) при включении источника света (лампы или лазера).

Виртуальная установка для имитации этого процесса состоит из некоторой поверхности (стола), на которой расположены: оптическая скамья, состоящая из закрепленной на стойке линейки с миллиметровыми делениями, передвижным экраном со щелью и держателем дифракционной решетки, газоразрядная лампа, источник лазерного излучения. Скриншоты установки показаны на рисунках 84-85.





Рисунок 84 – Скриншот установки лабораторной работы 2



Рисунок 85 – Скриншот установки лабораторной работы 2 с источником светом

Для реализации этой установки в файле `mainapp2.js` с помощью функций `three.js`, и специально разработанных для этой лабораторной работы `javascript`-функций была создана виртуальная сцена, на которую добавлены соответствующие объекты. Перед созданием сцены были объявлены необходимые для реализации глобальные переменные, код объявления которых

показан на рисунке 86.

```
mainapp2.js
1
2 var scene;
3 var camera;
4 var controls;
5 var light;
6 var renderer, container;
7 var Plane, Sphere, Cylinder, Line;
8 var PanelColorSpectorRight, PanelColorSpectorLft;
9 var geometryColorSpector, materialColor, rghtcolorObject, lftcolorObject;
10 var rghtcolorObjects = [];
11 var lftcolorObjects = [];
12 var distance;
13 var flag=0;
14 var flag_laser=0;
15 var flag_laser_color=0;
16 var flag_camera=0;
17 var flag_add_spector=0;
18 var flag_spector=0;
19 var flag_dist=1;
20 var flag_position=1;
21 var flag_grating=1;
22 var flag_clickt;
23 var flag_add_colorPanel=0;
24 var DinamLinePositionZ;
25 var materialLamp;
26
27 var Sht = new THREE.Group();
28 var DinamLine = new THREE.Group();
29 var Grating = new THREE.Group();
30 var lamp = new THREE.Group();
31 var laser = new THREE.Group();
32
33 init();
34
35 function init(){
36
37     CreateScene();
38     CreateObjects();
39     update();
40 }
41
```

Рисунок 86 – Код объявления переменных лабораторной работы 2

Также, в этой работе 3D объекты объединены в группы, для удобства разработки, поэтому с помощью функций THREE.Group были созданы группы для размещения в них объектов штатива, динамической линейки, дифракционной решетки, газоразрядной лампы и лазера. Как и в случае первой лабораторной работы, здесь запускается функция init, состоящая из функций создания сцены, создания объектов и анимации.

Процесс создания виртуальной сцены такой же, как и в лабораторной работе «Изучение явления внутреннего трения жидкости» и продемонстрирован в пункте 2.2.3. Объекты виртуальной сцены отличные от объектов первой лабораторной работы созданы теми же средствами библиотеки three.js и показаны на рисунке далее.

Для имитации монохромного светового спектра используются точки, появляющиеся при включении лазера на динамическом экране. Создание объектов для этих точек показано на рисунке 87.

```
117 for (var i = 0; i < 2; i++) {
118     var geometryLaserSpector = new THREE.CircleGeometry(0.02, 16);
119     var materialColor = new THREE.MeshPhongMaterial({color: 0xffffff, specular: 0x6b6b6b});
120     rghtcolorObject = new THREE.Mesh(geometryLaserSpector, materialColor);
121     lftcolorObject = rghtcolorObject.clone();
122     rghtcolorObjects.push(rghtcolorObject);
123     lftcolorObjects.push(lftcolorObject);
124 }
```

Рисунок 87 – Код создания световых точек лазера

Для имитации полихромного цветового спектра используются плоские панели с наложенными на них изображениями светового спектра. Процесс создания панелей показан на рисунке 88. Варианты цветового спектра созданы по расчетам для дифракционной решетки с периодом 1/100 мм и расстояний до экрана 250, 180 и 110 мм.

```
127 var geometryColorSpector250 = new THREE.PlaneGeometry(2.75, 0.25);
128 var TexturelftSpector250 = new THREE.ImageUtils.loadTexture( 'image/lftSpector250mm.png' );
129 var TextureRghtSpector250 = new THREE.ImageUtils.loadTexture( 'image/rghtSpector250mm.png' );
130 var material1Color250 = new THREE.MeshPhongMaterial({map: TexturelftSpector250, specular: 0x6b6b6b, transparent: true});
131 var material2Color250 = new THREE.MeshPhongMaterial({map: TextureRghtSpector250, specular: 0x6b6b6b, transparent: true});
132 LftColorSpector250 = new THREE.Mesh(geometryColorSpector250, material1Color250);
133 RghtColorSpector250 = new THREE.Mesh(geometryColorSpector250, material2Color250);
134 LftColorSpector250.position.set(-1.35, 0.476, 0.26);
135 RghtColorSpector250.position.set(1.35, 0.476, 0.26);
136
137 var geometryColorSpector180 = new THREE.PlaneGeometry(2.1, 0.18);
138 var TexturelftSpector180 = new THREE.ImageUtils.loadTexture( 'image/lftSpector180mm.png' );
139 var TextureRghtSpector180 = new THREE.ImageUtils.loadTexture( 'image/rghtSpector180mm.png' );
140 var material1Color180 = new THREE.MeshPhongMaterial({map: TexturelftSpector180, specular: 0x6b6b6b, transparent: true});
141 var material2Color180 = new THREE.MeshPhongMaterial({map: TextureRghtSpector180, specular: 0x6b6b6b, transparent: true});
142 LftColorSpector180 = new THREE.Mesh(geometryColorSpector180, material1Color180);
143 RghtColorSpector180 = new THREE.Mesh(geometryColorSpector180, material2Color180);
144 LftColorSpector180.position.set(-1.05, 0.476, 0.26);
145 RghtColorSpector180.position.set(1.05, 0.476, 0.26);
146
147 var geometryColorSpector110 = new THREE.PlaneGeometry(1.42, 0.11);
148 var TexturelftSpector110 = new THREE.ImageUtils.loadTexture( 'image/lftSpector110mm.png' );
149 var TextureRghtSpector110 = new THREE.ImageUtils.loadTexture( 'image/rghtSpector110mm.png' );
150 var material1Color110 = new THREE.MeshPhongMaterial({map: TexturelftSpector110, specular: 0x6b6b6b, transparent: true});
151 var material2Color110 = new THREE.MeshPhongMaterial({map: TextureRghtSpector110, specular: 0x6b6b6b, transparent: true});
152 LftColorSpector110 = new THREE.Mesh(geometryColorSpector110, material1Color110);
153 RghtColorSpector110 = new THREE.Mesh(geometryColorSpector110, material2Color110);
154 LftColorSpector110.position.set(-0.7, 0.476, 0.26);
155 RghtColorSpector110.position.set(0.7, 0.476, 0.26);
156
157 var geometryBlic = new THREE.PlaneGeometry(0.8, 0.8);
158 var TextureBlic = new THREE.ImageUtils.loadTexture( 'image/blik2.png' );
159 var materiaBlic = new THREE.MeshPhongMaterial({map: TextureBlic, specular: 0x6b6b6b, transparent: true, emissive: 0xffffff, emissiveIntensity: 2});
160 blic = new THREE.Mesh(geometryBlic, materiaBlic);
161 blic.position.set(0.01, 0.430, 0.26);
```

Рисунок 88 – Код создания панелей для имитации полихромного спектра

На этом же рисунке показан процесс создания панели с текстурой светового блика (рисунок 88, строки 157-161).

Геометрия основных объектов, использованных на сцене этой лабораторной работе, а также, загрузка изображений текстур для стола, линейки штатива и линейки экрана показана на рисунках 89-90.

```
165 var geometryTable = new THREE.BoxGeometry(9, 0.4, 20);
166
167 var geometryShtCylinder = new THREE.CylinderGeometry(0.1, 0.1, 2.5, 12, 1, true);
168 var geometryShtBox = new THREE.BoxGeometry(2, 0.3, 2);
169 var geometryShtLineBox = new THREE.BoxGeometry(0.4, 0.4, 12);
170 var geometryShtGratingDwnBox = new THREE.BoxGeometry(1.2, 0.2, 0.1);
171 var geometryShtGratingLftRghtBox = new THREE.BoxGeometry(0.2, 0.9, 0.1);
172
173 var geometryDinamBox = new THREE.BoxGeometry(0.6, 0.1, 0.5);
174 var geometryDinamLftRghtBox = new THREE.BoxGeometry(0.1, 0.5, 0.5);
175 var geometryDinamUpBox = new THREE.BoxGeometry(0.05, 0.08, 0.1);
176 var geometryDinamLineLftRghtBox = new THREE.BoxGeometry(2.45, 0.6, 0.1);
177
178 var geometryGratingGlassBox = new THREE.BoxGeometry(0.7, 0.6, 0.02);
179 var geometryGratingLftRghtBox = new THREE.BoxGeometry(0.06, 0.6, 0.06);
180 var geometryGratingUpDownBox = new THREE.BoxGeometry(0.8, 0.06, 0.06);
181
182 var geometryLampCylinder = new THREE.CylinderGeometry(0.1, 0.1, 0.5, 12, 1, true);
183 var geometryLampGlassCylinder = new THREE.CylinderGeometry(0.05, 0.05, 4, 12, 1);
184 var geometryLaserPoint = new THREE.CircleGeometry(0.02, 16);
185
```

Рисунок 89 – Код создания геометрии основных объектов

```
190
191 var TextureTable = new THREE.ImageUtils.loadTexture( 'image/textureTable.jpeg' );
192 var TextureLine = new THREE.ImageUtils.loadTexture( 'image/textureLine.jpg' );
193 var TextureShtLineBox = new THREE.ImageUtils.loadTexture( 'image/textureShtBox.jpg' );
194 var TextureBlackLineLft = new THREE.ImageUtils.loadTexture( 'image/TextureBlackLineLft.jpg' );
195 var TextureBlackLineRght = new THREE.ImageUtils.loadTexture( 'image/TextureBlackLineRght.jpg' );
196
```

Рисунок 90 – Загрузка внешних текстур

Материалы объектов создаются так же, как и в лабораторной работе 1, за исключением материалов для линеек. На рисунке 91 показан код создания основных материалов. Материал стекла использовался для имитации виртуальной дифракционной решетки. Так же, для этого материала было включено двухстороннее отображение граней.

```

225 var materialGlass = new THREE.MeshPhysicalMaterial({
226   color: 0x2e2e2e,
227   metalness: 0,
228   roughness: 0,
229   envMapIntensity: 1,
230   alphaTest: 0.5,
231   depthWrite: false,
232   transmission: 0.85, // use material.transmission for glass materials
233   opacity: 1, // set material.opacity to 1 when material.transmission is non-zero
234   transparent: true } );
235 var materialMetal = new THREE.MeshStandardMaterial({
236   color: 0xffffffff,
237   roughness: 0.5,
238   metalness: 1});
239 var materialPlastic = new THREE.MeshPhongMaterial({color: 0x000000, specular: 0x6b6b6b});
240 var materialGratingPlastic = new THREE.MeshPhongMaterial({color: 0x1f0a0a, specular: 0x6b6b6b});
241 var materialTable = new THREE.MeshPhongMaterial( { map: TextureTable } );
242 materialLamp = new THREE.MeshPhongMaterial({color: 0xffffffff, specular: 0x6b6b6b, emissive: 0xffffffff, emissiveIntensity: 1});
243 var materialLaserPoint = new THREE.MeshPhongMaterial({color: 0xff0000, specular: 0x6b6b6b, transparent: true, opacity: 0.6});
244
245 materialGlass.side = THREE.DoubleSide;

```

Рисунок 91 – Код создания основных материалов

Для создания линейки штатива и линейки экрана был использован объект библиотеки three.js типа box, который имеет шесть граней. Текстура линейки со шкалой деления должна быть наложена только на одну грань из шести, остальные грани не имеют, шкалы деления – только текстуру дерева (или цвет материала линейки). Для реализации этого был создан массив из 6 разных материалов, которые объединены в один с помощью функции THREE.MeshFaceMaterial. Код этого процесса показан на рисунке 92.

```

197 var ShtLineBoxMaterials = [
198   new THREE.MeshPhongMaterial( { map: TextureShtLineBox } ),
199   new THREE.MeshPhongMaterial( { map: TextureShtLineBox } ),
200   new THREE.MeshPhongMaterial( { map: TextureLine } ),
201   new THREE.MeshPhongMaterial( { map: TextureShtLineBox } ),
202   new THREE.MeshPhongMaterial( { map: TextureShtLineBox } ),
203   new THREE.MeshPhongMaterial( { map: TextureShtLineBox } )
204 ]
205 var materialShtLineBox = new THREE.MeshFaceMaterial(ShtLineBoxMaterials);

```

Рисунок 92 – Код создания и объединения материалов для линейки

Аналогично были созданы материалы для левой и правой части динамической линейки экрана (рисунок 93), вместо текстуры дерева для граней, не имеющих шкалы деления, был использован черный материал типа Phong.

```

206 var DinamLineLftBoxMaterials = [
207   new THREE.MeshPhongMaterial( { color: 0x000000 } ),
208   new THREE.MeshPhongMaterial( { color: 0x000000 } ),
209   new THREE.MeshPhongMaterial( { color: 0x000000 } ),
210   new THREE.MeshPhongMaterial( { color: 0x000000 } ),
211   new THREE.MeshPhongMaterial( { map: TextureBlackLineLft } ),
212   new THREE.MeshPhongMaterial( {color: 0x000000 } )
213 ]
214 var materialDinamLineLftBox = new THREE.MeshFaceMaterial(DinamLineLftBoxMaterials);

```

Рисунок 93 – Код создания материала для левой части линейки экрана

Код создания всех объектов с ранее созданными геометрией и материалами показан на рисунке 94.

```

249 ShtCylinder = new THREE.Mesh(geometryShtCylinder, materialMetal);
250 ShtBox = new THREE.Mesh(geometryShtBox, materialMetal);
251 ShtLineBox = new THREE.Mesh(geometryShtLineBox, materialShtLineBox);
252 ShtGratingDownBox = new THREE.Mesh(geometryShtGratingDownBox, materialMetal);
253 ShtGratingLftBox = new THREE.Mesh(geometryShtGratingLftRghtBox, materialMetal);
254 ShtGratingRghtBox = new THREE.Mesh(geometryShtGratingLftRghtBox, materialMetal);
255
256 DinamBox = new THREE.Mesh(geometryDinamBox, materialPlastic);
257 DinamLftBox = new THREE.Mesh(geometryDinamLftRghtBox, materialPlastic);
258 DinamRghtBox = new THREE.Mesh(geometryDinamLftRghtBox, materialPlastic);
259 DinamUpBox = new THREE.Mesh(geometryDinamUpBox, materialPlastic);
260 DinamDwnBox = DinamUpBox.clone();
261 DinamLineLftBox = new THREE.Mesh(geometryDinamLineLftRghtBox, materialDinamLineLftBox);
262 DinamLineRghtBox = new THREE.Mesh(geometryDinamLineLftRghtBox, materialDinamLineRghtBox);
263
264 GratingGlassBox = new THREE.Mesh(geometryGratingGlassBox, materialGlass);
265 GratingLftBox = new THREE.Mesh(geometryGratingLftRghtBox, materialGratingPlastic);
266 GratingRghtBox = new THREE.Mesh(geometryGratingLftRghtBox, materialGratingPlastic);
267 GratingUpBox = new THREE.Mesh(geometryGratingUpDwnBox, materialGratingPlastic);
268 GratingDwnBox = new THREE.Mesh(geometryGratingUpDwnBox, materialGratingPlastic);
269
270 LampCylinder = new THREE.Mesh(geometryLampCylinder, materialPlastic);
271 LampGlassCylinder = new THREE.Mesh(geometryLampGlassCylinder, materialLamp);
272 LaserPoint = new THREE.Mesh(geometryLaserPoint, materialLaserPoint);
273 LaserPoint.visible = false;
274

```

Рисунок 94 – Код создания 3d объектов лабораторной работы 2

Для большего реализма после включения лазера на дифракционной решетке появляется полупрозрачная точка. В строке 273 рисунка 94 устанавливается свойство, которое скрывает эту точку, пока лазер выключен.

После создания объектов, с помощью функции `position.set()` им указываются координаты размещения в группе, а с помощью функции `add` они добавляются в подготовленную для них группу, после чего устанавливается

положение группы на сцене. На рисунке 95 показан код этого процесса для элементов штатива.

```
277 ShtCylinder.position.set(0, 0, 0);
278 ShtBox.position.set(0, -1.4, 0);
279 ShtLineBox.position.set(0, 1.4, 0);
280 ShtGratingDwnBox.position.set(0, 1.6, 6.05);
281 ShtGratingLftBox.position.set(-0.5, 2.15, 6.05);
282 ShtGratingRghtBox.position.set(0.5, 2.15, 6.05);
283 Sht.add(ShtCylinder);
284 Sht.add(ShtBox);
285 Sht.add(ShtLineBox);
286 Sht.add(ShtGratingDwnBox);
287 Sht.add(ShtGratingLftBox);
288 Sht.add(ShtGratingRghtBox);
289 Sht.position.set(0, 2.6, 0);
```

Рисунок 95 – Код размещения элементов штатива на сцене

Аналогичным образом, размещаются и добавляются в группы элементы динамической линейки (экрана) и дифракционной решетки. После этого все элементы (группы) добавляются на сцену функцией add (рисунок 96).

```
333 scene.add(Table);
334 scene.add(Sht);
335 scene.add(DinamLine);
336 scene.add(Grating);
337 scene.add(lamp);
338 scene.add(laser);
```

Рисунок 96 – Код добавления на сцену групп объектов

Все координаты и размеры 3D объектов были рассчитаны заранее.

Имитация физического процесса отображение светового спектра на динамической линейке (экране) установки построена следующим образом: после включения источника света (лампы или лазера) предварительно созданные плоские панели полихромного или монохромного цветового спектра, которые до этого момента были с отключенным свойством visible, появляются на сцене. Для реализации этого были разработаны и созданы связанные между собой javascript-функции.

Кроме того, во второй (как и в третьей работе) был создан механизм в виде нескольких функции, позволяющий динамически менять положение

камеры, чтобы пользователь не имел трудностей с измерением расстояний. Код такого механизма для второй лабораторной работы представлен на рисунке 97.

На рисунке 98 показано изменение положения камеры для лабораторной работы «Изучение явления дифракции света».

