

Министерство науки и высшего образования РФ

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

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

"__" _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Платформа разработчика средств внешнего оценивания
для е-курсов СФУ

Руководитель	_____	_____	ст. преподаватель	О.В. Шмелёв
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		А.Э. Чичканов
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____	ст. преподаватель	О.В. Шмелёв
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	

Красноярск 2023

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего профессионального образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

« ____ » _____ 2023 г.

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

Студенту Чичканову Александру Эдуардовичу
фамилия, имя, отчество

Группа ЗКИ18-07Б Направление (специальность) 090301
номер код

Информатика и вычислительная техника
наименование

Тема выпускной квалификационной работы: Платформа разработчика средств внешнего оценивания для e-курсов СФУ

Утверждена приказом по университету № 7667/с от 17.05.2023

Руководитель ВКР: О.В. Шмелёв, ст. преподаватель кафедры ВТ ИКИТ
инициалы, фамилия, учёная степень, должность, место работы

СФУ

Исходные данные для ВКР:

- 1) электронные курсы СФУ на базе системы Moodle;
- 2) разрабатываемая web-служба (серверная часть);
- 3) организация хранения данных пользователей;
- 4) сторонние образовательные сервисы для которых будет выполняться проверка и оценивание работ.

Перечень разделов ВКР:

- 1) введение;
- 2) описание предметной области;
- 3) проектирование и разработка функциональной архитектуры;
- 4) реализация функциональной архитектуры;
- 5) тестирование разработанной системы;
- 6) внедрение и оценка качества надежности функционирования ПО.

Перечень графического материала:

- 1) презентация в формате PowerPoint.

Руководитель ВКР

_____ подпись

О.В. Шмелёв

инициалы, фамилия

Задание принял к исполнению

_____ подпись

А.Э. Чичканов

инициалы, фамилия

_____ Г.
дата

РЕФЕРАТ

Выпускная квалификационная работа по теме «Платформа разработчика средств внешнего оценивания для е-курсов СФУ» содержит 61 страницу текстового документа, 32 иллюстрации, 10 таблиц, 1 приложение, 13 использованных источников.

LMS, MOODLE, PHP, MYSQL, ПЛАТФОРМА РАЗРАБОТЧИКОВ, ВЕБ-СЕРВЕР, ИНТЕГРАЦИЯ УЧЕБНЫХ ПЛАТФОРМ, LTI.

Объект исследования – платформа электронных образовательных курсов на основе Moodle, с открытым исходным кодом.

Целью данной работы является создание платформы разработчика средств внешнего оценивания для е-курсов СФУ и интеграция с ней по протоколу LTI.

В результате работы была проанализирована предметная область задачи, выбран метод реализации с использованием веб-технологий, изучены способы и методы работы с протоколом LTI посредством языка PHP и использованием специальной библиотеки PHP-LTI.

В итоге была разработана веб-платформа, которая может взаимодействовать с системой Moodle посредством протокола LTI и обмениваться различными данными необходимыми для облегчения и систематизации оценивания выполненных работ студентами.

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ предметной области	6
1.1 Анализ задания и обоснование выбранного решения.....	6
1.2 Цели создания системы	10
1.3 Функциональные возможности	10
1.4 Анализ существующих решений.....	12
2 Проектирование и разработка функциональной архитектуры.....	14
2.1 Выбор технологий и инструментов для разработки.....	14
3 Проектирование и разработка функциональной архитектуры.....	16
3.1 Проектирование функциональности системы	16
3.2 Проектирование архитектуры системы	20
4 Реализация функциональной архитектуры	25
5 Тестирование разработанной системы.....	35
6 Внедрение и оценка качества надежности функционирования ПО	36
Заключение	41
Список использованных источников	42
Приложение А Листинг программы.....	44

ВВЕДЕНИЕ

Современные технологии в образовании, в том числе и электронные курсы, становятся все более популярными. Однако для эффективного обучения необходимо не только предоставить студентам доступ к материалам, но и оценивать их знания. В этом случае средства внешнего оценивания могут стать незаменимыми помощниками. Они позволяют более точно и объективно оценить успеваемость студентов, а также выявить слабые места в процессе обучения. Разработка таких средств является актуальной задачей в современном образовании, и многие учебные заведения уже используют подобные решения для оценки успеваемости своих студентов.

В настоящее время существует множество платформ и средств для образования, среди них особенно выделяются такие как Moodle (система электронного обучения и тестирования) с открытым исходным кодом функционалом которой могут пользоваться любые желающие, а также популярные частные платформы на вроде Coursera, Udemу, Linda и т.д. При помощи данных сервисов обучающиеся могут изучать новый материал, осваивать новые направления и тренировать какие-либо навыки – а авторы курсов или же преподаватели могут отслеживать успех своих учеников при помощи тестирования либо же выполнения каких-либо других заданий. Однако бывает и такое что некоторые задания и учебный материал может находится вне зоны платформы для обучения в силу своей особенности, например, создание виртуальной машины на другом ресурсе и настройка её, или же работа в стороннем приложении таком как Cisco Packet Tracer где необходимо создать макет локальной сети. Такие задания тоже требуют оценки преподавателя, однако этот процесс можем быть трудоёмким поскольку результаты выполненных работ не будут отображаться на ранее озвученных платформах.

Поэтому появилась необходимость создания такого ресурса, который позволял бы преподавателям организовывать проверку заданий на каких-либо сторонних ресурсах при помощи встроенных скриптов, оценивания их и

занесения результатов в образовательную систему, в нашем случае это Moodle.

Цель работы — разработка платформы разработчика средств внешнего оценивания для e-курсов СФУ.

Задачи работы:

Разрабатываемая «Платформа разработчика средств внешнего оценивания для e-курсов СФУ» является инструментальным средством, позволяющим автору курса в системе Moodle e-курсов СФУ (далее Разработчику) создавать с ее помощью программные средства оценивания успеваемости студентов, выполнения ими практических заданий и т.п. (Чекеры), в т.ч. и работающие в автоматизированном режиме.

После стадии разработки Чекера Платформа должна обеспечивать его работу в качестве внешнего средства оценивания при проведении обучения по курсу в соответствии с настоящим ТЗ для студентов и преподавателей этого курса.

Платформа должна быть реализована в виде Web-службы на языке php с использованием СУБД mysql.

1 Анализ предметной области

1.1 Анализ задания и обоснование выбранного решения

Первым шагом является определение требований к разрабатываемому средству. Это может включать в себя такие параметры, как тип заданий, которые должны быть оценены, формат вывода результатов, требования к безопасности и доступности, а также возможность интеграции с другими платформами. При выборе платформы необходимо учитывать требования к функциональности, бюджет, доступность ресурсов и технической экспертизе. Важно выбрать платформу, которая наилучшим образом соответствует потребностям конкретного проекта и позволяет эффективно решать поставленные задачи.