При движении камеры пользователь видит плавный поворот объектов сцены, так же как это могло бы происходить в реальности при движении головы пользователя, пытающегося изменить свое положение относительно установки, то есть переместить взор с обзора стола усаживаясь за стол и посмотрев на экран сквозь дифракционную решетку.



```

568 ▼ function clicktCamera(){
569     if (flag_camera==0) {
570         flag_camera=1;
571     } else {
572         flag_camera=0;
573     }
574 }
575
576 ▼ function movCameraDown(){
577 ▼     if (camera.position.x<0) {
578         camera.position.x += 0.15;
579         camera.position.y += -0.05;
580         camera.position.z += -0.090;
581         camera.rotation.x += 0.005;
582         camera.rotation.y += 0.01;
583         camera.rotation.z += 0.0025;
584 ▼     } else {
585         camera.position.x = 0;
586         camera.position.y = 4.8;
587         camera.position.z = 7.45;
588         camera.rotation.x = 0;
589         camera.rotation.y = 0;
590         camera.rotation.z = 0;
591         flag_spector=1;
592     }
593 }
594 ▼ function movCameraUp(){
595 ▼     if (camera.position.x>-9) {
596         camera.position.x += -0.15;
597         camera.position.y += 0.05;
598         camera.position.z += 0.090;
599         camera.rotation.x += -0.005;
600         camera.rotation.y += -0.01;
601         camera.rotation.z += -0.003;
602         flag_spector=0;
603 ▼     } else {
604         camera.position.x = -9.7;
605         camera.position.y = 8;
606         camera.position.z = 13.5;
607         camera.rotation.x = -0.3;
608         camera.rotation.y = -0.6;
609         camera.rotation.z = -0.17;
610     }
611 }

```

Рисунок 97 – Код функций изменения положения камеры



Рисунок 98 – Скриншоты страниц с различным положением камеры в лабораторной работе 2

В процессе выполнения измерений в данной лабораторной работе пользователю предоставляется возможность выбрать одно из трех расстояний положения динамической линейки (экрана) от дифракционной решетки – 25, 18 и 11 сантиметров. Визуализация размещения этой линейки анимирована и представлена в виде плавного перехода экрана с одного положения в другое. Реализовано это с помощью функций выбора одного из положений линейки `SelectDist[номер положения линейки]` (код на рисунке 99), которые привязаны к кнопкам левого меню и функции изменения положения группы объектов динамической линейки `movDinamLine` (код на рисунке 100), которая регулярно вызывается функцией анимации `update`.

```

470 ▼ function SelectDist1(){
471     flag_dist=1;
472     flag_position=1;
473     document.dist1form.dist1.value="Установлено";
474     document.dist2form.dist2.value="Установить";
475     document.dist3form.dist3.value="Установить";
476 }

```

Рисунок 99 – Код одной из функций выбора положения экрана

```

494 ▼ function movDinamLine(){
495 ▼     if (flag_dist==1) {
496 ▼         if (DinamLine.position.z>-4.12) {
497 ▼             if (flag_specter==1&&flag==1) {
498 ▼                 flag_clickt=1;
499 ▼                 clickt();
500 ▼             }
501 ▼             DinamLine.position.z+=-0.07;
502 ▼         } else {
503 ▼             DinamLine.position.z = -4.12;
504 ▼             distance-1;
505 ▼             if (flag_clickt==1){
506 ▼                 clickt();
507 ▼             }
508 ▼             flag_clickt=0;
509 ▼         }
510 ▼     }
511 ▼     if (flag_dist==2&&flag_position==1) {
512 ▼         if (DinamLine.position.z<-1.4) {
513 ▼             if (flag_specter==1&&flag==1) {
514 ▼                 flag_clickt=1;
515 ▼                 clickt();
516 ▼             }
517 ▼             DinamLine.position.z+=0.07;
518 ▼         } else {
519 ▼             DinamLine.position.z = -1.4;
520 ▼             flag_position=0;
521 ▼             distance+2;
522 ▼             if (flag_clickt==1){
523 ▼                 clickt();
524 ▼             }
525 ▼             flag_clickt=0;
526 ▼         }
527 ▼     }
528 ▼     if (flag_dist==2&&flag_position==3) {
529 ▼         if (DinamLine.position.z>-1.4) {
530 ▼             if (flag_specter==1&&flag==1) {
531 ▼                 flag_clickt=1;
532 ▼                 clickt();
533 ▼             }
534 ▼             DinamLine.position.z+=-0.07;
535 ▼         } else {
536 ▼             DinamLine.position.z = -1.4;
537 ▼             flag_position=0;
538 ▼             distance-2;
539 ▼             if (flag_clickt==1){
540 ▼                 clickt();
541 ▼             }
542 ▼             flag_clickt=0;
543 ▼         }
544 ▼     }
545 ▼     if (flag_dist==3) {
546 ▼         if (DinamLine.position.z<1.25) {
547 ▼             if (flag_specter==1&&flag==1) {
548 ▼                 flag_clickt=1;
549 ▼                 clickt();
550 ▼             }
551 ▼             DinamLine.position.z+=0.07;
552 ▼         } else {
553 ▼             DinamLine.position.z = 1.25;
554 ▼             distance+3;
555 ▼             if (flag_clickt==1){
556 ▼                 clickt();
557 ▼             }
558 ▼             flag_clickt=0;
559 ▼         }
560 ▼     }
561     DinamLinePositionZ = DinamLine.position.z;
562 }
563

```

Рисунок 100 – Код функции изменения положения экрана

Данная функция при определенных условиях изменяет положение экрана, в зависимости от того, какое положение выбрал пользователь, а также от того, какое положение в этот момент занимает этот экран.

При изучении полихромного спектра данной виртуальной лабораторной работой не предусмотрено динамическое изменение этого спектра при движении экрана, поэтому, функция `movDinamLine`, в момент перемещения отключает источник света (газоразрядную лампу) и включает его при прекращении перемещения. За включение и выключение лампы отвечает функция `clickt`, которая, помимо вызова функцией `movDinamLine`, так же вызывается при нажатии на кнопку «Включить лампу». Похожую структуру имеет функция включения лазера `clicktLaser`, которая вызывается только кнопкой «Включить лазер».

```
621 ▼ function clickt(){
622 ▼     if (flag==0&&flag_grating!=2) {
623         flag=1;
624         flag_laser=0;
625         LaserPoint.visible = false;
626         document.grating2form.grating2.disabled=true;
627         cylinder5.material.emissiveIntensity = 0;
628         document.startProcessForm.processStarter.value="Выключить лампу";
629         document.startLaserForm.laserStarter.value="Включить лазер";
630 ▼     } else {
631         flag=0;
632         document.grating2form.grating2.disabled=false;
633         document.startProcessForm.processStarter.value="Включить лампу";
634     }
635 }
636 }
637 ▼ function clicktLaser(){
638 ▼     if (flag_laser==0) {
639         flag_laser=1;
640         flag=0;
641         LaserPoint.visible = true;
642         document.grating2form.grating2.disabled=false;
643         cylinder5.material.emissiveIntensity = 2;
644         document.startLaserForm.laserStarter.value="Выключить лазер";
645         document.startProcessForm.processStarter.value="Включить лампу";
646 ▼     } else {
647         flag_laser=0;
648         LaserPoint.visible = false;
649         cylinder5.material.emissiveIntensity = 0;
650         document.startLaserForm.laserStarter.value="Включить лазер";
651     }
652 }
653 }
```

Рисунок 101 – Код функций `clickt` и `clicktLaser`

Эти функции меняют значение переменных `flag` и `flag_laser`, отвечающих за запуск анимированных функций включения источников света, а также меняют названия соответствующих кнопок.

Кроме того, они меняют значения доступности некоторых кнопок `disabled`, так как в лабораторной работе есть две дифракционные решетки, одна из которых не используется при включении газоразрядной лампы. Поэтому кнопка выбора второй дифракционной решетки должна быть неактивна после включения лампы, и наоборот.

Так же, кнопка включения лампы отключает лазер, а кнопка включения лазера, отключает лампу.

За включение (выключение) лампы отвечает анимированная функция `Lamp_OnOf` (рисунок 102), которая меняет свойства 3D объекта в зависимости от переменной, значение которой устанавливает функция `clickt` или `clicktLaser`.

```
655 function Lamp_OnOf(){
656     if (flag==1) {
657         LampLight.intensity = 1;
658         materialLamp.emissiveIntensity = 1;
659     }
660     else {
661         LampLight.intensity = 0;
662         materialLamp.emissiveIntensity = 0;
663     }
664 }
```

Рисунок 102 – Код функции `Lamp_OnOf`

Если переменная `flag` принимает значение 1, то функция делает значение интенсивности созданного источника свет с 0 на 1, а так же меняет свойство материала цилиндра, имитирующего лампу, для большего реализма. По умолчанию, источники света отключены. На рисунке 103 показаны скриншоты демонстрации установки с включенным и выключенным источником света (газоразрядной лампы или лазера).

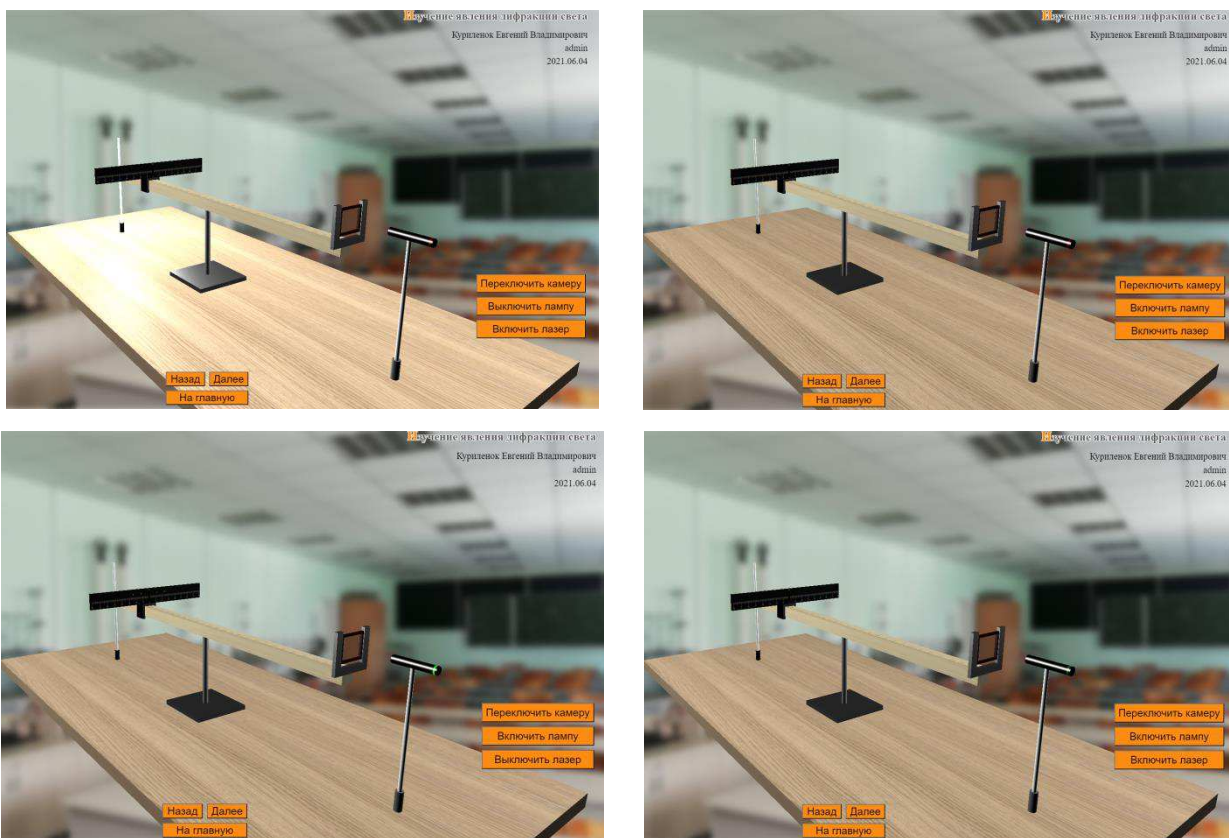


Рисунок 103 – Скриншот включения источников света лабораторной работы 2

Так как в этой лабораторной работе предусмотрены 2 дифракционные решетки, для их выбора были написаны функции SelectGrating1 и SelectGrating2 (рисунок 104).

```

453 function SelectGrating1(){
454     flag_grating=1;
455     document.grating1form.grating1.value="Выбрано";
456     document.grating2form.grating2.value="Выбрать";
457     document.startProcessForm.processtarter.disabled=false;
458 }
459 function SelectGrating2(){
460     flag_grating=2;
461     document.grating1form.grating1.value="Выбрать";
462     document.grating2form.grating2.value="Выбрано";
463     document.startProcessForm.processtarter.disabled=true;
464 }

```

Рисунок 104 – Код функций выбора дифракционных решеток

Эти функции меняют соответствующие переменные, названия кнопок, а также, функция кнопки выбора второй решетки, блокирует кнопку выбора первой.

Так же, в этой лабораторной работе монохромный источник света представлен в виде лазера двух цветов – красного и зеленого, поэтому для выбора цвета лазера были написаны функции `SelectLaserColor1` и `SelectLaserColor2`, код одной из которых показан на рисунке 105.

```
435 function SelectLaserColor1(){
436     flag_laser_color=0;
437     LaserPoint.material.color.setHex(0xff0000);
438     cylinder5.material.color.setHex(0x900000);
439     cylinder5.material.emissive.setHex(0xff0000);
440     document.laserColor1form.color1.value="Выбран";
441     document.laserColor2form.color2.value="Выбрать";
442 }
```

Рисунок 105 – Код функции `SelectLaserColor1`

Данная функция меняет значение переменной `flag_laser_color`, чтобы использующие ее функции «понимали», какого цвета лазер в текущий момент, а также меняет значение названий кнопок и цветовые свойства 3D объекта лазера.

Имитация появления на дифракционной решетке полихромного цветового спектра описана анимированной функцией `AddColorSpector`, код которой показан на рисунке 106.

```
341 function AddColorSpector(){
342     if (flag==1&&flag_spector==1&&flag_laser==0) {
343         DinamLine.add(blic);
344         CreateColorSpector();
345         flag_add_spector=1;
346     } else {
347         DinamLine.remove(blic);
348         DeleteColorSpector();
349         flag_add_spector=0;
350     }
351 }
```

Рисунок 106 – Код функции `AddColorSpector`

Эта функция проверяет значение переменных включения света (`flag`), движения динамической линейки (`flag_spector`) и выключения лазера (`flag_laser`). В случае нужных значений этих переменных (`flag=1`, `flag_spector=1` и `flag_laser=0`), функция добавляет в группу динамической линейки плоскую

панель с текстурой блика, вызывает функцию добавления в группу динамической линейки плоских панелей цветового спектра CreateColorSpector (код на рисунке 107), и меняет значение переменной flag\_add\_spector на 1. В противном случае, функция удаляет блик и вызывает функцию удаления цветового спектра DeleteColorSpector (код на рисунке 108).

```
361 ▼ function CreateColorSpector(){
362 ▼     if (distance==1) {
363         DeleteColorSpector();
364         DinamLine.add(LftColorSpector250);
365         DinamLine.add(RghtColorSpector250);
366     }
367 ▼     if (distance==2) {
368         DeleteColorSpector();
369         DinamLine.add(LftColorSpector180);
370         DinamLine.add(RghtColorSpector180);
371     }
372 ▼     if (distance==3) {
373         DeleteColorSpector();
374         DinamLine.add(LftColorSpector110);
375         DinamLine.add(RghtColorSpector110);
376     }
377 }
```

Рисунок 107 – Код функции CreateColorSpector

```
378 ▼ function DeleteColorSpector(){
379     DinamLine.remove(LftColorSpector250);
380     DinamLine.remove(RghtColorSpector250);
381     DinamLine.remove(LftColorSpector180);
382     DinamLine.remove(RghtColorSpector180);
383     DinamLine.remove(LftColorSpector110);
384     DinamLine.remove(RghtColorSpector110);
385 }
```

Рисунок 108 – Код функции DeleteColorSpector

Первая функция при каждом вызове обращается к функции удаления цветового спектра после чего, в зависимости от положения динамической линейки (экрана) добавляет в ее группу, панели с соответствующей текстурой цветового спектра, вторая – при каждом вызове удаляет из группы все панели цветового спектра.

На рисунке 109 показан скриншот демонстрации полихромного цветового спектра для разных положений экрана на линейке.



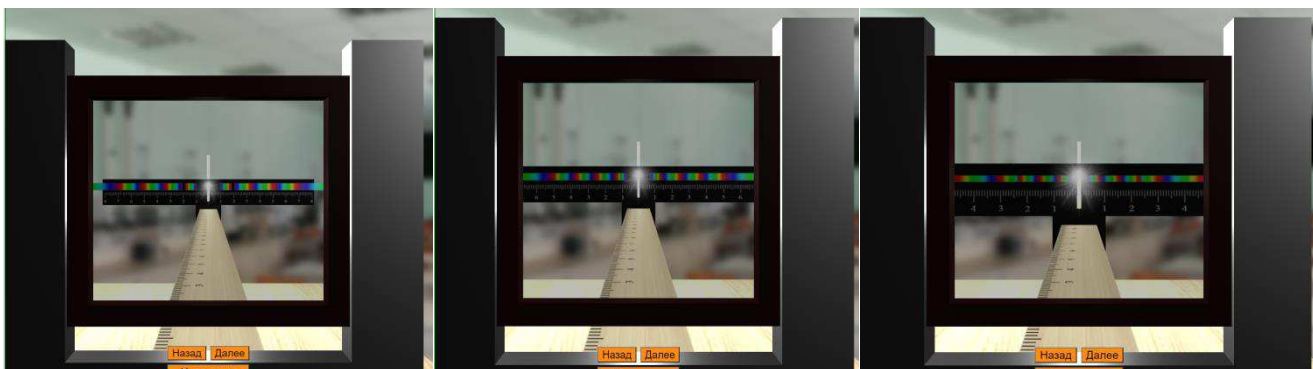


Рисунок 109 – Скриншот полихромного цветового спектра при разных положениях экрана

Имитация появления на экране монохромного цветового спектра описана анимированной функцией `AddLaserSpector`, код которой показан на рисунке 110.

```

352 function AddLaserSpector(){
353     if (flag_laser==1) {
354         CreatelaserSpector();
355     } else {
356         DeletelaserSpector();
357     }
358 }

```

Рисунок 110 – Код функции `AddLaserSpector`

Данная функция при включении лазера вызывает функцию создания монохромного спектра `CreateLaserSpector`, при отключении лазера – функцию `DeleteLaserSpector`. Код этих функций показан на рисунке 111.

```

486 function CreatelaserSpector(){
487     var numPoint=0;
488     var PointColors = [0xff0000, 0x00ff00];
489     var distScaleCf = 0.01;
490     var lineScaleCf = 0.035;
491     var dist = -26.07*DynamLinePositionZ*142.59;
492     var gratings = [0.01, 0.004];
493     var spectorLineLengths = [650, 500, 380];
494     for (var i = 0; i < 2; i++) {
495         var positionSpectorLine = lineScaleCf*((i+1)*spectorLineLengths[flag_laser_color]*1000000*(distScaleCf*dist)/gratings[flag_grating_1]);
496         var wightX = 0.01;
497         var positionX = positionSpectorLine*wightX;
498         var colorPoint=PointColors[flag_laser_color];
499         CreatePointColor(positionX, colorPoint, numPoint);
500         numPoint++;
501     }
502 }
503
504 function DeletelaserSpector() {
505     for (var i = 0; i < 2; i++) {
506         DynamLine.remove(rightcolorObjects[i]);
507         DynamLine.remove(leftcolorObjects[i]);
508     }
509 }

```

Рисунок 111 – Код функций `CreateLaserSpector` и `DeleteLaserSpector`

Первая функция, в зависимости от определенных условий (цвет лазера, расстояние от решетки до экрана, период дифракционной решетки), по формуле для расчета положения максимума интенсивности света на экране, циклом for рассчитывает координаты положения точек на экране (рисунок 111, строка 414) и передает их в функцию CreatePointColor (код на рисунке 112) вместе с цветом и порядком точки. Для определения цвета, длины волны этого цвета и периода решетки использованы массивы (рисунок 111, строки 407, 411 и 412).