В Сибирском федеральном университете используется система электронного образования и тестирования Moodle на основе открытого кода находящаяся по адресу e.sfu-kras.ru (Рисунок 1).

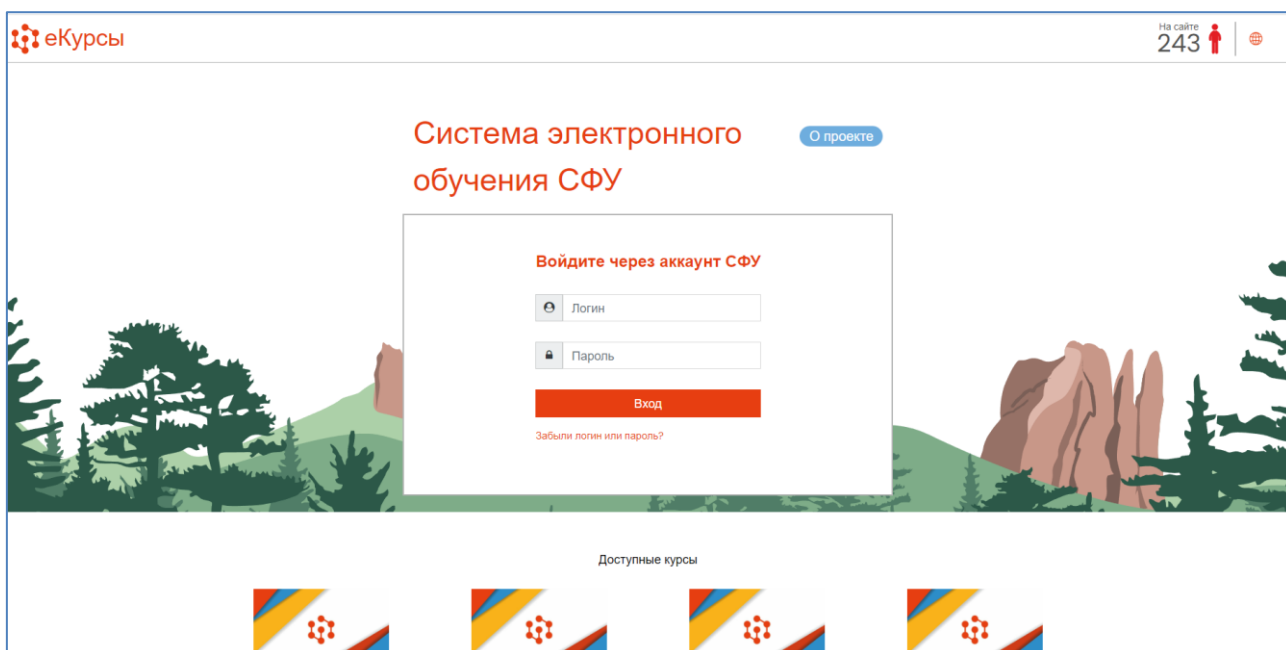


Рисунок 1 – Электронный курс СФУ

В данной системе размещены курсы от преподавателей различных институтов, входящих в состав СФУ. У каждого студента и преподавателя имеется учётная запись, под которой они авторизуются на данном ресурсе. Студентам по мере изучения материала открываются новые предметы, а преподавателям, которые ведут этот предмет, добавляются ученики. В данных дисциплинах размещаются учебные материалы для успешного прохождения курса, а также различные тесты и задания.

При прохождении тестов или заданий результат выполнения этих работ появляется у преподавателей, которые ведут этот курс, в зависимости от качества выполнения заданий ученику ставится оценка, рассчитываемая системой автоматически, а оценки отображаются в ведомости у преподавателя, который ведёт этот курс.

Однако в силу специфики образования некоторым студентам необходимо выполнять задания в электронной среде, которых нет в системе e.sfu-kras.ru, эти задания могут находиться на каких-либо сторонних ресурсах, либо же размещаться локально у студента на домашнем ПК. Например, необходимо создать виртуальную машину в среде Virtual Box, установить на неё определенную операционную систему, зарегистрировать пользователя и развернуть локальный web-сервер. Чтобы проверить выполнение такой работы, преподавателю необходимо, чтобы студенты демонстрировали результаты своих работ лично, на компьютерах университета либо же на своих собственных, к тому же студентов может быть много, и чтобы проверить у каждого, и оценить качество работы потребуется немало времени и ресурсов. Поэтому было решено облегчить этот процесс созданием специальной системы, при помощи которой можно было проверять работы на сторонних образовательных ресурсах, а результат сохранять в базу данных, из которой потом ресурс e.sfu-kras.ru будет забирать эти оценки и выставлять в свою систему. Тем самым упростив работу преподавателей.

Разработка web-службы планируется на языке PHP посредством Фреймворка Laravel. В системе должна быть предусмотрена система авторизи-

зации преподавателей для успешного подразделения курсов и выставленных оценок студентам. Принципом работы является служба, в которую преподаватель загружает свой исполнительный файл для проверки какой-либо работы на стороннем ресурсе, например, скрипт, содержащий команды для проверки необходимых данных которые должен был предоставить студент в работе, у скрипта формируется автоматически ссылка вида `https://domainname/checker/name_checker.php` по которой он становится доступным для загрузки с разрабатываемой платформы. Скрипт должен запускаться на стороне ученика, посредством нашей разрабатываемой службы, при помощи заранее созданного внешнего инструмента LTI интегрированного с моей платформой. При запуске инструмента LTI в курсе у студента появляется окно с инструкцией о запуске Чекера на его локальной машине где выполнялась лабораторная работа, например, командой «`curl https://domainname/checker/name_checker.php`», после запуска и выполнения команды отработанный скрипт отправляет результаты выполнения к нам на разрабатываемую платформу при помощи GET запроса, где мы принимаем и обрабатываем ответ, в результате чего записывается оценка в базу данных платформы. А именно какой студент запускал Чекер, сколько раз он был запущен, и оценки выполненной работы, где уже на основе имеющихся данных можно сформировать отчет.

Примерная схема данной работы изображена на рисунке 2.

Постановка требования при выполнении задачи.

Целью текущей бакалаврской работы является разработка платформа разработчика средств внешнего оценивания для e-курсов СФУ. Разработанная система должна обеспечивать возможность проверки работ вне образовательной системы СФУ `e.sfu-kras.ru`, и их оценивания.

Система должна удовлетворять следующим требованиям:

- система должна быть на основе web-платформы;
- обмен данными между web-службой и образовательной платформой СФУ должен быть организован при помощи протокола LTI;

- ресурс должен быть доступен из глобальной сети;
- иметь дружелюбный пользовательский интерфейс.

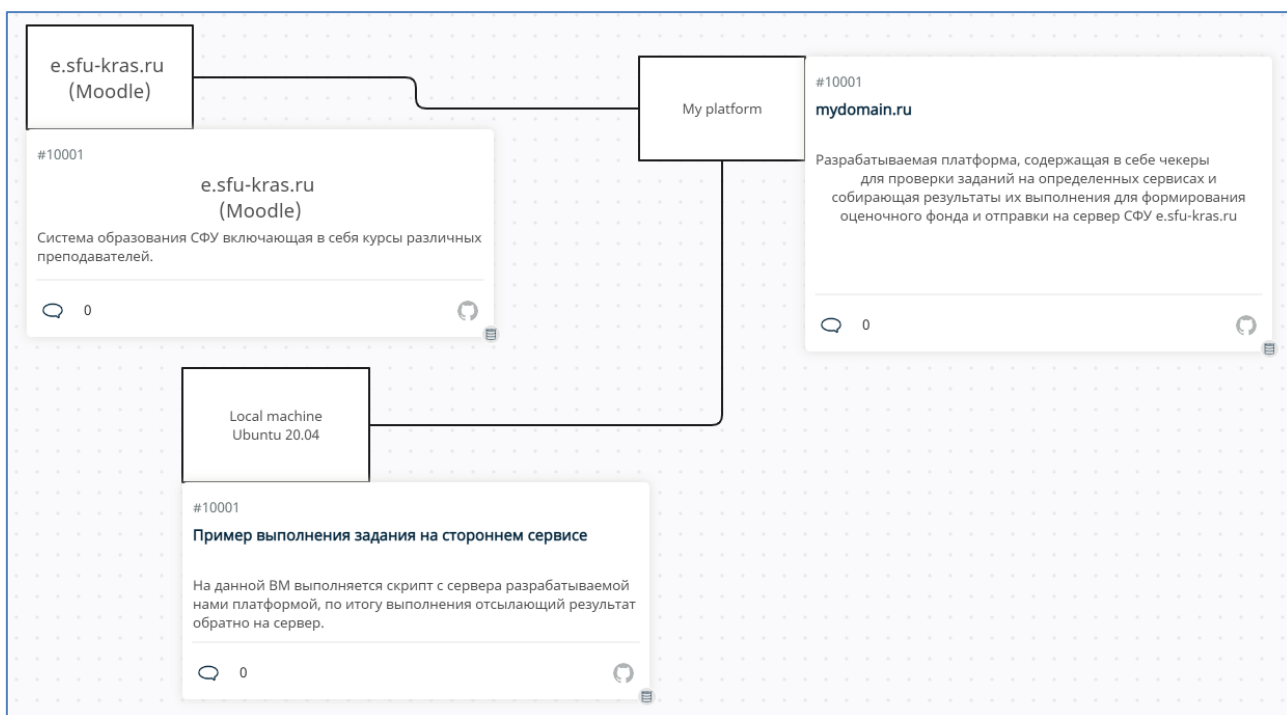


Рисунок 2 – Схема работы системы

Система должна быть удобна в пользовании и иметь возможность расширения:

- модульная разработка;
- использование СУБД;
- удобный ввод и вывод данных.

Система должна выполнять следующие задачи:

- обеспечить сбор и обработку в реальном времени для представления их в удобном для пользователя виде;
- создавать с ее помощью программные средства оценивания успеваемости студентов, выполнения ими практических заданий и т.п. (Чекеры), в т.ч. и работающие в автоматизированном режиме.

1.2 Цели создания системы

Целями создания данной системы являются:

- автоматизация процесса оценивания проверочных работ;
- создание удобного функционального сервиса для преподавателей с возможностью добавлять свои решения;
- централизованное управление отчетностью об успеваемости обучающихся.

1.3 Функциональные возможности

Платформа должна реализовывать следующие функции:

- регистрация Администратором Пользователя или Администратора и ведение его профиля;
 - регистрацию в Платформе модулей (файлов), содержащих код Чеков на языке php;
 - регистрацию П-Курса в Платформе, привязка к нему идентификатора Курса из Moodle;
 - регистрацию П-Оценок в П-Курсе с привязкой к идентификаторам соответствующих Оценок и указанием способа ее вычисления на основании журнала П-Чеков. Ограничение максимального количества П-Чеков для П-Оценки;
 - регистрацию Потока в составе Курса;
 - назначение Пользователям прав доступа к Курсу и Поток (Потокам).
- Возможные роли: Разработчик, Лектор, Ассистент, Студент, Наблюдатель.
- ведение журнала П-Чеков и выставленных в результате П-Оценок;
 - вычисление значения П-Оценки, выставляемой Студенту, и регистрацию ее в П-Журнале;

- формирование отчетов по П-Журналам с возможностью фильтрации по Группе, начальным буквам фамилии, сортировки по алфавиту, значению оценки, даты ее выставления и т.д;
- формировать детальные отчеты по П-Чекам, связанные с П-Журналом, путем вызова соответствующих хуков Чекера;
- экспорт П-Оценок в соответствующие Оценки e-курса, в т.ч. в режиме экспорта всего П-Журнала;
- платформа должна поддерживать интерфейсы взаимодействия с системой Moodle e-курсов, в т.ч;
- с использованием элемента Внешний инструмент Moodle по протоколу LTI;
- с использованием элемента Гиперссылка Moodle по протоколу http(s) и механизма импорта оценок в формате XML;
- платформа должна предоставлять программный интерфейс с Чеке-ром, реализующий следующие возможности для Чекера;
- регистрировать пользователя в Платформе с назначением ему роли Студент для конкретного П-Потока;
- регистрировать в платформе свои функции php (хуки), привязанные к запросам/командам http или LTI со стороны Moodle;
- выставлять П-Оценку Студенту;
- платформа должна предоставлять программный интерфейс с Чеке-ром, реализующий следующий функционал со стороны Платформы;
- вызывать зарегистрированные хуки Чекера при поступлении запроса/команды со стороны Moodle с передачей сопутствующих параметров. Результат работы хуков возвращать в качестве результата запроса/команды;
- вызывать predefined хуки Чекера для формирования детальных отчетов по П-Журналам.

1.4 Анализ существующих решений

В настоящее время существует множество платформ и инструментов для разработки средств внешнего оценивания для LMS (learning management system). Некоторые из них являются коммерческими продуктами, а другие – открытыми проектами с открытым исходным кодом.

Одной из самых популярных платформ для разработки средств внешнего оценивания является Learning Tools Interoperability (LTI) [4].

Этот протокол позволяет интегрировать различные инструменты и приложения в учебный процесс, обеспечивая единый интерфейс для пользователей.

Одним из примеров использования протокола LTI является платформа Moodle. Эта система управления обучением имеет встроенную поддержку протокола LTI, что позволяет разработчикам создавать собственные средства внешнего оценивания, которые могут быть интегрированы в Moodle.

Еще одним примером платформы для разработки средств внешнего оценивания на базе протокола LTI является Blackboard. Эта коммерческая платформа предоставляет широкий спектр инструментов для создания и интеграции средств внешнего оценивания в учебный процесс.

Кроме того, существуют и другие открытые проекты с открытым исходным кодом, например, Open edX и Sakai.