Вторая функция, DeleteLaserSpector, при ее вызове циклом for удаляет монохромный спектр (точки лазера).

```
387 function CreatePointColor(positionX, colorPoint, numPoint){
388     rghtcolorObjects[numPoint].material.color.setHex(colorPoint);
389     lftcolorObjects[numPoint].material.color.setHex(colorPoint);
390     var positionZ;
391     if (positionX>2.42) {
392         positionZ=-1000;
393     }
394     else {
395         positionZ=0.26;
396     }
397     rghtcolorObjects[numPoint].position.set(positionX, 0.476, positionZ);
398     lftcolorObjects[numPoint].position.set(-positionX, 0.476, positionZ);
399     DinamLine.add(rghtcolorObjects[numPoint]);
400     DinamLine.add(lftcolorObjects[numPoint]);
401 }
```

Рисунок 112 – Код функции CreatePointColor

Функция CreatePointColor, в зависимости от полученных переменных цвета, положения на сцене, а также порядковом номере точки меняет свойства 3D объекта. Кроме этого, функция проверяет, чтобы панели, имитирующие точки лазера не «висели в воздухе» за пределами динамической линейки и, если они находятся за ее пределами, то их координаты корректируются так, чтобы их не было видно (рисунок 112, строки 391-396).

Все параметры лабораторной работы 2 рассчитаны заранее и максимально приближают виртуальный физический процесс к реальному.

Все функции, производящие динамические изменения параметров 3D объектов, как и в первой лабораторной работе, вызываются функцией update, код которой показан на рисунке 113.

```

659  function update(){
660      Lamp_OnOf();
661      AddColorSpector();
662      AddLaserspector();
663      movDinamLine();
664      movingCamera();
665      render();
666      requestAnimationFrame(update);
667  }
668
669  function render()
670  {
671      renderer.render(scene, camera);
672  }

```

Рисунок 113 – Код функций update и render

Она вызывает себя функцией `requestAnimationFrame()` обработки 3D объектов в анимации примерно 60 раз в секунду, а также вызывает функции `Lamp_OnOf`, `AddColorSpector`, `AddLaserSpector`, `movDinamLine`, `movingCamera`, и `render`.

Функция `render` создана для рендеринга сцены и камеры, а поскольку в приложении используется анимация, то и рендеринг должен быть постоянным, поэтому он вызывается в функции `update`. На рисунках 114-115 показаны варианты отображения цветового спектра лазера при разном положении динамической линейки, значении дифракционной решетки и цвете лазера.



Рисунок 114 – Скриншот экрана при включении монохроматического спектра (красный лазер)



Рисунок 115 – Скриншот экрана при включении монохроматического спектра (зеленый лазер)

Пользователь непосредственно может наблюдать изменение дифракционной картины (расположения точек лазера) на экране при перемещении экрана вдоль линейки.

А также изменения дифракционной картины в зависимости от выбранной решетки и цвета лазера.

### **2.2.5 Создание имитационной модели физического процесса лабораторной работы «Изучение неупругого удара»**

В случае выбора пользователем третьей лабораторной работы, файл process.php, также как и для предыдущих работ генерирует левое и правое меню на странице отображения физического процесса (рисунки 116 и 117).

Так как, в этой лабораторной работе не предполагается выбора исходных параметров установки, левое меню работы содержит лишь информацию об этих параметрах и инструкцию к выполнению. Правое меню содержит кнопки управления винтовкой, маятником, а также, кнопку переключения положения камеры. На рисунке 118 показаны скриншоты этих меню на веб-странице.

```

166 <?php }
167 if ($id_lr==3) { ?>
168 <div class="leftcontainer">
169 <script type="module" src="js/mainapp<?php echo($id_lr) ?>.js"></script>
170 <div class="leftupcontainer">
171 <h1>Параметры для выполнения работы</h1>
172 </div>
173 <div class="instruction">
174 <div class="recom">
175 <p>Масса цилиндра М = 350 гр.</p>
176 <p>Масса пули m = 0,5 гр.</p>
177 <p>Длина нити l = 2 м.</p>
178 </div>
179 <h1>Инструкция к выполнению работы</h1>
180 <p>1. Зарядите пневматическую винтовку</p>
181 <p>2. Произведите выстрел</p>
182 <p>3. Зафиксируйте расстояние, на которое отклонился груз</p>
183 <p>4. Повторите опыт несколько раз.</p>
184 </div>
185 </div>
186 <?php } ?>

```

Рисунок 116 – Код левого меню лабораторной работы 3

```

233 <?php }
234 if ($id_lr==3) { ?>
235 <div class="buttoncontainer">
236 <div class="button">
237 <form name="startProcess1Form" method="post">
238 <input id="btnGun" name="processStarter1" type="button" value="Зарядить" class="input2" onclick="clickt_gun()">
239 </form>
240 <form name="startProcess2Form" method="post">
241 <input name="processStarter2" type="button" value="Сделать выстрел" class="input2" disabled onclick="clickt()">
242 </form>
243 <form name="stopProcessForm" method="post">
244 <input name="processStoper" type="button" value="Остановить маятник" class="input2" onclick="stop_system()">
245 </form>
246 <form name="startCameraForm" method="post">
247 <input name="processCameraStarter" type="button" value="Переключить камеру" class="input2" onclick="clicktCamera()">
248 </form>
249 </div>
250 </div>
<?php } ?>

```

Рисунок 117 – Код правого меню лабораторной работы 3

**Параметры для выполнения работы**

Масса цилиндра  $M = 350$  гр.

Масса пули  $m = 0,5$  гр.

Длина нити  $l = 2$  м.

**Инструкция к выполнению работы**

1. Зарядите пневматическую винтовку
2. Произведите выстрел
3. Зафиксируйте расстояние, на которое отклонился груз
4. Повторите опыт несколько раз.

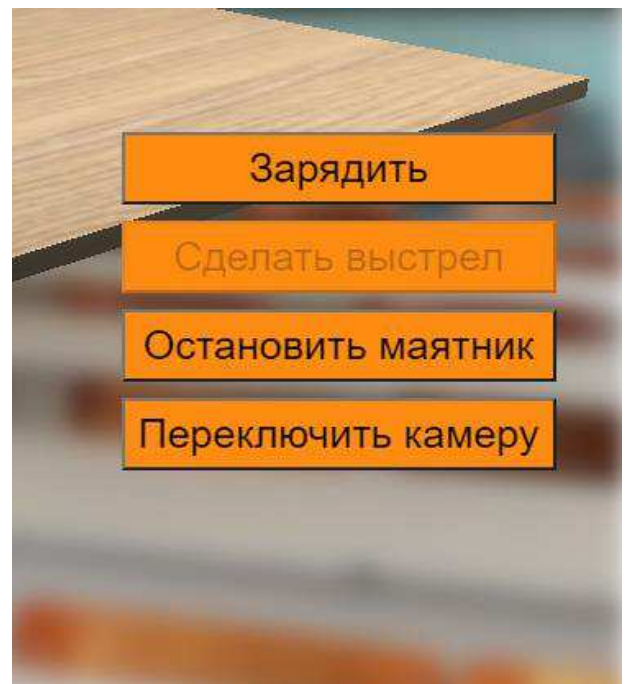


Рисунок 118 – Скриншот левого и правого меню лабораторной работы 3

Имитационная установка лабораторной работы «Изучение неупругого удара» состоит из некоторой поверхности (стола), баллистического маятника, который представляет собой штатив, с подвешенным на нити и наполненным пластилином цилиндром, пневматического ружья, пули. На рисунке 119 скриншот виртуальной лабораторной установки.

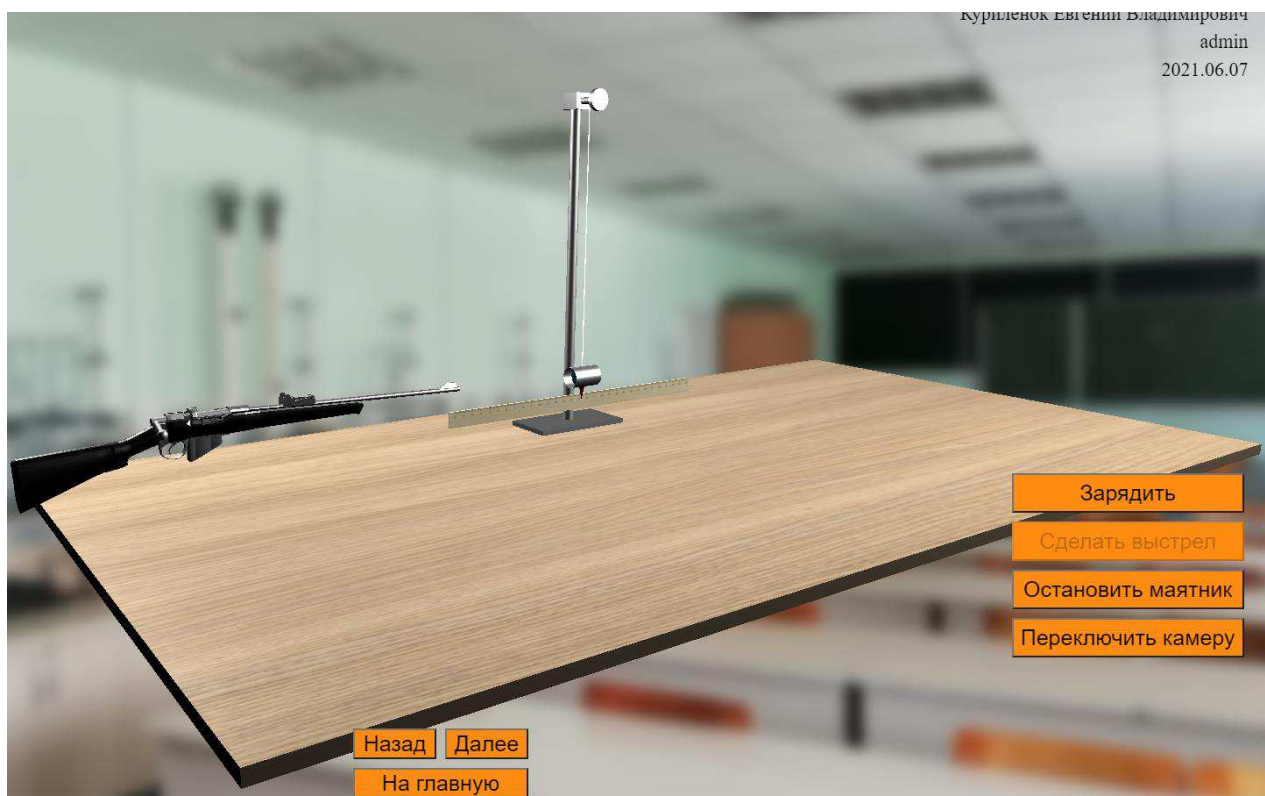


Рисунок 119 – Скриншот лабораторной установки

Для реализации этой виртуальной установки в файле `mainapp3`, аналогично, как и в двух предыдущих работах объявляются глобальные переменные, создается виртуальная сцена со светом и камерой, создаются соответствующие этой работе 3D объекты.

На рисунке 120 показан код объявления глобальных переменных и создания сцены.

Переменная `deg` создана для перевода значений углов поворота объектов из радиан в градусы.

Так же, как и во второй работе, объекты объединены в группы, для

удобства работы с ними. Кроме того, в этой лабораторной работе будет использована внешняя 3D модель винтовки, поэтому для ее добавления на сцену необходимо импортировать загрузчик объектов формата GLTF (рисунок 120, строка 18). Функция `init` состоит из аналогичных второй и третьей работе функций.

```
mainapp3.js
1
2
3 var scene;
4 var camera;
5 var light;
6 var renderer, container;
7 var Sphere;
8 var materialMetal, materialTable;
9 var guns;
10 var deg=Math.PI/180;
11
12 var System = new THREE.Group();
13 var Sht = new THREE.Group();
14 var DinamBox = new THREE.Group();
15 var Gun = new THREE.Group();
16 var Bolt = new THREE.Group();
17
18 import { GLTFLoader } from './js/threejs/examples/jsm/loaders/GLTFLoader.js';
19 init();
20
21 function init(){
22     CreateScene();
23     CreateObjects();
24     update();
25 }
26 function CreateScene(){
27     scene = new THREE.Scene();
28
29     camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 1, 10000 );
30     camera.position.set(-63.9, 27, 63.7);
31     camera.rotation.set(-13.81*Math.PI/180, -31.12*Math.PI/180, -7.24*Math.PI/180);
32
33     var light1 = new THREE.PointLight( 0xffffff);
34     var light2 = new THREE.PointLight( 0xffffff);
35     light1.position.set( 50, 50, 50);
36     light2.position.set( -50, 50, -50);
37
38     scene.add(camera);
39     scene.add(light1);
40     scene.add(light2);
41
42     renderer = new THREE.WebGLRenderer({ alpha: true });
43     renderer.setClearColor(0x000000, 0);
44     renderer.setSize(window.innerWidth, window.innerHeight);
45     renderer.shadowMapEnabled = true;
46
47     container = document.getElementById("MyWebGLApp");
48     container.appendChild( renderer.domElement );
49
50     window.addEventListener('resize', onWindowResize, false );
51 }
52
53 function onWindowResize()
54 {
55     camera.aspect = window.innerWidth / window.innerHeight;
56     camera.updateProjectionMatrix();
57     renderer.setSize(window.innerWidth, window.innerHeight);
58 }
```

Рисунок 120 – Код объявления глобальных переменных и создания сцены

На рисунке 121 показан код создания геометрии и материалов основных объектов сцены.

```
60 function CreateObjects(){
61 // Геометрия основных объектов
62 var geometryTable = new THREE.BoxGeometry(100, 1, 50);
63 var geometrySphere = new THREE.SphereGeometry(0.16, 8, 8);
64
65 var geometryShtDwnBox = new THREE.BoxGeometry(10, 0.5, 5);
66 var geometryShtCylinder = new THREE.CylinderGeometry(0.5, 0.5, 30, 12, 1, true);
67 var geometryShtLineCylinder = new THREE.CylinderGeometry(0.2, 0.2, 1, 8, 1, true);
68 var geometryShtLineBox = new THREE.BoxGeometry(30, 1.5, 0.2);
69 var geometryShtUpBox = new THREE.BoxGeometry(1.5, 1.5, 2);
70 var geometryShtUpCylinder = new THREE.CylinderGeometry(0.3, 0.3, 2, 8, 1, true);
71 var geometryShtUpCylinder2 = new THREE.CylinderGeometry(1, 1, 0.5, 16, 1, false);
72
73 var geometrySystCylinder = new THREE.CylinderGeometry(0.05, 0.05, 25, 8, 1, true);
74
75 var geometryBoxCylinder = new THREE.CylinderGeometry(1, 1, 3, 24, 1, true);
76 var geometryBoxBackCylinder = new THREE.CylinderGeometry(1, 1, 0.1, 24, 1, false);
77 var geometryBoxInCylinder = new THREE.CylinderGeometry(0.98, 0.98, 1, 24, 1, false);
78 var geometryBoxPointerCylinder = new THREE.CylinderGeometry(0.3, 0.02, 1.5, 16, 1, false);
79
80 //Текстуры
81 var TextureTable = new THREE.ImageUtils.loadTexture( 'image/textureTableLr1.jpeg' );
82 var TextureLine = new THREE.ImageUtils.loadTexture( 'image/textureLine2.jpg' );
83 var TextureShtLineBox = new THREE.ImageUtils.loadTexture( 'image/textureShtBox.jpg' );
84
85 //Материалы
86 var materialTable = new THREE.MeshPhongMaterial( { map: TextureTable} );
87 var ShtLineBoxMaterials = [
88 new THREE.MeshPhongMaterial( { map: TextureShtLineBox } ),
89 new THREE.MeshPhongMaterial( { map: TextureShtLineBox } ),
90 new THREE.MeshPhongMaterial( { map: TextureShtLineBox } ),
91 new THREE.MeshPhongMaterial( { map: TextureShtLineBox } ),
92 new THREE.MeshPhongMaterial( { map: TextureLine } ),
93 new THREE.MeshPhongMaterial( { map: TextureShtLineBox } )
94 ]
95 var materialShtLineBox = new THREE.MeshFaceMaterial(ShtLineBoxMaterials);
96
97 materialMetal = new THREE.MeshStandardMaterial({
98 color: 0xffffff,
99 roughness: 0.5,
100 metalness: 0.8});
101 var materialMetalBoxCylinder = new THREE.MeshStandardMaterial({
102 color: 0xffffff,
103 roughness: 0.5,
104 metalness: 0.8});
105 var materialPlastic = new THREE.MeshPhongMaterial({color: 0x000000, specular: 0x6b6b6b});
106 var materialRedPlastic = new THREE.MeshPhongMaterial({color: 0x6f0b0b, specular: 0x6b6b6b});
107 var materialInCylinder = new THREE.MeshStandardMaterial({color: 0x1a2a47});
108 var materialThread = new THREE.MeshBasicMaterial({color: 0xffffffff});
109
110 materialMetalBoxCylinder.side = THREE.DoubleSide;
111
```

Рисунок 121 – Код создания геометрии и материалов объектов

На рисунке 122 показан код создания объектов с использованием геометрии и материалов, а также, расположение этих объектов на сцене и добавление их в группы.



```

112 //Создание объектов с геометрией и материалами
113 var Table = new THREE.Mesh(geometryTable, materialTable);
114 Sphere = new THREE.Mesh(geometrySphere, materialMetal);
115 var ShtDownBox = new THREE.Mesh(geometryShtDownBox, materialMetal);
116 var ShtCylinder = new THREE.Mesh(geometryShtCylinder, materialMetal);
117 var ShtLineCylinder = new THREE.Mesh(geometryShtLineCylinder, materialMetal);
118 var ShtLineBox = new THREE.Mesh(geometryShtLineBox, materialShtLineBox);
119 var ShtUpBox = new THREE.Mesh(geometryShtUpBox, materialPlastic);
120 var ShtUpCylinder = new THREE.Mesh(geometryShtUpCylinder, materialMetal);
121 var ShtUpCylinder2 = new THREE.Mesh(geometryShtUpCylinder2, materialMetal);
122
123 var SystCylinder = new THREE.Mesh(geometrySystCylinder, materialThread);
124
125 var BoxCylinder = new THREE.Mesh(geometryBoxCylinder, materialMetalBoxCylinder);
126 var BoxBackCylinder = new THREE.Mesh(geometryBoxBackCylinder, materialMetal);
127 var BoxInCylinder = new THREE.Mesh(geometryBoxInCylinder, materialInCylinder);
128 var BoxPointerCylinder = new THREE.Mesh(geometryBoxPointerCylinder, materialRedPlastic);
129
130 //Положение объектов в группе
131 Table.position.set(0, 1.1, 0);
132
133 ShtDownBox.position.set(0, 0, 0);
134 ShtCylinder.position.set(0, 15.3, 0);
135 ShtLineCylinder.position.set(0, 2.4, 0.93);
136 ShtLineBox.position.set(0, 2.4, 1.43);
137 ShtUpBox.position.set(0, 31, 0);
138 ShtUpCylinder.position.set(0, 31, 2);
139 ShtUpCylinder.rotation.set(Math.PI/2, 0, 0);
140 ShtUpCylinder2.position.set(0, 31, 3.2);
141 ShtUpCylinder2.rotation.set(Math.PI/2, 0, 0);
142
143 SystCylinder.position.set(0, -12.5, 0);
144
145 BoxCylinder.position.set(0, 0, 0);
146 BoxBackCylinder.position.set(0, -1.54, 0);
147 BoxInCylinder.position.set(0, 0, 0);
148 BoxPointerCylinder.position.set(-1.5, 0, 0);
149 BoxPointerCylinder.rotation.set(0, 0, -Math.PI/2);
150
151 // Добавление объектов в группы, положение групп на сцене
152 DinamBox.add(BoxCylinder);
153 DinamBox.add(BoxBackCylinder);
154 DinamBox.add(BoxInCylinder);
155 DinamBox.add(BoxPointerCylinder);
156 DinamBox.position.set(0, -26, 0);
157 DinamBox.rotation.set(0, 0, Math.PI/2);
158
159 System.add(SystCylinder);
160 System.add(DinamBox);
161 System.position.set(0, 31, 2);
162
163 Sht.add(ShtDownBox);
164 Sht.add(ShtCylinder);
165 Sht.add(ShtLineCylinder);
166 Sht.add(ShtLineBox);
167 Sht.add(ShtUpBox);
168 Sht.add(ShtUpCylinder);
169 Sht.add(ShtUpCylinder2);
170 Sht.add(System);
171 Sht.position.set(0, 1.8, -15);

```

Рисунок 122 – Код создания и размещения на сцене объектов

После создания основных объектов, нужно загрузить заранее подготовленную и помещенную в папку models внешнюю 3D модель с помощью загрузчика GLTFLoader.