Эти платформы также поддерживают протокол LTI и предоставляют возможность разработчикам создавать собственные средства внешнего оценивания.

В целом, выбор платформы для разработки средств внешнего оценивания зависит от многих факторов, включая требования к функциональности, бюджет, доступность ресурсов и технической экспертизы. Важно выбрать платформу, которая наилучшим образом соответствует потребностям конкретного проекта и позволяет эффективно решать поставленные задачи.

Конкретно под мои задачи, в настоящий момент известных аналогичных решений в коммерческой среде не имеется, так как проект разрабатывается под конкретную систему управления обучения e.sfu-kras.ru реализуемый посредством протокола LTI v1.3 [4].

Где в зависимости от задания разрабатывается свой чекер (модуль) проверки результата выполнения работы.

Система управления обучением (LMS) может использовать LTI для размещения учебного контента и инструментов, предоставляемых внешними сторонними системами, на своих сайтах, дистанционных учебных платформах, не требуя от пользователя – учащегося, студента, преподавателя входа на площадку отдельно во внешних системах.

2 Проектирование и разработка функциональной архитектуры

2.1 Выбор технологий и инструментов для разработки

Для создания web-службы был выбран язык программирования PHP и фреймворк Laravel. PHP является широко используемым языком программирования для создания веб-приложений. Он имеет большое сообщество разработчиков и обширную документацию, что облегчает процесс разработки. Laravel – это один из самых популярных фреймворков для разработки веб-приложений на PHP. Он предоставляет множество готовых решений для распространенных задач веб-разработки, таких как маршрутизация, работа с базой данных и аутентификация пользователей [7].

Выбор PHP и Laravel обусловлен несколькими причинами. Во-первых, PHP является широко используемым языком программирования для создания веб-приложений. Он имеет большое сообщество разработчиков и обширную документацию, что облегчает процесс разработки. Это позволяет ускорить процесс разработки и сосредоточиться на реализации ключевой функциональности системы.

Кроме того, использование фреймворка Laravel позволяет обеспечить высокий уровень безопасности системы. Фреймворк предоставляет множество готовых решений для защиты от распространенных угроз безопасности, таких как SQL-инъекции и XSS-атаки. Это позволяет обеспечить надежную защиту данных пользователей и предотвратить несанкционированный доступ к системе.

Для реализации клиентской части веб-приложения были выбраны технологии JavaScript, HTML и CSS [10]. Эти технологии являются стандартом в индустрии веб-разработки и широко используются для создания современных и функциональных веб-приложений.

JavaScript является мощным и гибким языком программирования, который позволяет создавать динамические и интерактивные элементы на веб-

страницах. С помощью JavaScript можно обрабатывать пользовательские события, такие как клики по кнопкам и нажатия клавиш, а также изменять содержимое страницы в реальном времени.

HTML (HyperText Markup Language) используется для создания структуры веб-страницы и определения ее содержимого. С помощью HTML можно создавать различные элементы страницы, такие как заголовки, параграфы, списки и таблицы.

CSS (Cascading Style Sheets) используется для описания внешнего вида веб-страницы. С помощью CSS можно определить цвета, шрифты, отступы и другие стилевые параметры элементов страницы.

Использование этих технологий вместе позволяет создавать современные и функциональные веб-приложения с динамическим содержимым и интерактивными элементами. Они имеют большое сообщество разработчиков и постоянно развиваются, что обеспечивает их актуальность и применимость для решения широкого спектра задач.

Для хранения данных была выбрана СУБД MySQL так как она является одной из самых популярных реляционных баз данных, которая широко используется во многих сферах, включая веб-разработку, бизнес-аналитику, научные исследования и другие области. Она обладает высокой производительностью и надежностью, а также предоставляет широкий набор функций и возможностей для управления данными [1]. Данный продукт поддерживает многопоточность и многопользовательскую работу, что позволяет ей обрабатывать большие объемы данных и обеспечивать доступ к ним для множества пользователей одновременно. Она также поддерживает различные типы данных, включая числовые, текстовые, даты и времена, бинарные данные и другие, а также предоставляет широкий набор инструментов и интерфейсов для управления базами данных, включая консольные и графические утилиты, а также API для различных языков программирования, например, PHP – что может пригодиться в создании моего web-приложения.

3 Проектирование и разработка функциональной архитектуры

3.1 Проектирование функциональности системы

Архитектура системы состоит из нескольких компонентов. Серверная часть отвечает за обработку запросов от клиентов и взаимодействие с базой данных. Она написана на языке PHP с использованием фреймворка Laravel. Клиентская часть представляет собой веб-интерфейс для преподавателей и студентов. Веб-интерфейс позволяет преподавателям загружать исполнительные файлы для проверки работ на сторонних образовательных ресурсах и просматривать результаты выполнения скриптов.

Серверная часть системы обрабатывает запросы от клиентов и взаимодействует с базой данных. Она отвечает за авторизацию преподавателей, загрузку исполнительных файлов и обработку результатов выполнения скриптов. Серверная часть написана на языке PHP с использованием фреймворка Laravel. Laravel предоставляет множество готовых решений для распространенных задач веб-разработки, таких как маршрутизация, работа с базой данных и аутентификация пользователей [9].

Клиентская часть системы представляет собой веб-интерфейс для преподавателей и студентов. Веб-интерфейс позволяет преподавателям загружать исполнительные файлы для проверки работ на сторонних образовательных ресурсах и просматривать результаты выполнения скриптов.

Моё web-приложение будет разрабатываться на общепринятой модели архитектуры MVC, которая позволит мне создать приложение с более высокой степенью организации и структурированности. Модель MVC разделяет приложение на три основных компонента: модель, представление и контроллер.

Модель отвечает за управление данными и логикой приложения. Она содержит все необходимые методы для работы с базой данных, включая запросы на чтение, запись и изменение данных. Модель также может содер-

жать бизнес-логику приложения, которая обрабатывает данные и выполняет различные операции в соответствии с требованиями приложения [5].

Представление отвечает за отображение данных пользователю. Оно может быть представлено в виде HTML-страниц, графических элементов или других форматов. Представление использует данные, полученные из модели, и отображает их в удобном для пользователя виде.

Контроллер отвечает за управление пользовательским вводом и обработку запросов. Он получает данные от пользователя, обрабатывает их и передает их модели для дальнейшей обработки. Контроллер также может управлять навигацией в приложении, перенаправляя пользователя на различные страницы в зависимости от его действий.

Использование модели MVC позволяет мне разрабатывать приложение более эффективно и удобно. Она позволяет мне легко масштабировать приложение, добавлять новые функции и улучшать его производительность. Благодаря этому я смогу создать более качественное и удобное приложение для пользователей.

Разработка структуры входных и выходных сообщений.

Входные данные:

- внесение информации преподавателем;
- информация о курсе;
- авторизация;
- информация о пользователе.

Выходные данные:

- информация;
- отчетные данные.

Информационная база данных представлена на рисунке 3.

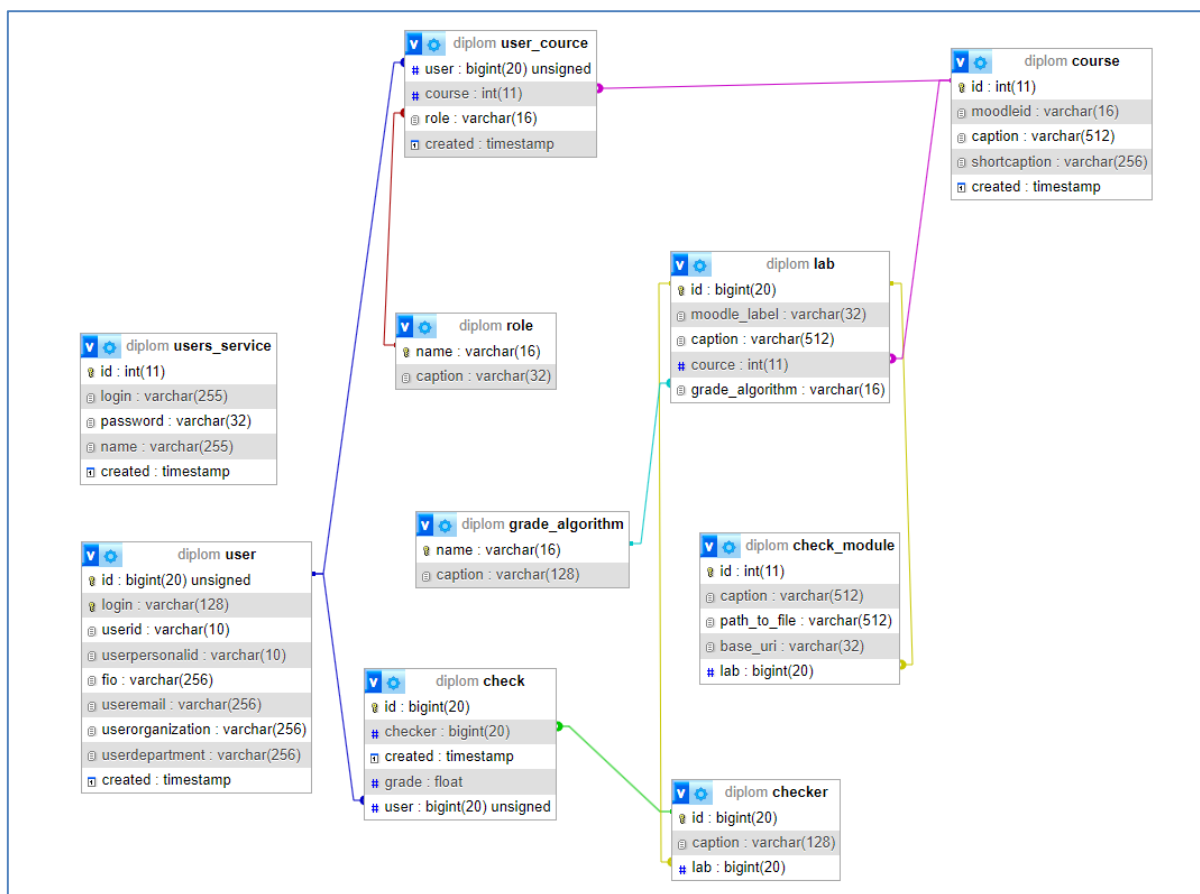


Рисунок 3 – Информационная база данных

На рисунке 3 изображена проектируемая, реляционная схема базы данных для платформы разработчика средств внешнего оценивания для е-курсов СФУ. Ниже представлены таблицы и поля, которые они в себя включают, так же тип данных которые они хранят и свойства полей (зависимости) первичный и вторичные ключи (Таблицы 1–10).

Таблица 1– User

Имя поля	Тип данных	Свойства поля	Назначение поля
id	Bigint(20)	Primary key	Уникальный ключ
Login	varchar(128)	—	Логин
Userid	varchar(10)	—	Id пользователя
Userpersonalid	varchar(10)	—	Персональный id
Fio	varchar(256)	—	Фамилия, имя, отчество
Usermail	varchar(256)	—	Почта
Userorganization	varchar(256)	—	Организация, к которой принадлежит пользователь
Userdepartament	varchar(256)	—	Отдел, к которому принадлежит пользователь
Created	timestamp	—	Дата создания

Таблица 2 – User_course

Имя поля	Тип данных	Свойства поля	Назначение поля
User	Bigint(20)	primary key	Пользователь
Course	Int(11)	–	Курс
Role	varchar(16)	–	Роль
created	timestamp	–	Дата создания

Таблица 3 – Role

Имя поля	Тип данных	Свойства поля	Назначение поля
Name	varchar(16)	–	Название
Caption	varchar(32)	–	Описание

Таблица 4 – Lab

Имя поля	Тип данных	Свойства поля	Назначение поля
id	Bigint(20)	Primary key	Уникальный ключ
Model_label	varchar(32)	–	Подпись из Moodle
Caption	varchar(512)	–	Описание
Course	Int(11)	foreign key	Курс
Grade_algorithm	varchar(16)	foreign key	Алгоритм вычисления оценки

Таблица 5 – Grade_algorithm

Имя поля	Тип данных	Свойства поля	Назначение поля
Name	Varchar(16)	Primary key	Название
Caption	varchar(128)	–	Описание

Таблица 6 – Course

Имя поля	Тип данных	Свойства поля	Назначение поля
id	Integer(11)	Primary key	Уникальный ключ
Moodleid	varchar(16)	–	Id в Moodle
caption	varchar(512)	–	Описание
shortcaption	varchar(256)	–	Короткое название
created	timestamp	–	Дата создания

Таблица 7 – Check_module

Имя поля	Тип данных	Свойства поля	Назначение поля
id	Integer(11)	Primary key	Уникальный ключ
caption	varchar(512)	–	Описание
Path_to_file	varchar(512)	–	Путь к файлу
Base_url	varchar(32)	–	Адресная ссылка
lab	Bigint(20)	foreign key	Лабораторная работа

Таблица 8 – Checker

Имя поля	Тип данных	Свойства поля	Назначение поля
id	Bigint(20)	Primary key	Уникальный ключ
caption	varchar(128)	–	Описание
Lab	Bigint(20)	foreign key	Проверяемая лабораторная работа

Таблица 9 – Check

Имя поля	Тип данных	Свойства поля	Назначение поля
id	Bigint(20)	Primary key	Уникальный ключ
checker	Bigint(20)	foreign key	Чекер
created	timestamp	–	Дата создания
grade	float	–	Оценка
user	Bigint(20)	foreign key	Пользователь

Таблица 10 – User_service

Имя поля	Тип данных	Свойства поля	Назначение поля
id	Integer(11)	Primary key	Уникальный ключ
login	varchar(255)	–	Логин
password	varchar(32)	–	Пароль
name	varchar(255)	–	Имя
created	timestamp	–	Дата создания

3.2 Проектирование архитектуры системы

Основные функции системы включают в себя авторизацию преподавателей, загрузку исполнительных файлов и обработку результатов выполнения скриптов. Авторизация преподавателей реализована с использованием стандартных механизмов аутентификации, предоставляемых фреймворком Laravel. Это позволяет преподавателям входить в систему с использованием своих учетных данных и получать доступ к функциям системы.