На рисунке 123 показан код этого процесса.

```

173 //Загрузка внешней 3D модели
174 const loader = new GLTFLoader();
175
176 loader.load( 'models/gun.glb', function ( gltf ) {
177
178     guns = gltf.scene;
179     var deletBolt;
180     var gun = guns.getObjectByName('Gun_Stock');
181     var deleteBolt = guns.getObjectByName('Bolt');
182     var deleteBolt2 = guns.getObjectByName('Firing_Pin');
183     gltf.scene.remove(deleteBolt);
184     gltf.scene.remove(deleteBolt2);
185     guns.position.set(-29, -13, -23.9)
186     guns.scale.set(4, 4, 4);
187     guns.rotation.set(0, Math.PI/2, 0);
188     guns.traverse( function ( obj ) {
189
190         if (obj==guns.getObjectByName('Gun_Action')) {
191             obj.material = materialMetal;
192         }
193         if (obj==guns.getObjectByName('Cylinder004')) {
194             obj.material = materialPlastic;
195         }
196     } );
197     scene.add(guns);
198 }, undefined, function ( error ) {
199     console.error( error );
200 } );

```

Рисунок 123 – Код загрузки внешней 3D модели

Модель ружья представлена на рисунке 124, заимствована модель на сайте <https://free3d.com/>.

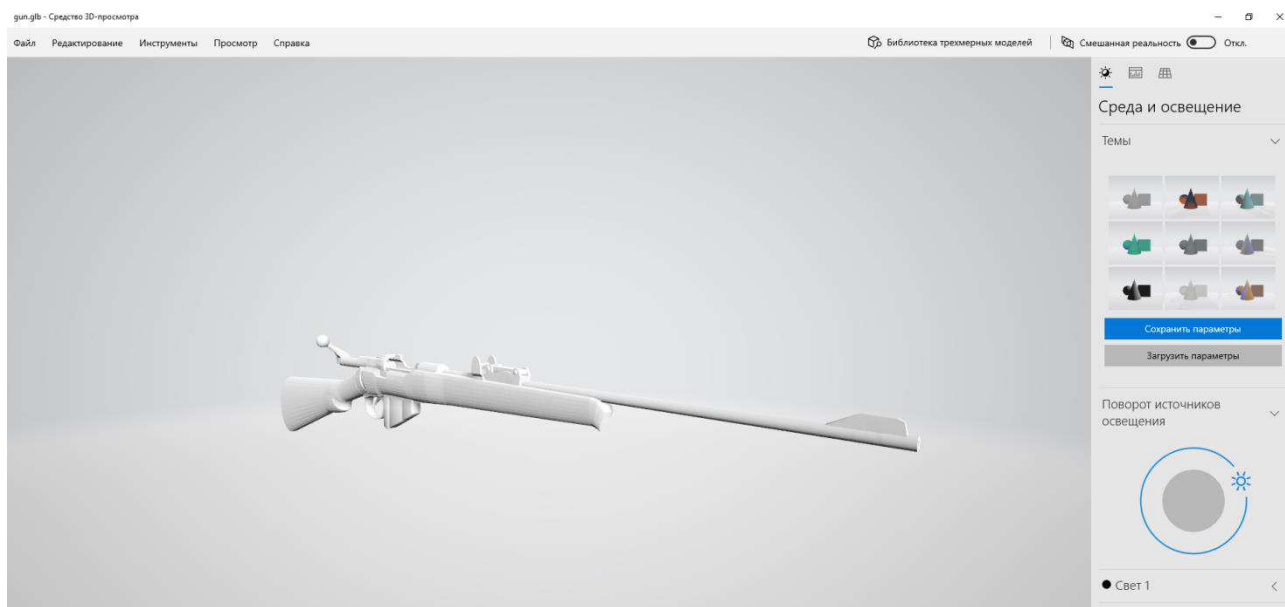


Рисунок 124 – Скриншот 3D-модели винтовки

Модель ружья, внедренная на сцену, представлена на рисунке 125.



Рисунок 125 – Модель винтовки, добавленная на сцену лабораторной работы 3

В процессе загрузки были изменены некоторые свойства этого 3D объекта, например, изменены его размеры, положение на сцене, наложены материалы, а также, удалена часть модели объекта, поскольку далее эта часть будет анимирована, а для удобства анимации, объект будет создан заново, вне загруженной модели. На рисунке 126 показан процесс создания этого дополнительного объекта, после чего, все группы объектов добавлены на сцену.

```
201
202 // Создание дополнительного объекта для 3д модели
203 var geometryBoltCylinder1 = new THREE.CylinderGeometry(0.3, 0.3, 7, 12);
204 var geometryBoltCylinder2 = new THREE.CylinderGeometry(0.2, 0.2, 1.5, 12, 1, true);
205 var geometryBoltSphere = new THREE.SphereGeometry(0.4, 16, 16);
206
207 var BoltCylinder1 = new THREE.Mesh(geometryBoltCylinder1, materialMetal);
208 var BoltCylinder2 = new THREE.Mesh(geometryBoltCylinder2, materialMetal);
209 var BoltSphere = new THREE.Mesh(geometryBoltSphere, materialMetal);
210
211 BoltCylinder1.rotation.set(0, 0, Math.PI/2);
212 BoltCylinder1.position.set(0, 0, 0);
213 BoltCylinder2.position.set(-3.2, 0.9, 0);
214 BoltSphere.position.set(-3.2, 1.9, 0);
215
216 Bolt.add(BoltCylinder1);
217 Bolt.add(BoltCylinder2);
218 Bolt.add(BoltSphere);
219 Bolt.position.set(-38.7, 7.8, -13);
220 Bolt.rotation.set(110*Math.PI/180, 0, 0);
221
222 //добавление групп на сцену
223 scene.add(Table);
224 scene.add(Sht);
225 scene.add(Bolt);
226 }
```

Рисунок 126 – Код создания дополнительного объекта винтовки и добавления групп на сцену

Для создания имитации процесса лабораторной работы «Изучение неупругого удара» использовался импорт загрузчика, поэтому файл `mainapp3` подключается к веб-странице с типом `module`. Кнопки веб-страницы не могут управлять функциями этого типа `js`-файла, а значит, пользователь не сможет управлять процессом выполнения работы. Чтобы избежать этого, был создан файл `startedFunctionLr3.js` типа `text/javascript`, в котором написан код (рисунок 127), для управления процессом кнопками «Зарядить», «Сделать выстрел», «Остановить маятник», «Переключить камеру» (рисунок 128).

```
startedFunctionLr3.js
1  var flag_gan=0;
2  var flag_shot=0;
3  var flag_camera=0;
4  var flag_stop=0;
5  function clickt(){
6      if (flag_shot==0&&flag_gan==1) {
7          flag_shot=1;
8          flag_gan=0;
9          document.startProcess1Form.processStarter1.value="Зарядить";
10         document.startProcess1Form.processStarter1.disabled=true;
11         document.startProcess2Form.processStarter2.disabled=true;
12     }
13 }
14 function clickt_gun(){
15     if (flag_gan==0) {
16         flag_gan=1;
17         document.startProcess1Form.processStarter1.value="Разрядить";
18     } else {
19         flag_gan=0;
20         document.startProcess1Form.processStarter1.value="Зарядить";
21     }
22 }
23 function clicktCamera(){
24     if (flag_camera==0) {
25         flag_camera=1;
26     } else {
27         flag_camera=0;
28     }
29 }
30 function stop_system(){
31     if (flag_stop==0) {
32         flag_stop=1;
33     } else {
34         flag_stop=0;
35     }
36 }
```

Рисунок 127 – Код файла `startedFunctionLr3`

Данный код состоит из функций, изменяющих глобальные переменные, видимые файлом `mainapp3`, которые влияют на виртуальный физический

процесс.

Аналогично второй работе, создается механизм изменения положения камеры, состоящий из нескольких функций, одна из которых регулярно вызывается функцией анимации update. Код этого механизма показан на рисунке 128.

```
236 ▼ function movCameraDown(){
237 ▼     if (camera.position.x<-2) {
238         camera.position.x += 0.6;
239         camera.position.y += -0.156;
240         camera.position.z += -0.426;
241         camera.rotation.x += 0.0858*Math.PI/180;
242         camera.rotation.y += 0.303*Math.PI/180;
243         camera.rotation.z += 0.0708*Math.PI/180;
244 ▼     } else {
245         camera.position.x = -2;
246         camera.position.y = 10;
247         camera.position.z = 20;
248         camera.rotation.x = -5*Math.PI/180;
249         camera.rotation.y = 0;
250         camera.rotation.z = 0;
251     }
252 }
253 ▼ function movCameraUp(){
254 ▼     if (camera.position.x>-63.9) {
255         camera.position.x += -0.6;
256         camera.position.y += 0.156;
257         camera.position.z += 0.426;
258         camera.rotation.x += -0.0858*Math.PI/180;
259         camera.rotation.y += -0.303*Math.PI/180;
260         camera.rotation.z += -0.0708*Math.PI/180;
261 ▼     } else {
262         camera.position.x = -63.9;
263         camera.position.y = 27;
264         camera.position.z = 63.7;
265         camera.rotation.x = -13.81*Math.PI/180;
266         camera.rotation.y = -31.12*Math.PI/180;
267         camera.rotation.z = -7.24*Math.PI/180;
268     }
269 }
270 ▼ function movingCamera(){
271     if (flag_camera==1) {
272         movCameraDown();
273     } else {
274         movCameraUp();
275     }
276 }
```

Рисунок 128 – Код механизма изменения положения камеры

На рисунке 129 показано изменение положения камеры для лабораторной работы «Изучение неупругого удара».



Рисунок 129 – Скриншоты страниц с различным положением камеры в лабораторной работе 3

Для большего реализма, в этой лабораторной работе была добавлена анимация перезарядки пневматической винтовки. Для этого были созданы необходимые переменные (рисунок 130).

```

278 var v=0;
279 var v_bolt=0.08;
280 var a=0.05;
281 var flag_position2_sphere=0;
282 var flag_add_sphere=0;

```

Рисунок 130 – Код создания переменных перезарядки винтовки

После чего была создана функция перезарядки AnimationLoadGan,

представляющая собой сложный механизм, проверяющий множество условий.

```
284 function AnimationLoadGan(){
285     if (flag_gan==1) {
286         if (flag_add_sphere==0) {
287             if (Bolt.rotation.x>55*deg) {
288                 Bolt.rotation.x+=-0.05;
289             } else {
290                 Bolt.rotation.x=55*deg;
291                 if (Bolt.position.x>-41.2) {
292                     Bolt.position.x+=-v_bolt;
293                 } else {
294                     Bolt.position.x=-41.2;
295                     Sphere.position.set(-37, 11, -13);
296                     scene.add(Sphere);
297                     flag_add_sphere=1;
298                 }
299             }
300         }
301         if (Sphere.position.y>7.8) {
302             Sphere.position.y += -v;
303             v+=a;
304         }
305         else if (flag_position2_sphere==0&&flag_add_sphere==1){
306             Sphere.position.y = 7.8;
307             v=0;
308             if (Bolt.position.x<-38.7) {
309                 Bolt.position.x+=v_bolt;
310             } else {
311                 Bolt.position.x=-38.7;
312                 if (Bolt.rotation.x<110*deg) {
313                     Bolt.rotation.x+=0.05;
314                 } else {
315                     Bolt.rotation.x=110*deg;
316                     flag_position2_sphere=1;
317                     document.startProcess2Form.processStarter2.disabled=false;
318                 }
319             }
320         } else if (flag_shot==0){
321             document.startProcess2Form.processStarter2.disabled=true;
322             if (flag_position2_sphere==1) {
323                 if (Bolt.rotation.x>55*deg) {
324                     Bolt.rotation.x+=-0.05;
325                 } else {
326                     Bolt.rotation.x=55*deg;
327                     if (Bolt.position.x>-41.2) {
328                         Bolt.position.x+=-v_bolt;
329                     } else {
330                         Bolt.position.x=-41.2;
331                         if (Sphere.position.y<11) {
332                             Sphere.position.y += 0.05;
333                         } else {
334                             Sphere.position.set(-37, 11, -13);
335                             scene.remove(Sphere);
336                             flag_position2_sphere=0;
337                         }
338                     }
339                 }
340             } else if (flag_position2_sphere==0&&flag_add_sphere==1){
341                 if (Bolt.position.x<-38.7) {
342                     Bolt.position.x+=v_bolt;
343                 } else {
344                     Bolt.position.x=-38.7;
345                     if (Bolt.rotation.x<110*deg) {
346                         Bolt.rotation.x+=0.05;
347                     } else {
348                         Bolt.rotation.x=110*deg;
349                         flag_add_sphere=0;
350                         v=0;
351                     }
352                 }
353             }
354         }
355     }
```

Рисунок 131 – Код функции AnimationLoadGan

Так как функция представляет процесс анимации зарядки винтовки, то

она регулярно вызывается функцией анимации update. Основной процесс начинает работать, после изменения значения переменной flag\_gun с 0 на 1, после вызова функции кнопкой «Зарядить». После этого функция проверяет значение первоначального положения пули и если оно равно 0, то значит зарядка только началась и функция меняет положение элемента затвора винтовки (рисунок 131, строки 286-293), проверяя при этом, положение этого элемента и при достижении определенных, заранее рассчитанных координат, на сцену добавляется пуля и начинается процесс ее падения (рисунок 131, строки 293-304).

После достижения пулей положения внутри винтовки, это положение фиксируется и начинается обратный процесс изменения положения затвора ружья, после остановки которого, снимается блокировка с кнопки «Сделать выстрел» (рисунок 131, строки 305-319).

Далее, код функции (рисунок 131, строки 320-354) представляет процесс, полностью противоположный зарядке и начинает работать в случае, если была нажата кнопка «Разрядить» (после зарядки кнопка «Зарядить» принимает название «Разрядить») и при этом, не был сделан выстрел.

На рисунке 132 показан скриншот процесса зарядки винтовки в разное время.



Рисунок 132 – Скриншот процесса зарядки винтовки

После того, как пользователь зарядил виртуальную пневматическую винтовку, он должен сделать выстрел, после которого пуля, достигая цилиндра маятника, заставит его делать механические затухающие колебания.



Для имитации этого, были объявлены некоторые переменные (рисунок 133), использующиеся в функции процесса выстрела.

```
357 var c=1.2;
358 var flag_position3_sphere=0;
359 var flag_position4_sphere=0;
360 var t=0;
361 var B=0.4;
362 var Amp;
363 function Random(min, max) {
364     return Math.random() * (max - min) + min;
365 }
```

Рисунок 133 – Код объявления переменных и функции Random

Кроме переменных, так же была написана небольшая функция, возвращающая случайное значение из заданного интервала чисел.

После объявления переменных была создана функция анимации процесса выстрела и последующих после него колебаний маятника AnimationShot (рисунок 134).

```
367 function AnimationShot(){
368     if (flag_shot==1) {
369         if (flag_position3_sphere==0) {
370             Sphere.position.set(-15.5, 7.22, -12.975);
371             flag_position3_sphere=1;
372             Amp=Random(23, 26);
373         }
374         if (Sphere.position.x<=-0.555&&flag_position4_sphere==0) {
375             Sphere.position.x+=c;
376         } else {
377             Sphere.position.x=-0.555;
378             flag_position4_sphere=1;
379         }
380         if (flag_position4_sphere==1) {
381             t+=0.01667;
382             scene.remove(Sphere);
383             System.rotation.z=Math.exp(-B*t)*Amp*deg*Math.cos(2.214*t-90*deg);
384         }
385     }
386 }
```

Рисунок 134 – Код функции AnimationShot

Данная функция вызывается функцией анимации update, проверяя значение переменной flag\_shot, которое меняется кнопкой «Сделать выстрел», и, если значение изменилось с 0 на 1, то начинается процесс движения пули из ствола винтовки в сторону маятника. После достижения пулей груза маятника, начинается процесс его колебаний, которые создаются расчетом из формулы положения угла в текущий момент времени (рисунок 134, строка 383). Эта формула состоит из коэффициента затухания, случайной амплитуды колебаний

из значений 23-26 градусов, и косинуса угла в момент времени  $t$ .

На рисунке 135 скриншот различных положений цилиндра после выстрела.

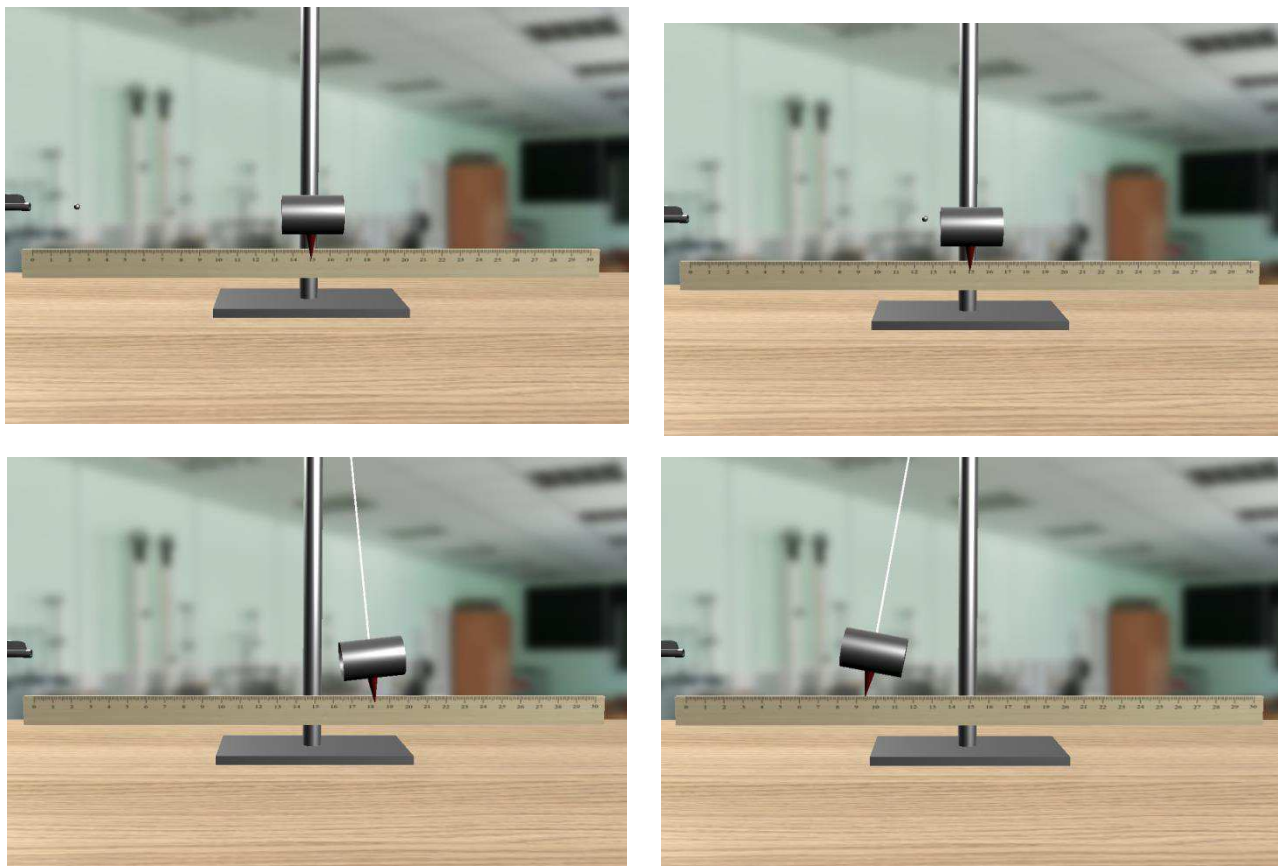


Рисунок 135 – Скриншот положений маятника

Даже в случае затухающих колебаний, маятник совершает их достаточно долго, поэтому была написана функция `reset`, останавливающая колебания либо кнопкой «Остановить маятник», либо при достижении времени 20 секунд от начала колебаний. Код этой функции на рисунке 136.

```

388 function reset(){
389     if (t>15||flag_stop==1) {
390         if (System.rotation.z<-0.1*deg||System.rotation.z>0.1*deg) {
391             System.rotation.z=Math.exp(-B*t)*Amp*deg*Math.cos(2.214*t-90*deg);
392         } else {
393             System.rotation.z=0;
394             flag_shot=0;
395             flag_add_sphere=0;
396             flag_position2_sphere=0;
397             flag_position3_sphere=0;
398             flag_position4_sphere=0;
399             t=0;
400             document.startProcess1Form.processStarter1.disabled=false;
401             flag_stop=0;
402         }
403     }
404 }

```

Рисунок 136 – Код функции reset

Все параметры этой лабораторной работы также, как и других лабораторных работ, рассчитаны заранее и максимально приближают виртуальный физический процесс к реальному.

Все функции, производящие динамические изменения параметров 3D объектов, как в первой и во второй лабораторной работе, вызываются функцией update, код которой показан на рисунке 137.

```

407 function update(){
408     movingCamera();
409     AnimationLoadGan();
410     AnimationShot();
411     reset();
412     render();
413     requestAnimationFrame(update);
414 }
415
416 function render()
417 {
418     renderer.render(scene, camera);
419 }

```

Рисунок 137 – Код функций update и render

Эта функция вызывает себя функцией обработки 3D объектов в анимации примерно 60 раз в секунду requestAnimationFrame, а также вызывает функции movingCamera, AnimationLoadGan, AnimationShot, reset и render. Функция render создана для рендеринга сцены и камеры.

### 2.3 Описание последовательности действий пользователя при выполнении виртуальных лабораторных работ

Студент или преподаватель проходит по ссылке в eКурсе дисциплины «Физика» на веб-приложение «Виртуальные лабораторные работы по физике» (рисунок 138).

Далее, студент входит или регистрируется в системе, преподаватель входит в систему под своей учетной записью (рисунки 139-140).

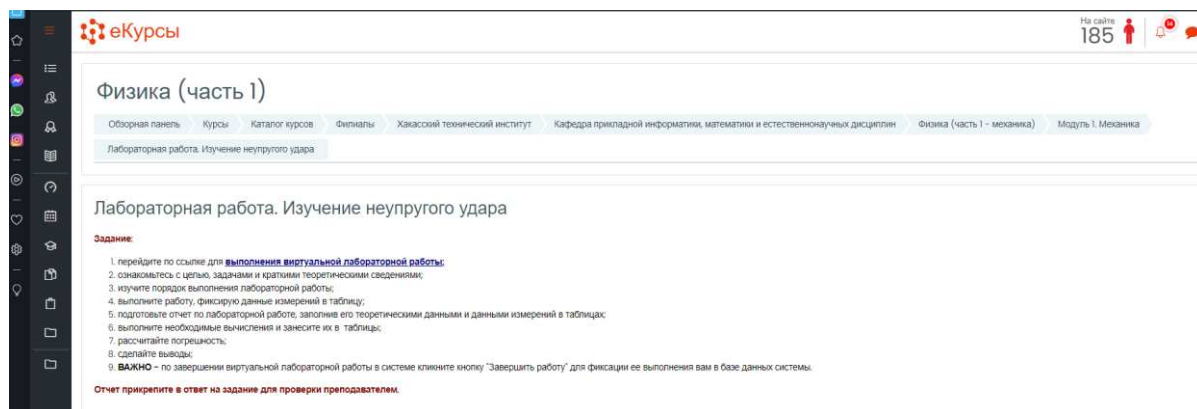


Рисунок 138 – Скриншот страницы eКурса по физике с заданием по выполнению виртуальной лабораторной работы

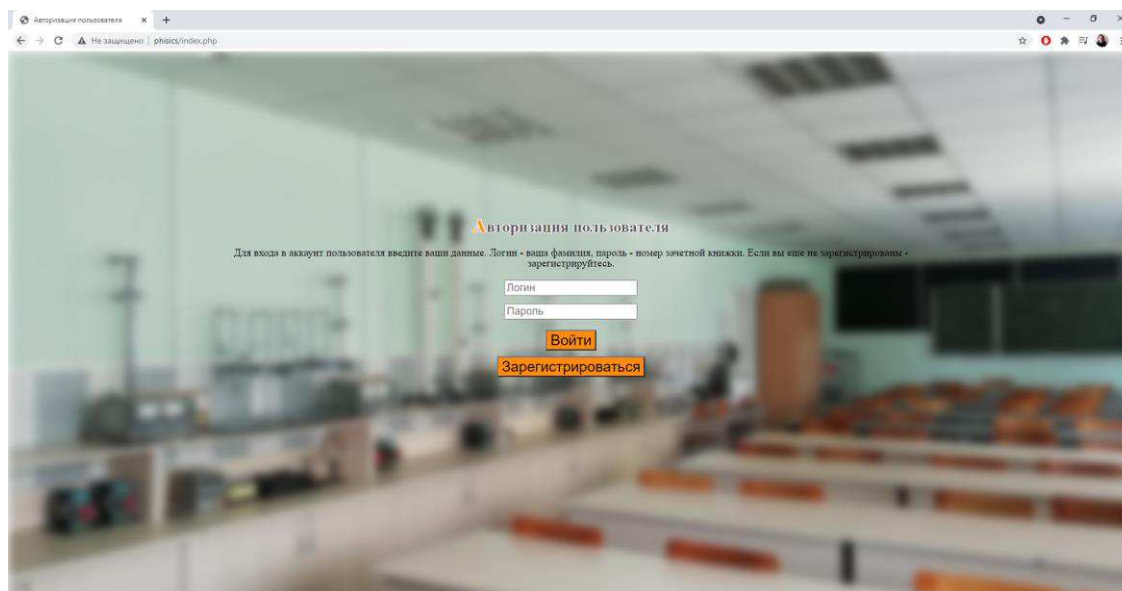


Рисунок 139 – Скриншот страницы входа в систему

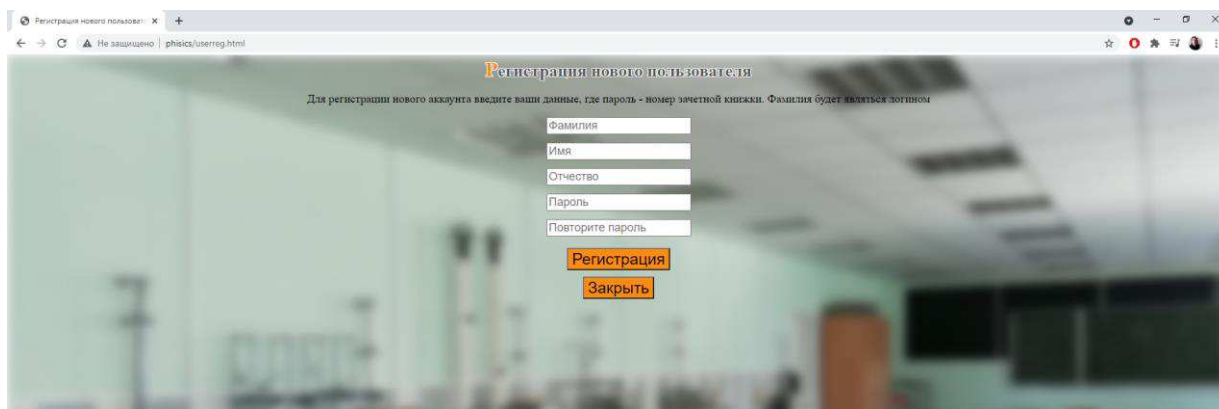


Рисунок 140 – Скриншот страницы регистрации в системе

После успешного входа в систему, перед студентом открывается меню выбора лабораторных работ, перед преподавателем открывается список студентов, выполнивших лабораторные работы, под которым находится меню выбора лабораторных работ (рисунки 141-142).

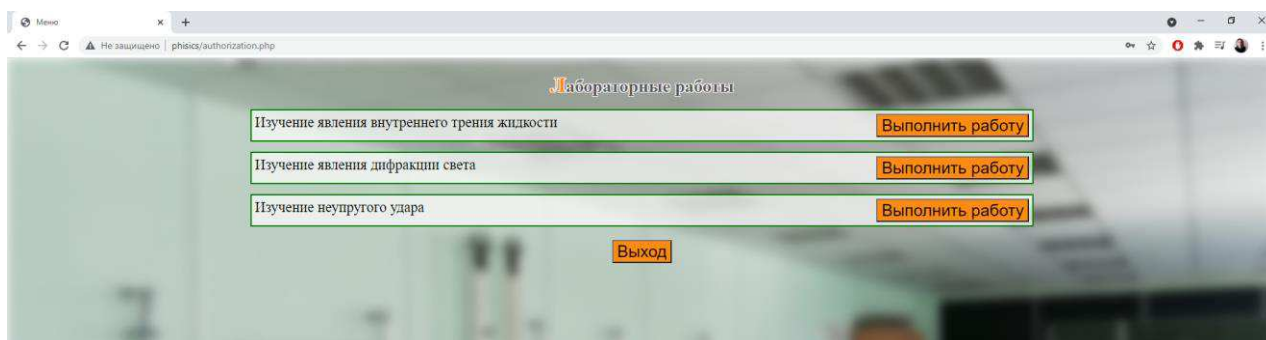


Рисунок 141 – Скриншот меню лабораторных работ для студента

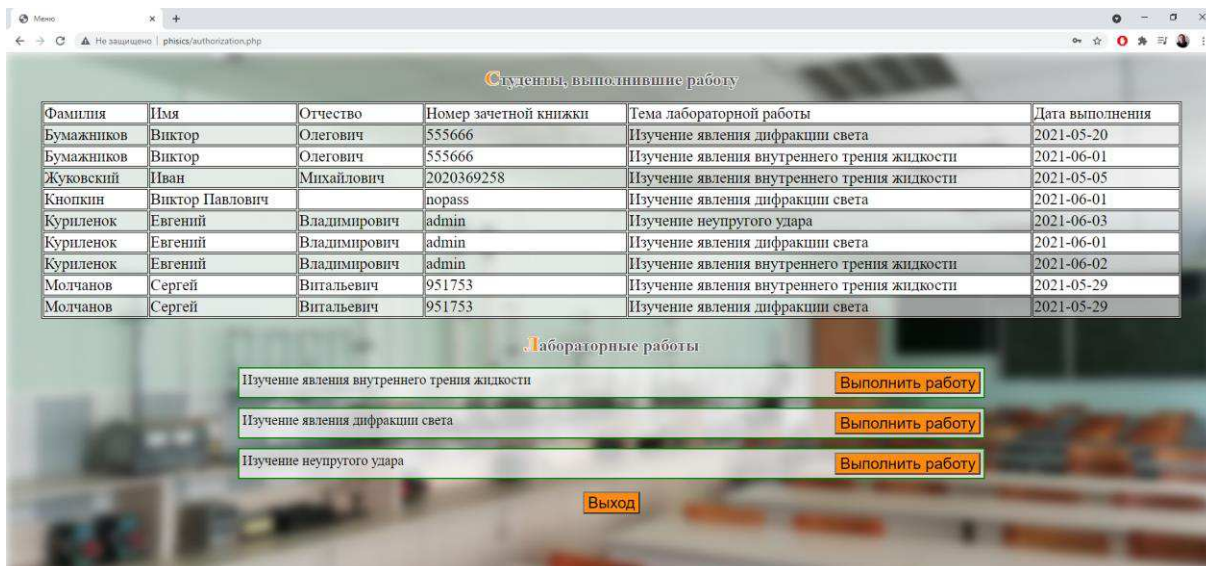


Рисунок 142 – Скриншот меню лабораторных работ для преподавателя

Для продолжения работы студент выбирает лабораторную работу, и нажимает кнопку «Выполнить работу», после чего его перенаправляет на первую страницу теории лабораторной работа, а именно – страницу цели и оборудования (рисунок 143).

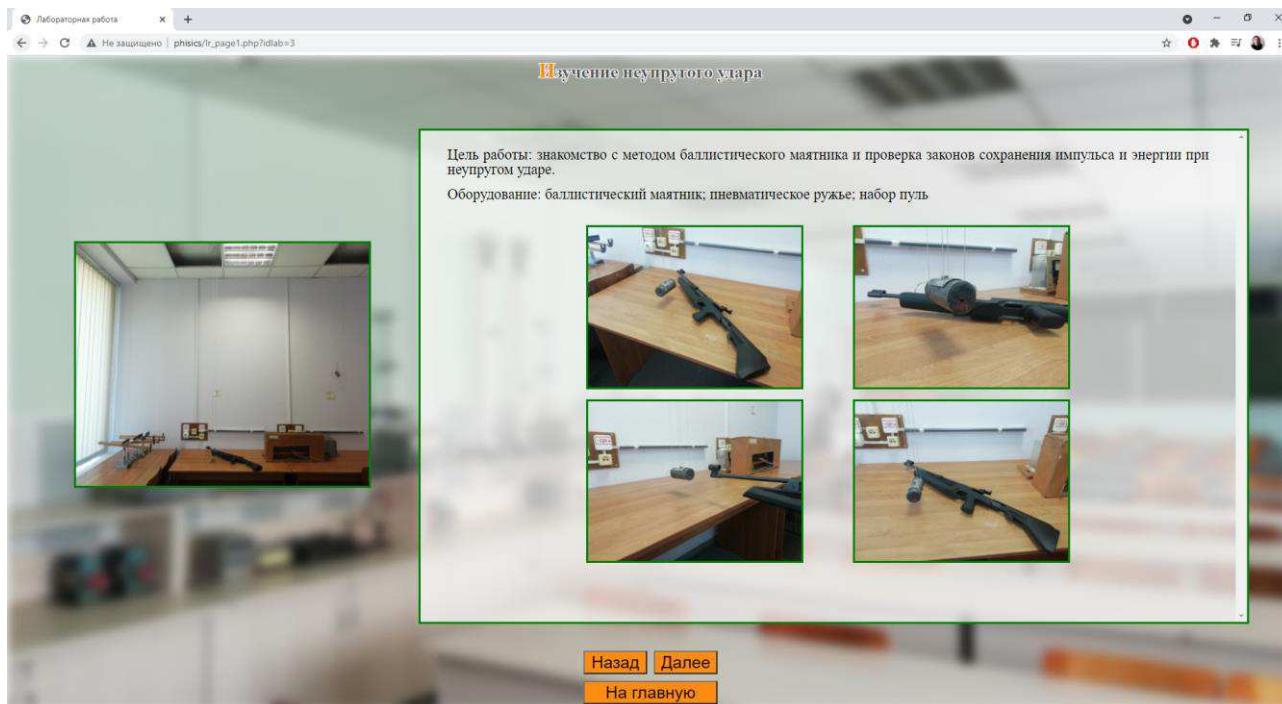


Рисунок 143 – Скриншот первой страницы выполнения лабораторной работы 3

После изучения цели, рассмотрения оборудования выбранной лабораторной работы, студент нажимает кнопку «Далее», его перенаправляет на страницу с краткими теоретическими сведениями (рисунок 144).