Загрузка исполнительных файлов осуществляется через веб-интерфейс. Преподаватель может выбрать исполнительный файл на своем компьютере и загрузить его в систему. После загрузки файла, он отправляется на сервер, где запускается и выполняется. Результат выполнения скрипта обрабатывается и сохраняется в базе данных. Обработка результатов выполнения скриптов осуществляется на сервере. Сервер получает результат выполнения скрипта и анализирует его. На основе анализа формируется отчет о выполнении работы

студентом. Отчет содержит информацию о том, какой курс был проверен, каким студентом была запущена проверка и какая оценка была выставлена за выполненную работу. На основе полученных данных строится отчет с содержанием групп студентов, выполнивших работу и возможностью их сортировки. Эти данные отправляются обратно на сервер e.sfu-kras.ru, где заносятся в базу данных и отображаются в ведомости у преподавателя.

В заключение можно сказать, что проектирование архитектуры системы включает в себя реализацию таких ключевых функций, как авторизация преподавателей, загрузка исполнительных файлов и обработка результатов выполнения скриптов. Эти функции позволяют преподавателям загружать исполнительные файлы для проверки работ на сторонних образовательных ресурсах и просматривать результаты выполнения скриптов.

Бизнес процессы для данного веб-приложения выглядят следующим образом:

- преподаватель создает Чекры с помощью Платформы разработчика средств внешнего оценивания для е-курсов СФУ;
- чекры сохраняются в базе данных, хранящейся на сервере;
- преподаватель выбирает Чекры, которые он хочет использовать в курсе;
- система Moodle загружает Чекры из базы данных и использует их для оценки успеваемости студентов;
- результаты оценки сохраняются в базе данных на сервере веб-приложения;
- студенты могут просмотреть свои результаты оценки через систему Moodle;
- преподаватель может просматривать и редактировать Чекры через Платформу разработчика средств внешнего оценивания для е-курсов СФУ, а также видеть отчеты, сформированные после использования Чекры на стороне е-курсов системы Moodle;

– администратор системы может управлять пользователями и настройками Платформы разработчика средств внешнего оценивания для е-курсов СФУ через административный интерфейс;

– в случае необходимости, разработчик может обновить Платформу разработчика средств внешнего оценивания для е-курсов СФУ, добавив новые функции или исправив ошибки.

На рисунке 4 изображена IDEF диаграмма 1 уровня платформы разработчика средств внешнего оценивания для е-курсов СФУ.

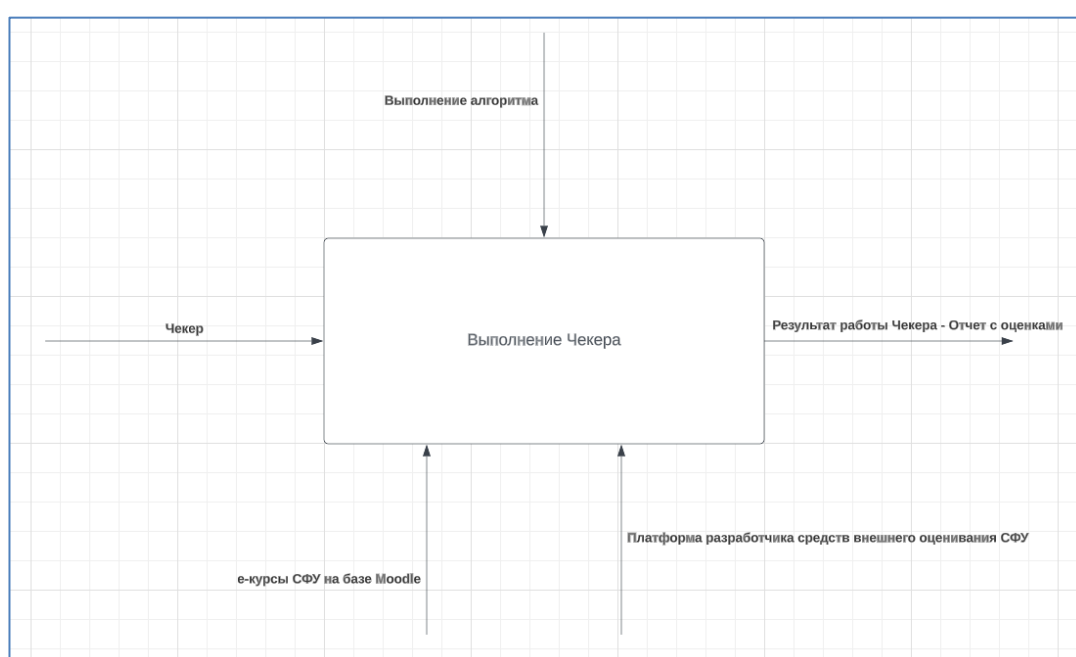


Рисунок 4 – edf0 диаграмма бизнес процессов 1 уровня

Входными данными для проекта являются:

– чекеры, загружаемые преподавателями или администратором в систему.

Выходными данными являются:

– отчёты с успеваемостью студентов запустившими Чекеры на стороне выполняемого задания.

На рисунке 5 изображена IDEF диаграмма 2 уровня платформы разработчика средств внешнего оценивания для е-курсов СФУ, где представлена декомпозиция процесса выполнения Чекеров на подпроцессы, а именно:

- загрузка Чекера в разрабатываемую платформу;
- исполнение его на стороне выполняемого задания студентом;
- возврат результата выполнения в разрабатываемую платформу;
- формирование отчёта с оценками на основе исполнения Чекера.

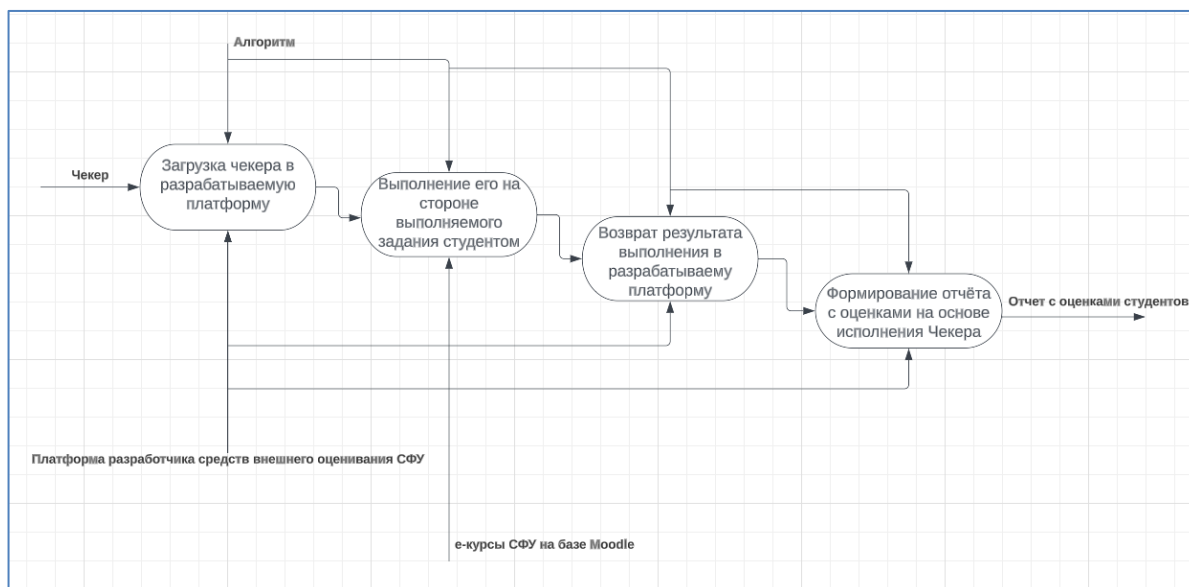


Рисунок 5 – Развернутая диаграмма edf0 бизнес процессов 2-го уровня

На рисунке 7 изображена карта разрабатываемой платформы, где на главной странице после авторизации, пользователю доступны такие разделы как «Пользователи», «Курсы», «Чекеры» и «Отчеты».

В трёх первых разделах доступно редактирование объектов, добавление новых и удаление уже существующих данных.

В разделе «Отчёты» есть возможность сформировать несколько типов отчетности, а именно формирование отчёта с алгоритмом «Grade» чья реализация доступна на рисунке 6, который вычисляет минимальную, среднюю и максимальную оценки студентов в зависимости от количества срабатывания Чекера в проверке лабораторной работы у студента.

Так же доступно формирование отчёта на основе выбранной лабораторной работы, т.е. список студентов и их оценки за выполнение конкретно выбранной лабораторной работы в курсе и формирование отчета по всем студентам присутствующим в курсе.

```

grade.php
1  <?php
2  if (isset($_POST['submit'])) {
3      $grades = $_POST['grades'];
4      $action = $_POST['action'];
5      $grades = explode(",", $grades);
6      if ($action == "min") {
7          $result = min($grades);
8      } elseif ($action == "max") {
9          $result = max($grades);
10     } elseif ($action == "average") {
11         $result = array_sum($grades) / count($grades);
12     }
13 }
14 ?>

```

Рисунок 6 – Реализация алгоритма Grade

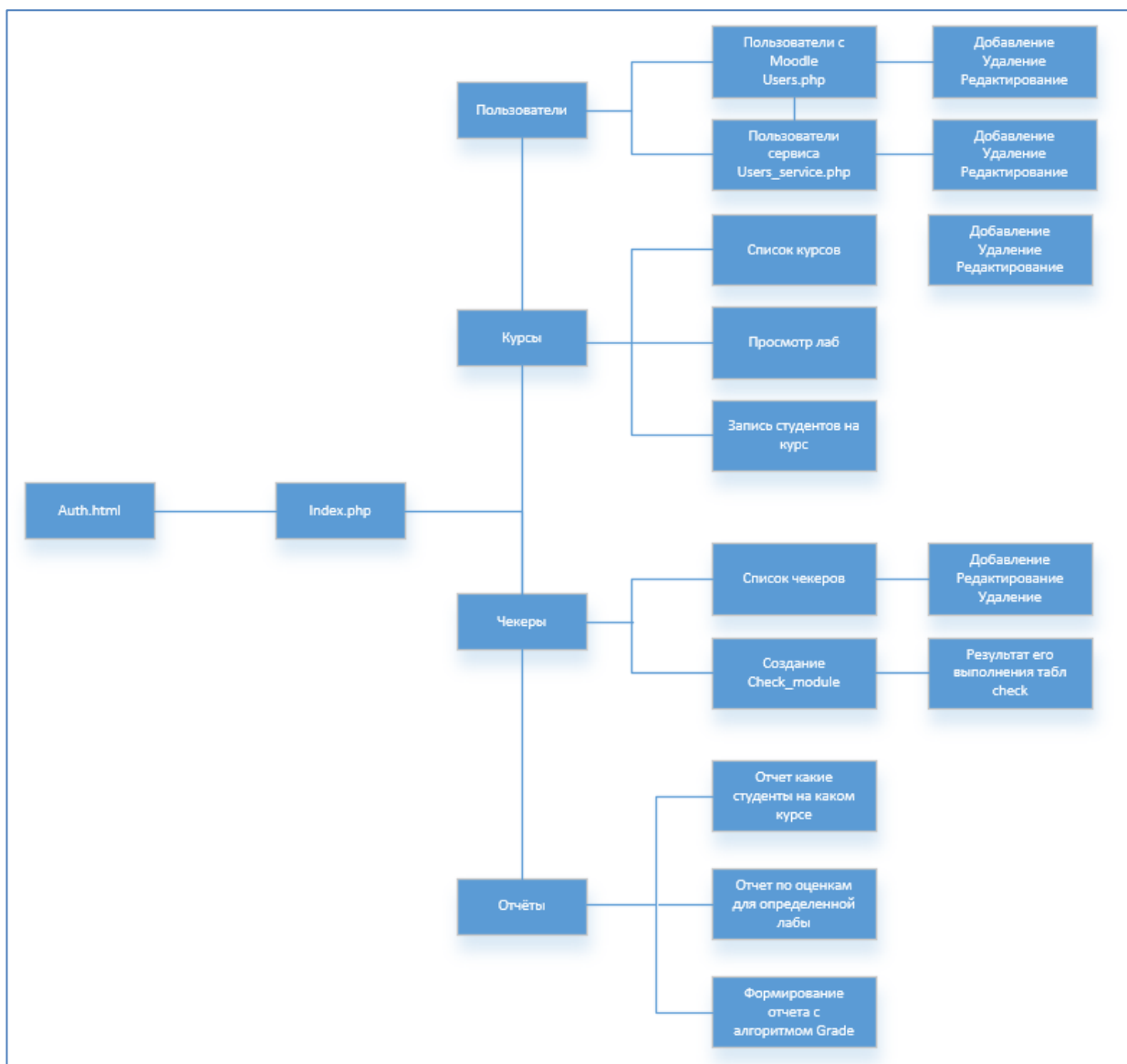


Рисунок 7 – Карта разрабатываемой платформы

4 Реализация функциональной архитектуры

База данных для данного программного продукта состоит из десяти таблиц [2] (Рисунок 8):

– таблица User – хранит данные о пользователях платформы (Таблица 1);

– таблица User_source – хранит данные о пользователях и их принадлежности к тому или иному курсу (Таблица 2);

– таблица Role – хранит данные о статусе пользователя в том или ином курсе (Таблица 3);

– таблица Lab – хранит данные о созданных курсах преподавателями в курсах из Moodle (Таблица 4);

– таблица Grade_algorithm – хранит данные о способе вычисления результатов выполнения лабораторных работ (Таблица 5).