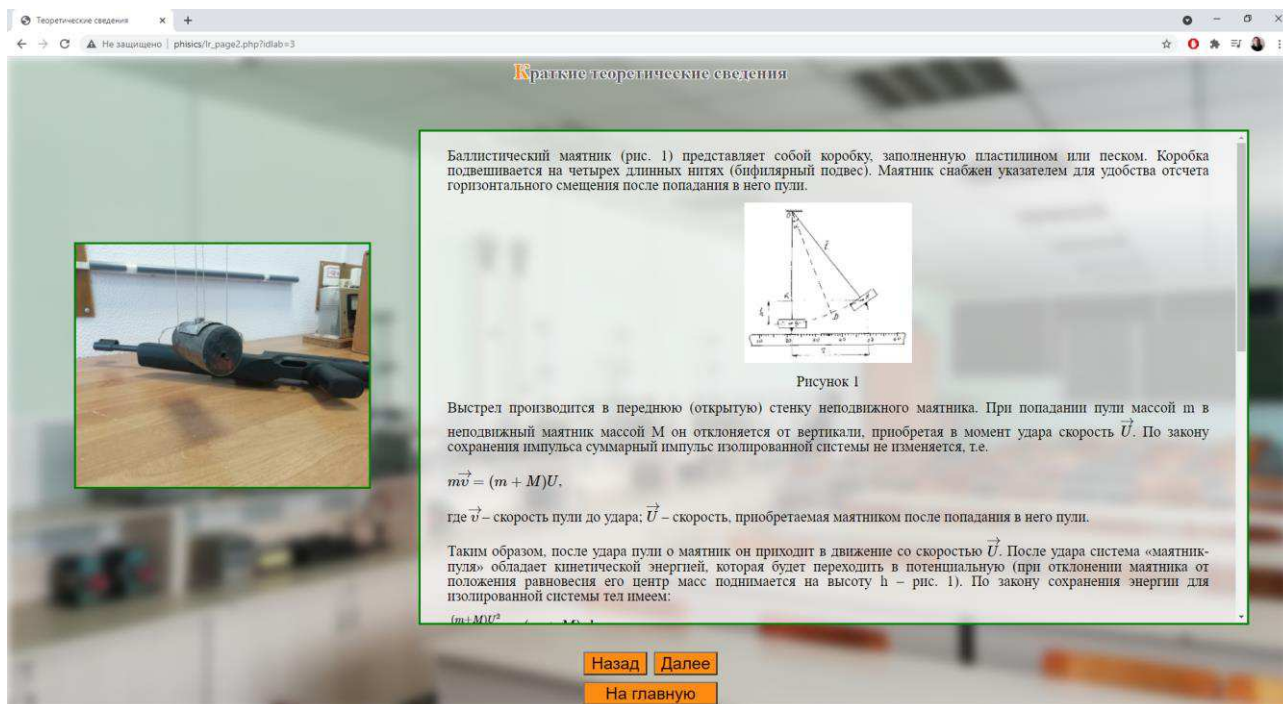


Рисунок 144 – Скриншот второй страницы выполнения лабораторной работы 3

После изучения теоретических сведений, студент также нажимает кнопку «Далее», после чего его перенаправляет на страницу с порядком выполнения этой лабораторной работы (рисунок 145).

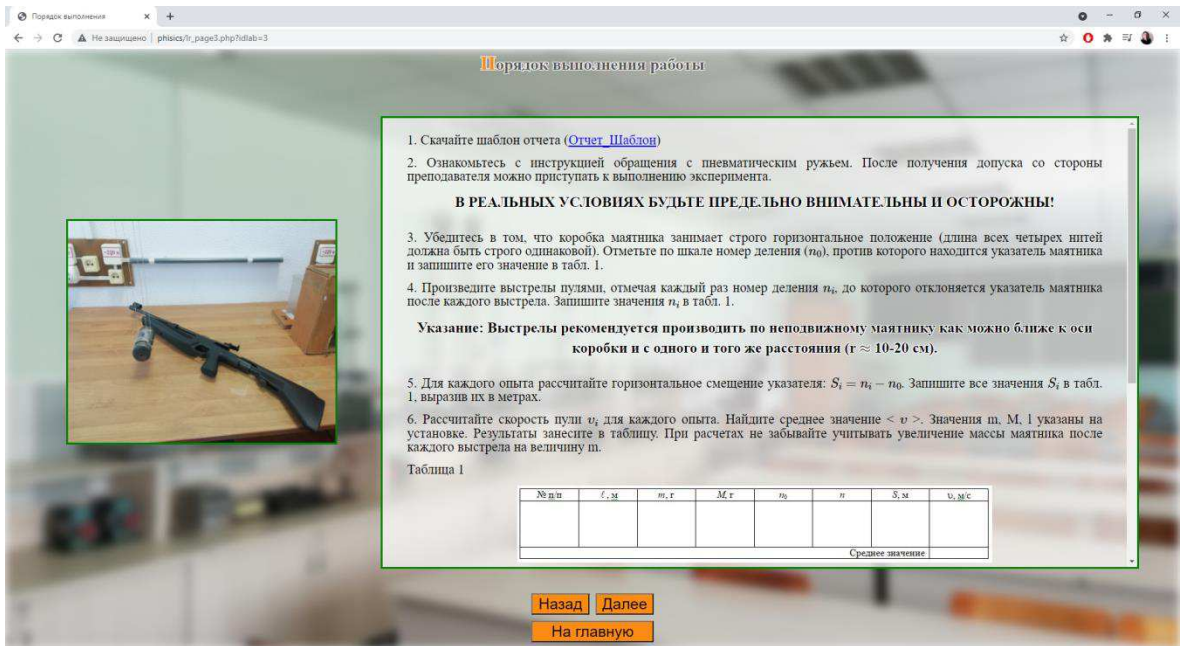


Рисунок 145 – Скриншот третьей страницы выполнения лабораторной работы 3

Далее, студенту необходимо скачать шаблон отчета (рисунок 146) по лабораторной работе и перейти на следующую страницу (рисунок 147).

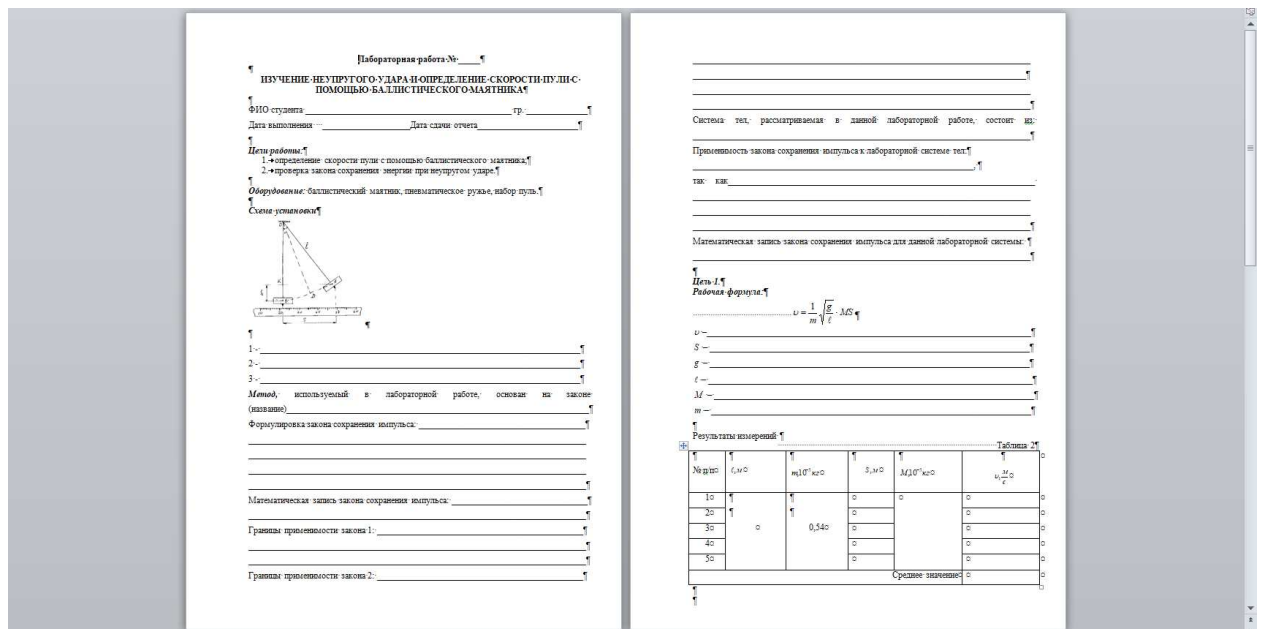


Рисунок 146 – Скриншот шаблона отчета лабораторной работы 3



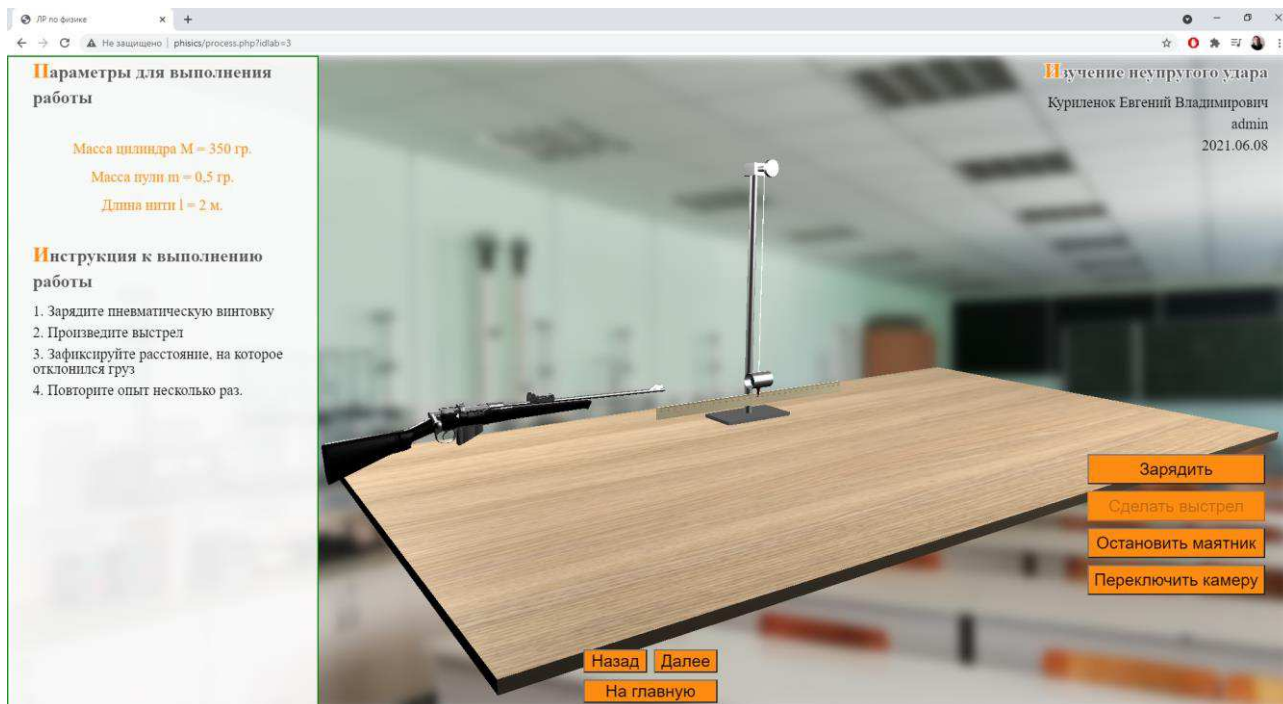


Рисунок 147 – Скриншот страницы демонстрации физического процесса лабораторной работы 3

На этой странице студенту необходимо провести опыт, описанный в порядке выполнения работы, то есть, он должен зарядить винтовку, произвести выстрел, зафиксировать все необходимые результаты, после чего перейти на завершающую страницу лабораторной работы (рисунок 148).

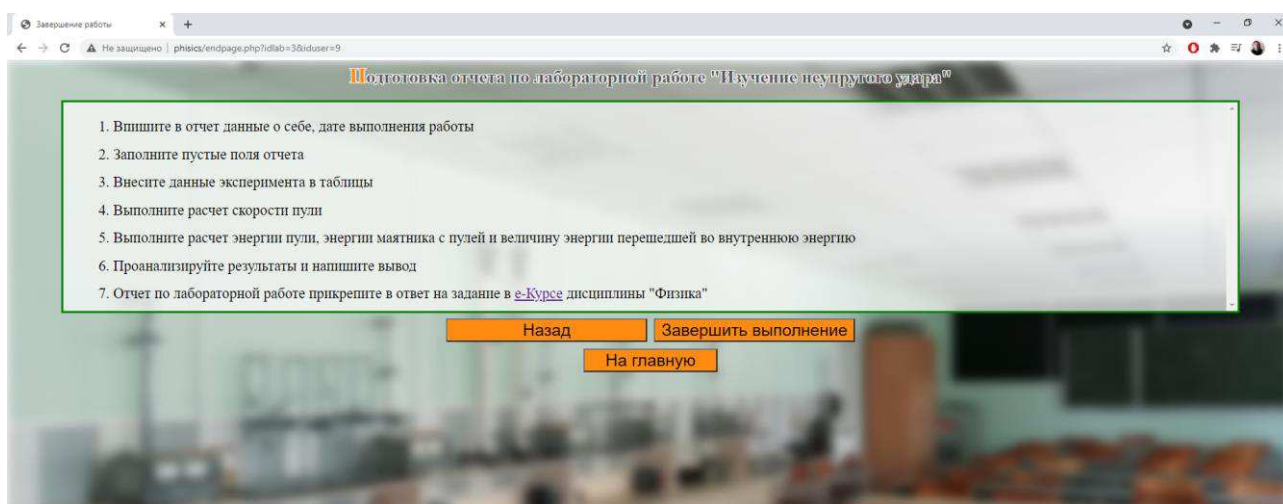


Рисунок 148 – Скриншот страницы завершения лабораторной работы 3

На этой странице студент читает информацию с рекомендациями по заполнению отчета о выполнении лабораторной работы, после чего он должен нажать кнопку «Завершить выполнение», чтобы данные о выполнении лабораторной работы зафиксировались в системе (ФИО студента, дата выполнения, тема лабораторной работы). На рисунке 149 показан скриншот сообщения, об успешном выполнении работы.

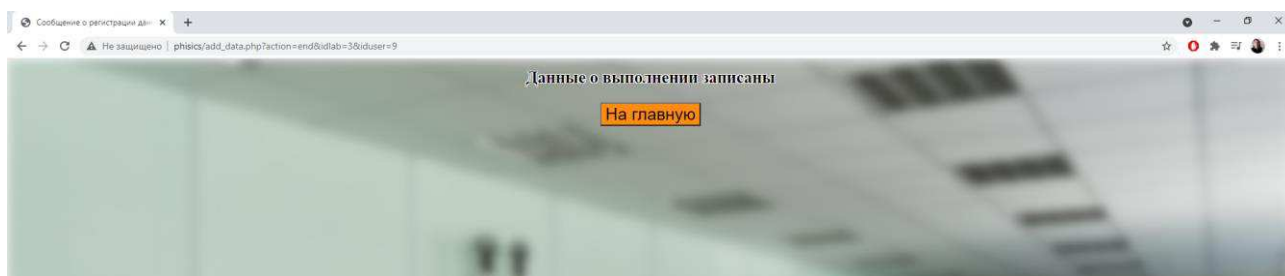


Рисунок 149 – Скриншот сообщения о записи данных выполнения

Для окончательного выполнения, заполненный отчет студент должен прикрепить в eКурсе дисциплины «физика» и отправить на проверку преподавателю. Скриншоты этого процесса показаны на рисунках 150-151.

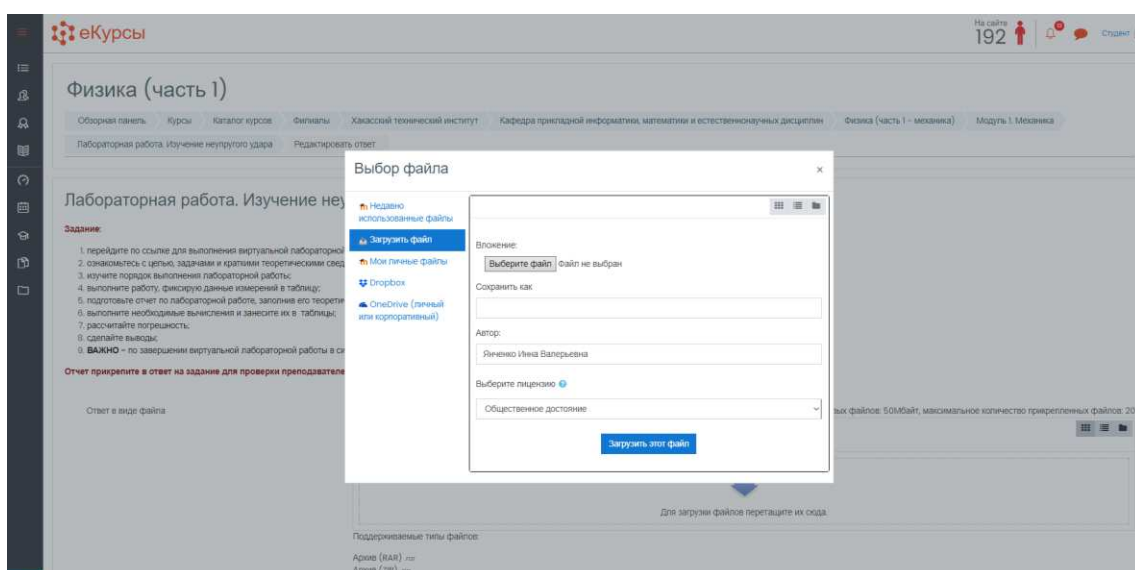


Рисунок 150 – Скриншот страницы eКурса по физике с возможностью

## представления отчета по лабораторной работе

Редактировать ответ    Удалить ответ

Вы можете внести изменения в представленную работу

Отправить на проверку

Однажды представив ответ на это задание Вы больше не сможете изменить его.

---

### Лабораторная работа. Изучение неупругого удара

**Задание:**

1. перейдите по ссылке для выполнения виртуальной лабораторной работы
2. ознакомьтесь с целью, задачами и краткими теоретическими сведениями;
3. изучите порядок выполнения лабораторной работы;
4. выполните работу, фиксируя данные измерений в таблицу;
5. подготовьте отчет по лабораторной работе, заполнив его теоретическими данными и данными измерений в таблицах;
6. выполните необходимые вычисления и занесите их в таблицы;
7. рассчитайте погрешность;
8. сделайте выводы;
9. **ВАЖНО** – по завершении виртуальной лабораторной работы в системе кликните кнопку "Завершить работу" для фиксации ее выполнения вам в базе данных системы.

**Отчет прикрепите в ответ на задание для проверки преподавателем.**

#### Подтвердить отправку ответа

Это мой собственный ответ на задание, кроме случаев, когда подтверждено использование ответов других людей.

Вы уверены, что хотите представить свою работу для оценивания? Вы больше не сможете изменить свой ответ.

[Продолжить](#)    [Отмена](#)

---

6. выполните необходимые вычисления и занесите их в таблицы;
7. рассчитайте погрешность;
8. сделайте выводы;
9. **ВАЖНО** – по завершении виртуальной лабораторной работы в системе кликните кнопку "Завершить работу" для фиксации ее выполнения вам в базе данных системы.

**Отчет прикрепите в ответ на задание для проверки преподавателем.**

#### Состояние ответа

Номер попытки	Попытка 1.
Состояние ответа на задание	Ответы для оценки
Состояние оценивания	Не оценено
Последнее изменение	Воскресенье, 6 Июнь 2021, 12:33
Ответ в виде файла	<a href="#">лр1.doc</a> 6 Июнь 2021, 12:31
Комментарии к ответу	<a href="#">Комментарии (0)</a>

Рисунок 151 – Скриншоты страницы eКурса по физике с процессом отправки отчета по лабораторной работе

Таким образом, студенту, выполняя работу, необязательно находиться в

аудитории и в институте вообще, достаточно провести опыты в любом удобном месте, где есть компьютер средней мощности. Преподаватель может контролировать выполнение лабораторных работ, заходя в систему «Виртуальные лабораторные работы по физике» под своей учетной записью, а также, благодаря отчетам, которые пишут и прикрепляют в eКурсе студенты после выполнения работы.

### **Выводы по разделу «Описание разработки информационной системы «Виртуальные лабораторные работы по физике»**

В проектном разделе выпускной квалификационной работы был полностью описан процесс создания информационной системы, основанной на клиент-серверной архитектуре, в виде веб-приложения «Виртуальные лабораторные работы по физике».

В частности, представлено последовательно описание:

1. создания интерфейса всех лабораторных работ системы;
2. процесса создания базы данных веб-приложения;
3. процесса создания разметки и дизайна всех веб-страниц приложения;
4. процесса создания имитационных моделей лабораторных работ на темы:
  - «Изучение внутреннего трения жидкости»;
  - «Изучение дифракции света»;
  - «Изучение неупругого удара».

Описание разработки виртуальных лабораторных работ сопровождается скриншотами кода приложения и результатами его исполнения.

Приведено подробное описание процесса выполнения работы студентом на примере лабораторной работы «Изучение неупругого удара» со всеми необходимыми скриншотами страниц веб-приложения и eКурса «Физика» от

входа в систему электронного обучения, перехода в систему виртуальных лабораторных работ до прикрепления отчета в eКурсе дисциплины «Физика».

Таким образом, цель работы, состоящая в разработке системы виртуальных лабораторных работ, достигнута.

### **3 Оценка экономической эффективности проекта**

#### **3.1 Расчет затрат реализации проекта веб-приложения «Виртуальная лабораторная работа по физике»**

Расчет затрат на проектирование ИС производится по формуле:

$$K_{\text{пр}} = K_{\text{зп}} + K_{\text{ипс}} + K_{\text{свт}} + K_{\text{проч}}, \quad (1)$$

где  $K_{\text{зп}}$  – заработная плата проектировщиков;

$K_{\text{ипс}}$  – затраты на инструментальные программные средства;

$K_{\text{свт}}$  – затраты на средства вычислительной техники;

$K_{\text{проч}}$  – прочие затраты.

Разработчики: программист; дизайнер.

Программное обеспечение для разработки:

- ПО для создания макетов, объектов физических процессов;
- ПО для написания программного кода приложения.

Вычислительная техника разработки ИС: ПК для разработки.

##### **3.1.1 Затраты на заработную плату разработчиков**

Заработная плата (ЗП) проектировщиков:

1. Программист:
  - ЗП программиста – 20 000 руб.
  - Срок работы разработчика – 4 месяца (88 рабочих дней).

- Рабочий день – неполный (3 часа).
  - Срок работы (полных рабочих дней) – 26 дней.
  - Система оплаты труда – тарифная.
2. Дизайнер:
- ЗП дизайнера – 15 000 руб.
  - Срок работы дизайнера – 24 дня.
  - Рабочий день – неполный (3 часа).
  - Срок работы (полных рабочих дней) – 6 дней.

Расчет ЗП проектировщиков представлен в таблицах 3 и 4.

Таблица 3 – Зарплата разработчика за 26 дней

	Начислено, руб	Удержано, руб
По окладу	20 800	
Премия	100	
СК	6 240	
РК	6 240	
НДФЛ		5 421
Итого	41 700	
На руки	36 279	

Таблица 4 – Зарплата дизайнера за 6 дней

	Начислено, руб	Удержано, руб
По окладу	3 600	
СК	1 080	
РК	1 080	
НДФЛ		748,8
Итого	5 760	
На руки	5 011,2	

Из таблиц видно, что разработчик и дизайнер на руки получают 36 279 и 5 011,2 рублей соответственно.

При учете стоимости проектирования программного продукта в нее будет входить заработная плата без учета налогов. То есть в фонд оплаты труда будет включена зарплата разработчика и дизайнера до налоговых вычетов (41 700 и 5 760 рублей соответственно).

Кроме того, в фонд оплаты труда пойдут отчисления во внебюджетные фонды такие, как в пенсионный фонд, фонд медицинского страхования и т.д. которые составят 30,2%.

Фонд оплаты труда – затраты на ЗП + 30,2% (отчисления в фонды).

Итого затраты на ЗП ( $K_{зп}$ ) –  $(41\ 700+5\ 760)+30,2\%= 61\ 792,92$  руб.

### 3.1.2 Затраты на средства вычислительной техники

Средствами вычислительной техники для разработки данного проекта будет персональный компьютер для разработки веб-приложения. Так как веб-приложением будет являться виртуальная лаборатория по физике, то и компьютер нужен соответствующий. И разработчику, и дизайнеру нужен компьютер одинаковой мощности, что делает возможным использование в разработке только одного ПК. Стоимость оборудования показана в таблице 5 [14].

Таблица 5 – Стоимость оборудования для разработки

№	Наименование оборудования	Количество, шт	Стоимость, руб	Срок службы
1	Процессор Intel Core i3-10100F	1	8 500	5 лет
2	Видеокарта Asus GeForce GT 1030 PHOENIX 2 ГБ	1	8 700	5 лет
3	Материнская плата ASRock H470M-HDV	1	5 700	5 лет
4	Оперативная память AMD Radeon R7 Performance Series [R748G2400U2S-U] 8 ГБ	1	3 300	5 лет
5	Жесткий диск Toshiba P300 1ТБ	1	3 200	5 лет
6	Блок питания Winard 500W	1	1 400	5 лет
7	Корпус DEXP DC-201M	1	1 700	10 лет
8	Монитор DEXP FF201H 19.5"	1	7 000	5 лет

9	Клавиатура Defender Next HB-440	1	300	5 лет
10	Мышь проводная Oklick 115S	1	300	5 лет

Так как использоваться ПК будет не весь срок службы, то необходимо рассчитать амортизацию оборудования. Расчет амортизации одного компьютера за 3 месяца использования:

$$A_r = 40\,100 * 1/5 = 8\,020.$$

$A_{пр} = (8\,020 / 12) * 4 = 2\,670$  р. отчисления на амортизацию за 1 компьютер.

### 3.1.3 Затраты на инструментальные программные средства

Затратами на инструментальные программные средства в данном проекте будут операционная система одного персонального компьютера, а также программное обеспечение для создания веб-приложения (таблица 6).

Таблица 6 – Стоимость программного обеспечения

№	Наименование ПО	Количество, шт	Стоимость, руб	Срок службы
1	Windows 10	1	9 900	3 года
2	Sublime Text	1	бесплатно	3 года
3	OsPanel	1	бесплатно	3 года
4	PHPMyAdmin	1	бесплатно	3 года
5	GoogleChrome	1	бесплатно	3 года

Стоимость затрат на ПО будет включать в себя только амортизацию операционной системы на срок разработки.

$$A_r = 9\,900/3 = 3\,300.$$



$$A_{\text{пр}} = 3\,300 / 4 = 825.$$

Затраты на ПО ( $K_{\text{ипс}}$ ) = 825 руб.

Прочие затраты, такие как оплата коммунальных услуг, аренда помещения и т.д. это примерно 3% от всех затрат на проектирование и составят 1 940 руб.

Затраты на проектирование ( $K_{\text{пр}}$ ) = (61 792,92 + 2 670 + 825) + 3% = 67 246,56 р.

Капитальные затраты рассчитываются по формуле:

$$K = K_{\text{пр}} + K_{\text{тс}} + K_{\text{лс}} + K_{\text{по}} + K_{\text{ио}} + K_{\text{об}} + K_{\text{оэ}},$$

(2)

где  $K_{\text{пр}}$  – затраты на проектирование ИС;

$K_{\text{тс}}$  – затраты на технические средства управления;

$K_{\text{лс}}$  – затраты на создание линий связи локальных сетей;

$K_{\text{по}}$  – затраты на программные средства;

$K_{\text{ио}}$  – затраты на формирование информационной базы;

$K_{\text{об}}$  – затраты на обучение персонала;

$K_{\text{оэ}}$  – затраты на опытную эксплуатацию.

$K_{\text{пр}}$  посчитаны ранее и составляет 67 246,56 руб.

$K_{\text{тс}}$  равны 0 руб., так как приложение будет работать на устройстве любого пользователя.

$K_{\text{лс}}$  равны 0 руб. – так как при проектировании веб-приложения используется локальный сервер.

$K_{\text{по}}$  равны 0 руб. Для работы данного приложения нужен только браузер.

$K_{ин}$  равны 0 руб. Информационная база (БД) веб-приложения создана на этапе проектирования.

$K_{об}$  – затраты на обучение персонала.

Чтобы веб-приложение «Виртуальные лабораторные работы по физике» функционировало так, как задумано изначально, необходимо обучить преподавателя, который будет использовать это веб-приложения для обучения студентов. Обучать будет программист 3 дня, по 2 часа в один день. Поэтому  $K_{об}$  это зарплата программиста за один полный рабочий день или  $20\ 000 / 25 + 30,2\%$  (отчисления во внебюджетные фонды) = 1041,6 руб.

$K_{оп}$  – затраты на опытную эксплуатацию – зарплата программиста за время опытной эксплуатации 2 недели или 14 полных рабочих дней. Тестировать приложение программист будет неполный рабочий день – 2 часа, поэтому зарплата рассчитывается за  $2 * 14 = 28$  часов или 4 полных рабочих дня. ЗП программиста составляет 20 000 руб в месяц (без надбавок). ЗП программиста на период опытной эксплуатации равна  $20\ 000 / 25 * 4 + 30,2\% = 4\ 166,4$  руб.

Капитальные затраты (К) приведены в таблице 7.

Таблица 7 – Капитальные затраты проекта

Наименование затрат	Затраты, руб.
Затраты на проектирование ИС	67 246,56
Затраты на технические средства управления	0
Затраты на создание линий связи локальных сетей	0
Затраты на программные средства	0
Затраты на формирование информационной базы	0
Затраты на обучение персонала	1 041,6
Затраты на опытную эксплуатацию	4 166,4
Итого	72 454,56

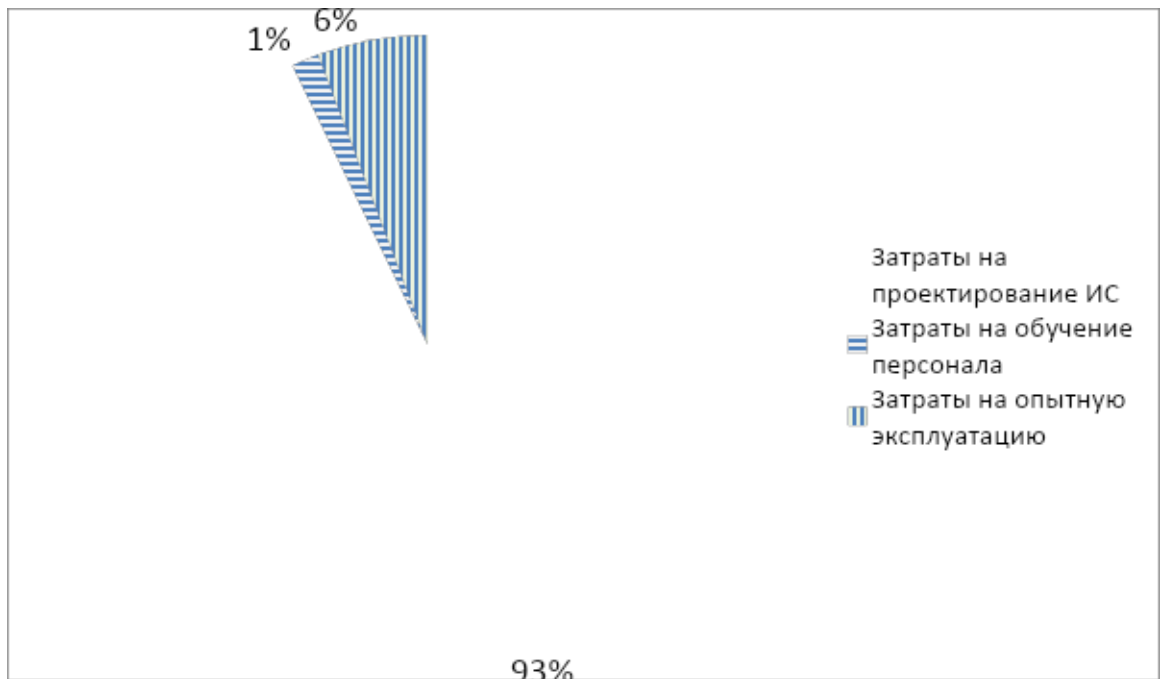


Рисунок 152 – Диаграмма соотношения капитальных затрат

### 3.1.4 Эксплуатационные затраты

Расчет эксплуатационных затрат производится по формуле

$$C = C_{зп} + C_{ао} + C_{то} + C_{лс} + C_{ни} + C_{проч}, \quad (3)$$

где  $C_{зп}$  – ЗП персонала, работающего с использованием ИС;

$C_{ао}$  – амортизационные отчисления;

$C_{то}$  – затраты на техническое оборудование;

$C_{лс}$  – использование вычислительных сетей;

$C_{ни}$  затраты на носители информации.

Для поддержания веб-приложения в рабочем состоянии необходимы затраты на заработную плату администратора, который будет периодически обновлять базу данных пользователей (примерно 1 000 р на данное приложение), затраты на сервер для размещения веб-приложения (5 000 руб.), затраты на интернет (1000 руб.). Амортизационные

отчисления для веб-приложения будут минимальными и учитываются в прочих затратах, Итого эксплуатационные затраты составят 7 000 руб. за 1 год (Таблица 8).

Таблица 8 – Эксплуатационные затраты проекта

Наименование затрат	Затраты, руб.
ЗП персонала, работающего с использованием ИС	1000
Амортизационные отчисления	0
Затраты на техническое оборудование	5000
Использование вычислительных сетей	1000
Итого	7000

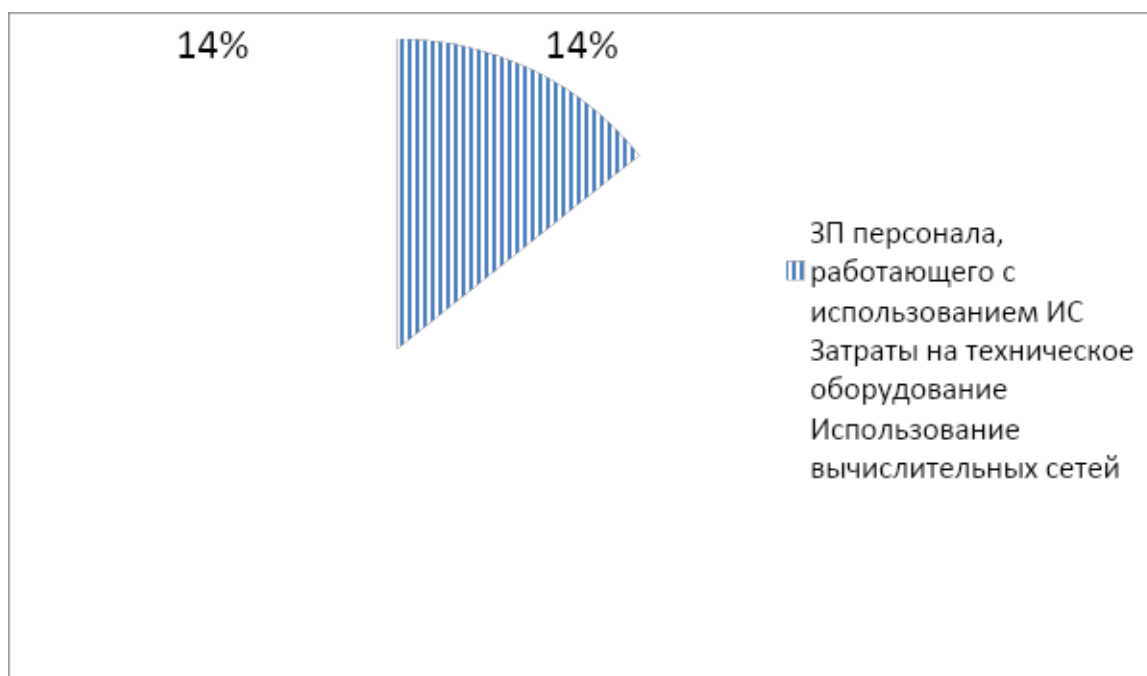


Рисунок 153 – Диаграмма соотношения эксплуатационных затрат

### 3.2 Расчет затрат реализации проекта методом ТСО (TotalCostofOwership)

Согласно методике ТСО совокупная стоимость владения рассчитывается по формуле:

$$\text{ТСО} = \text{DE} + \text{IC1} + \text{IC2},$$

(4)

где DE – прямые расходы;

IC1, IC2 – косвенные расходы группы 1 и 2.

Косвенные затраты не учитываются, поскольку при оценке риска мероприятия по их преодолению не требуют дополнительных финансовых вложений (см. пункт 3.4).

Прямые расходы рассчитываются по формуле:

$$\text{DE} = \text{DE1} + \text{DE2} + \text{DE3} + \text{DE4} + \text{DE5} + \text{DE6} + \text{DE7} + \text{DE8},$$

(5)

где DE1 – капитальные затраты;

DE2 – расходы на управление ИТ;

DE3 – расходы на техническую поддержку АО и ПО;

DE4 – расходы на разработку прикладного ПО внутренними силами;

DE5 – расходы на аутсорсинг;

DE6 – командировочные расходы;

DE7 – расходы на услуги связи;

DE8 – другие группы расходов.

Капитальные затраты посчитаны ранее и составляют 72 454,56 руб.

Расходы на управление ИТ –  $\text{DE2} = C_{\text{зп}} = 1\,000$  руб.

Для данной системы  $\text{DE3} = C_{\text{ао}} + C_{\text{то}} = 0 + 5\,000 = 5\,000$  руб.

$\text{DE4} = 0$  руб. Веб-приложение «Виртуальные лабораторные работы по физике» будет работать в браузере компьютера, который является бесплатным для пользователя.

DE5 = 0 руб. Для данной системы расходы на аутсорсинг будут минимальными.

DE6 = 0 руб. Командировочных затрат нет.

Для данной системы DE7 = C<sub>лс</sub> = 1000 руб.

DE8 составляют 3% от всех прямых расходов и равны 2 328 руб.

Таблица 9 – Список всех расходов

Наименование затрат	Затраты, руб.
Капитальные затраты	72 454,56
Расходы на управление ИТ	1 000
Расходы на техническую поддержку АО и ПО	5 000
Расходы на разработку прикладного ПО внутренними силами	0
Расходы на аутсорсинг	0
Командировочные расходы	0
Расходы на услуги связи	1 000
Другие группы расходов	2 384
Итого	81 838,2

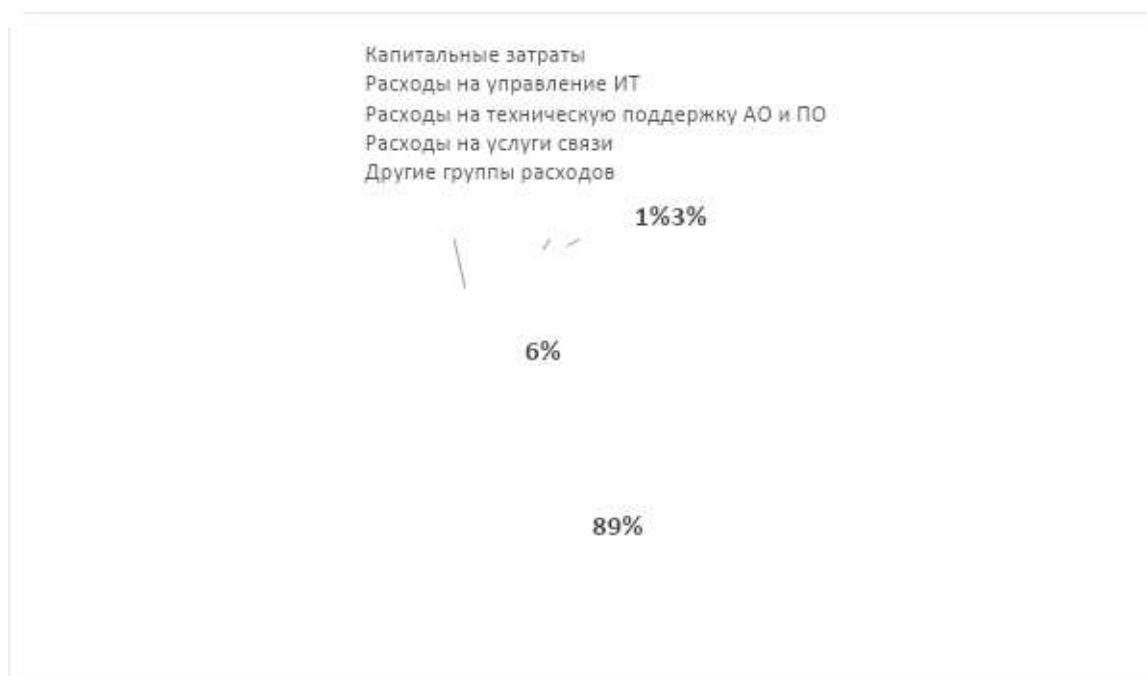


Рисунок 154 – Диаграмма соотношения всех расходов

### 3.3 Определение экономической эффективности реализации проекта

### 3.3.1 Анализ рынка продуктов-аналогов. Установление стоимости программного продукта

Для установления цены программного продукта, проведём сравнительный анализ конкурентов в таблице 10.

Таблица 10 — Сравнительная таблица продуктов-аналогов

Характеристики	Приложение «Виртуальная лаборатория по физике»	Виртуальный лабораторный стенд «Механика»	Лабораторные работы по физике ProgramLab	Веб-приложение «Виртуальные лабораторные работы по физике»
Функционал	Содержит набор виртуальных лабораторных работ по физике для подготовки к выполнению реальных лабораторных работ или их замене при отсутствии соответствующего оборудования	Программное обеспечение для ПК, которое позволяет изучить конструкцию и принцип работы различных приборов и технологических установок, взаимодействуя с их реалистичными интерактивными 3D-моделями.	Сервис предоставляет возможность приобрести учебным заведениям виртуальные лабораторные по физике, которые выполнены с использованием 3D графики.	Предоставляет возможность выполнения 3х лабораторных работ в удаленном формате, с занесением сведений о выполнении этих работ в таблицу для преподавателя.
Оценка функционала	2/5	5/5	3/5	4/5
Дополнительное ПО	Не требуется	Не требуется	Не требуется	Браузер
Характеристики устройства	100 Мб ПЗУ, 2 гб ОЗУ.	Процессор: 2-х ядерный, 1,4 ГГц ОЗУ: 4 Гб, DDR3 ПЗУ 1 Гб DVD привод	Процессор: 2-х ядерный, 2,4 ГГц 1 гб ПЗУ, 4 гб ОЗУ	Процессор: 2-х ядерный, 2,4 ГГц 1 гб ПЗУ, 4 гб ОЗУ
Стоимость	35 000 руб	188 240 руб	Стоимость 10 лицензий на одну работу 15 640 руб. Стоимость 3х работ 46 920 руб	Стоимость на основе сравнения с аналогами – 75 000 руб.

Из данной таблицы видно, что приложения-аналоги, которые были выбраны для сравнения, ничем не уступают создаваемому приложению «Виртуальные лабораторные работы по физике». Самым главным недостатком этих аналогов будет цена, поскольку лицензионные версии программ, которые может позволить себе учебное заведение, стоят очень дорого. Кроме того, у данных аналогов не все работы, которые необходимы институту, имеются в наличии, и имеется множество работ, которые не входят в программу обучения.

Установление стоимости программного продукта.

#### 1. Затратный метод.

Стоимость программного продукта рассчитывается на основании расчёта капитальных затрат, определяется себестоимость продукта, прибавляется прибыль, 10% от себестоимости, прибавляется стоимость рекламы и продвижения продукта.

Капитальные затраты на создание веб-приложения «Виртуальные лабораторные работы по физике» составляют:

$K=72\,454,56$  руб.

Прибыль, 10% от себестоимости:

$P=72\,454,56 \cdot 10\% = 7\,245,45$  рублей.

Для данного веб-приложения нет необходимости в рекламном продвижении, поскольку оно создается в учебных целях.

Затраты на рекламу 0 руб.

Итоговая стоимость программного продукта:

$72\,454,56 + 7\,245,45 + 0 = 79\,700,01$  руб.

#### 2. Доходный метод

Для того чтобы определить стоимость данного программного продукта доходным методом, нужно посчитать доходы, которые будет приносить данное веб-приложение. Приложение «Виртуальные лабораторные работы по физике» создается в учебных целях для конкретного института, поэтому оно будет бесплатным и никаких доходов приносить не будет.



### 3.3.2 Экономическая эффективность реализации проекта

Проект «Виртуальные лабораторные работы по физике» создается для учебного заведения, из чего следует, что программный продукт не будет приносить прибыль или сокращать расходы на персонал. Экономическая эффективность от внедрения веб-приложения заключается в том, что институту не нужно закупать дорогостоящие лабораторные установки, кроме этого, обучение в формате современных компьютерных технологий делает возможным привлечение новых студентов, что, несомненно, является положительной стороной использования данного веб-приложения.

Стоимость разработанного веб-приложения составляет 79 700 рублей. Стоимость лабораторного оборудования для трех лабораторных работ показана в таблице 11 [15].

Таблица 11 – Стоимость лабораторных установок

Лабораторное оборудование	Стоимость, руб.
Лабораторное оборудование «Определение коэффициента внутреннего трения жидкости по методу Стокса»	125 000
Установка «Изучение дифракции света»	85 000
Учебный модульный комплекс «Механика-1».	90 000
Итого	300 000

Из данной таблицы видно, что стоимость оборудования даже для одной лабораторной работы превышает стоимость веб-приложения, поэтому, можно сделать вывод, что использование веб-приложения экономически эффективно для института.

Также, проект предоставляет возможность студентам выполнять лабораторные работы дистанционно, что актуально в настоящее время для студентов очной формы обучения и всегда будет актуально для студентов

заочной формы обучения, которые могут получить наглядные знания по такой дисциплине, как физика без использования специального оборудования.

### **3.4 Оценка рисков реализации IT-проекта**

Создание любого продукта, в частности интеллектуального, сопровождается как прибылью, так и убытками, что связано с рисками реализации проекта. При создании веб-приложения существует достаточно большое количество рисков. К ним можно отнести: несвоевременная сдача проекта; увеличение расходов на разработку; несоответствие требованиям заказчика.

Чтобы повысить эффективность приложения, нужно учитывать, что такие риски могут появиться, а также знать, как уменьшить вероятность появления этих рисков и, если такие риски появились, как их преодолевать.

Виды рисков IT-проекта: 1. риск соответствия IT-проекта; 2. реализационный риск; 3. технологический риск; 4. риск денежных потоков.

При создании веб-приложения «Виртуальные лабораторные работы по физике» самая менее вероятная группа рисков – это риск денежных потоков, так как после реализации проекта он не будет приносить доход предприятию. Наиболее вероятные группы рисков – риски соответствия IT-проекта и реализационные риски.

Одним из рисков соответствия будет несоответствие требованиям заказчика. Созданное веб-приложение может получиться не таким, каким его видел заказчик, а это дополнительное время на доработку.

Реализационным риском будет несвоевременная сдача проекта, так как в процессе разработки могут возникнуть непредвиденные трудности и время реализации проекта может стать больше, чем ожидалось на этапе планирования. Кроме того, могут быть увеличены расходы на разработку в связи с изменением цен на оборудование для разработки или в связи с иными непредвиденными расходами.

### 3.4.1 Оценка рисков проекта

Оценка рисков проекта производится по 3х уровневой шкале влияния на реализацию проекта: низкий уровень, средний уровень, высокий уровень.

Для веб-приложения «Виртуальные лабораторные работы по физике» определены следующие риски реализации проекта и их оценка:

– *Несоответствие требованиям заказчика*, «Низкий уровень». Этот риск имеет достаточно низкую вероятность, так как все требования и условия заказчика подробно описаны в техническом задании. Но все предвидеть невозможно, поэтому такой риск, пусть и не высокий, есть.

– *Несвоевременная сдача проекта*, «Средний уровень». Этот риск имеет достаточно высокую вероятность, так как даже при условии того, что все сроки прописаны в договоре, есть вероятность непредвиденных ситуаций, из-за которых разработка проекта может затянуться. Такими ситуациями могут быть проблемы с поставкой оборудования или программного обеспечения для разработки, технические проблемы (отключение электричества, интернета), личные обстоятельства разработчика (например, больничный).

– *Увеличенные расходы на разработку*, «Средний уровень». Риск зависит в основном от экономической ситуации в мире, то есть от цен на оборудование, которые постоянно растут. Примером такого риска может быть стоимость видеокарт, которые за последние несколько месяцев увеличились более чем на 40%. Поэтому этот риск имеет достаточно высокую вероятность.

### 3.4.2 Мероприятия по снижению вероятности рисков проекта

Для того чтобы повысить эффективность данного проекта, необходимо уменьшить вероятность возникновения тех или иных рисков. И чтобы предотвратить или снизить вероятность рисков нужно запланировать следующие мероприятия:

1. Чтобы снизить риск *несоответствия требованиям заказчика*, необходимо более тщательно обговорить условия разработки, более подробно описать их в техническом задании. Кроме того, во время разработки необходимо постоянно консультироваться с заказчиком.

2. Чтобы снизить *риск несвоевременной сдачи проекта* необходимо дополнительно обговорить с заказчиком о непредвиденных ситуациях, которые могут повлиять на время разработки, а также, выделить дополнительное время, на случай если такие ситуации возникнут. Установить срок сдачи базовых элементов проекта, разработчик в любом случае сможет реализовать. Возможно, нанять дополнительного разработчика, если сроки невозможно сдвинуть.

3. Чтобы снизить *риск увеличенных расходов на разработку приложения* необходимо дополнительно обговорить стоимость затрат на оборудование и программное обеспечение и прописать её в договоре.

Таблица 12 – Риски проекта

№ п/п	Группы рисков	Перечень рисков проекта	Уровень влияния риска на проект	Вероятность риска	Возможность снижения риска
1	Реализационные риски	Несвоевременная сдача проекта	Средний	Средняя	Проработка технического задания, установка сроков сдачи базовых элементов проекта, наем второго программиста.
2		Увеличенные расходы на разработку	Средний	Высокая	Проработка технического задания

3	Риск соответствия	Несоответствие требованиям заказчика	Низкий	Низкая	Подробное техническое задание, постоянная консультация с заказчиком.
---	-------------------	--------------------------------------	--------	--------	--

### **Выводы по разделу «Оценка экономической эффективности проекта»**

В данном разделе были выполнены расчеты затрат на реализацию информационной системы «Виртуальные лабораторные работы по физике» с целью определить стоимость разработки. Для этого было выполнено:

1. расчет затрат на проектирование веб-приложения;
2. расчет эксплуатационных затрат;
3. расчет капитальных затрат;
4. расчет затрат на реализацию проекта методом ТСО;

Кроме этого, была определена экономическая эффективность реализации проекта путем сравнительного анализа продуктов аналогов, а также по расчетам затрат на разработку веб-приложения.

Проведена оценка рисков, связанных с реализацией проекта и определены мероприятия по снижению возникновения этих рисков в проекте.

## ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы было спроектировано, подготовлено для реализации и полностью реализовано веб-приложение «Виртуальные лабораторные работы по физике». В ходе написания этой работы была достигнута цель и выполнены следующие задачи для ее достижения:

1. охарактеризована основная деятельность объекта исследования ХТИ – филиала СФУ;
2. выполнен анализ существующего процесса выполнения лабораторных работ в вузе;
3. проанализированы аналогичные программные средства для выполнения виртуальных лабораторных работ;
4. осуществлен выбор архитектуры ИС и инструментальных средств разработки ИС «Виртуальные лабораторные работы по физике»;
5. разработана ИС «Виртуальные лабораторные работы по физике»;
6. оценена экономическая эффективность разработки ИС «Виртуальные лабораторные работы по физике»;
7. оценены риски и способы их минимизации.

В первом разделе выпускной работы проанализировано учебное заведение ХТИ – филиал СФУ, его деятельность и структура. Был проведен анализ предметной области, актуальности разработки, были изучены подобные разрабатываемой программы.

Во втором разделе было разработано веб-приложение.

В третьем разделе работы была посчитана экономическая эффективность разработки, проведен анализ рисков реализации проекта, выполнен поиск путей решений по снижению вероятности возникновения этих рисков.

Все задачи выпускной квалификационной работы выполнены, цель полностью достигнута.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СФУ [Электронный ресурс] – Режим доступа: <http://www.sfu-kras.ru/>.
2. ХТИ – филиал СФУ [Электронный ресурс] – Режим доступа: <http://khti.sfukras.ru/>.
3. Трухин, А.В., «Об использовании виртуальных лабораторий в образовании» / А. В. Трухин // Открытое и дистанционное образование. – 2002. – № 4 – С. 81-82.
4. Информационный портал «VC.RU». Статья «8 лучших онлайн-ресурсов для лабораторных работ» [Электронный ресурс]. – Режим доступа: <https://vc.ru/education/139879-8-luchshih-onlayn-resursov-dlya-laboratornyh-rabot>
5. Сервис виртуальных лабораторных работ «Профессиональная группа» [Электронный ресурс]. – Режим доступа: <https://www.professionalgroup.ru/>
6. Сервис виртуальных лабораторных работ «ProgramLab» [Электронный ресурс]. – Режим доступа: <https://pl-llc.ru/>
7. Архитектура ИС [Электронный ресурс]. – Режим доступа: [http://it-claim.ru/Education/Course/ISDevelopment/Lecture\\_3.pdf](http://it-claim.ru/Education/Course/ISDevelopment/Lecture_3.pdf)
8. Свободная энциклопедия «Википедия» [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/>
9. Информационный портал «Ресурсы для разработчиков». Статья «WebGL разработка» [Электронный ресурс]. – Режим доступа: [https://developer.mozilla.org/ru/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/ru/docs/Web/API/WebGL_API)
10. Вильданов, А.Н., 3D-моделирование на WebGL с помощью библиотеки Three.js: учебное пособие / А.Н. Вильданов – Уфа: РИЦ БашГУ, 2014. – 113 с.
11. СДО eКурсы [Электронный ресурс]. – Режим доступа: <https://e.sfu-kras.ru/>

12. Физика. Лабораторные работы [Электронный ресурс] : методические указания / И. В. Янченко, В. В. Тимченко, В. В. Стреж ; Сиб. федер. ун-т; ХТИ - филиал СФУ. - Электрон. текстовые дан. Электрон. граф. дан. ( файла : 0,64 Мбайтов). - Абакан : ХТИ - филиал СФУ, 2018.

13. Выполнение и защита выпускной квалификационной работы по направлению 09.03.03 "Прикладная информатика" [Electronic resource] : метод. указания / Е. Н. Скуратенко, В. И. Кокова, И. В. Янченко ; Сиб. федер. ун-т; ХТИ - филиал СФУ. - Электрон. текстовые дан. Электрон. граф. дан. - Абакан : ХТИ - филиал СФУ, 2017.

14. Интернет-магазин DNS [Электронный ресурс]. – Режим доступа: <https://www.dns-shop.ru/>

15. Портал «Учебная техника». Страница «Цены на лабораторные стенды и учебное оборудование по физике» [Электронный ресурс]. – Режим доступа: <https://www.xn--80aabpcsouc2cg4at0j.xn--p1ai/wps/glavnaya/uchebnaya-texnika-dlya-professionalnogo-obrazovaniya/fizika/>



Выпускная квалификационная работа выполнена мной самостоятельно.  
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 15 наименований.

Один экземпляр сдан на кафедру.

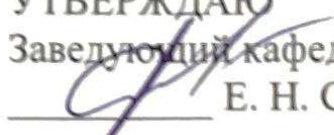
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

\_\_\_\_\_ Куриленок Евгений Владимирович

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт –  
филиал ФГАОУ ВО «Сибирский федеральный университет»

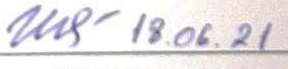
Кафедра прикладной информатики, математики  
и естественно-научных дисциплин

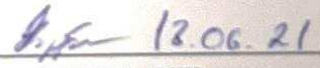
УТВЕРЖДАЮ  
Заведующий кафедрой  
  
Е. Н. Скуратенко  
« 18 » 06 20 21 г.

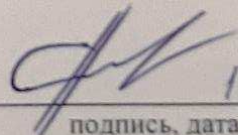
**БАКАЛАВРСКАЯ РАБОТА**

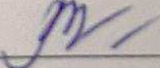
09.03.03 Прикладная информатика

Разработка виртуальной лабораторной работы по физике

Руководитель  13.06.21 доцент, канд. пед. наук И. В. Янченко  
подпись, дата

Выпускник  13.06.21 Е. В. Куриленок  
подпись, дата

Консультант  
по экономическому разделу  13.06.21 Е. Н. Скуратенко  
подпись, дата

Нормоконтролер  13.06.21 В. И. Кокова  
подпись, дата