– таблица Course – хранит информацию о существующих курсах в системе Moodle (Таблица 6);

– таблица Check_module – хранит информацию о созданных чекерах в разрабатываемой системе (Таблица 7);

– таблица Checker – хранит информацию о Чекере, задействованном в лабораторной работе (Таблица 8);

– таблица Check – хранит информацию о исполнении и результатах выполнения чекера на разрабатываемой платформе (Таблица 9);

– таблица User_service – хранит о пользователях платформы, и никак не связана с пользователями Moodle (Таблица 10).

База данных была реализована в СУБД PhpMyadmin, методом SQL запроса [13]:

```
CREATE SCHEMA IF NOT EXISTS `diplom` DEFAULT CHARACTER SET utf8_general_ci ;
```

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> check	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_general_ci	48,0 КиБ	-
<input type="checkbox"/> checker	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_general_ci	32,0 КиБ	-
<input type="checkbox"/> check_module	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_general_ci	32,0 КиБ	-
<input type="checkbox"/> course	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_general_ci	16,0 КиБ	-
<input type="checkbox"/> grade_algorithm	★ Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8mb4_general_ci	16,0 КиБ	-
<input type="checkbox"/> lab	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_general_ci	48,0 КиБ	-
<input type="checkbox"/> role	★ Обзор Структура Поиск Вставить Очистить Удалить	5	InnoDB	utf8mb4_general_ci	16,0 КиБ	-
<input type="checkbox"/> user	★ Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8mb3_general_ci	32,0 КиБ	-
<input type="checkbox"/> users_service	★ Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8mb4_general_ci	16,0 КиБ	-
<input type="checkbox"/> user_course	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_general_ci	64,0 КиБ	-
10 таблиц	Всего	10	InnoDB	utf8mb4_general_ci	320,0 КиБ	0 Байт

Рисунок 8 – Список таблиц базы данных

Подключение созданной базы данных было организовано при помощи строки подключения, содержащей информацию о названии и сервере, на котором хранятся данные (Рисунок 9).

```
$mysql = new mysqli('localhost','oracle','Treant123.','diplom');
$result = $mysql->query("SELECT * FROM `users_service` WHERE `login` = '$login' AND `password` = '$pass'");
$user = $result->fetch_assoc();
```

Рисунок 9 – Строка подключения БД в PHP и запрос для аутентификации пользователей

Форма авторизации в данном приложении написана на языке HTML и хранится в файле «auth.html», при переходе на главную страницу проекта открывается форма авторизации, которая изображена на рисунке 10, – данные из этой формы обрабатывает скрипт на языке PHP «auth.php».

Рисунок 10 – Форма авторизации

Аутентификация в разрабатываемом проекте реализована через БД, в которой хранятся пользователи и их учетные данные [8], таблица «User_service». При авторизации создается сессия, которая хранит в глобальном массиве такие переменные как «login» и «password» (Рисунок 11).

Таким образом, на каждой из функциональных страниц имеется сессия, для обработки этих передаваемых значений в переменных, что позволяет ограничить функционал проекта только для авторизованных пользователей.

```
<?php
// обработка авторизации с формы auth.html
session_start();
$login = filter_var(trim($_POST['login']),
FILTER_SANITIZE_STRING);
$password = filter_var(trim($_POST['password']),
FILTER_SANITIZE_STRING);

if($login == '' || $password == '')
echo "Введите логин или пароль";

$mysql = new mysqli('localhost', 'oracle', 'Treant123.', 'diplom');

$result = $mysql->query("SELECT * FROM `users_service` WHERE `login` = '$login' AND `password` = '$password'");
$user = $result->fetch_assoc();

if(count((array)$user) == 0 )
{
    echo "Такой пользователь не найден";
    exit();
}

$_SESSION['password'] = $user['password']; // записываем данные авторизации в массив сессии
$_SESSION['login'] = $user['login'];

setcookie("user", $user['login'], time()+3600, "/");
$mysql->close();
header('Location: index.php');

?>
```

Рисунок 11 – Скрипт авторизации пользователей на платформе

На рисунке 12 изображена функция для постоянного подключения к API Moodle с использованием библиотеки php-lti на языке PHP [12].

```
$moodleUrl = 'https://moodle.r-19.ru';
$consumerKey = '54623452346';
$consumerSecret = '139bc04ce6ba0078cd45181dd3387772';

$tool = connectToMoodleLTI($moodleUrl, $consumerKey, $consumerSecret);

// Использование подключения для вызова функций API
$response = $tool->makeServiceRequest('core_course_get_courses', ToolProvider::HTTP_GET);
```

Рисунок 12 – Использование функции подключения к API moodle

После авторизации, пользователь попадает на главную страницу проекта, в котором ему доступно контекстное меню с основными разделами платформы (Рисунок 13). Пункт меню «Пользователи» содержит два подраздела «Пользователи Moodle» и «Пользователи сервиса». В пользователях Moodle хранятся данные студентов и преподавателей курсов, а пользователях сервиса – непосредственно администраторы платформы.

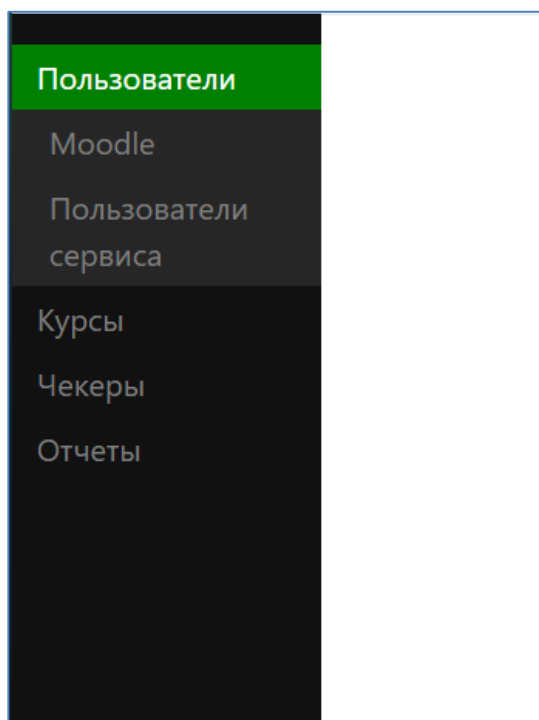


Рисунок 13 – Главное после авторизации пользователя

Создание, редактирование и удаление пользователей доступно при помощи функциональных кнопок, располагающихся снизу (Рисунок 14). При нажатии кнопки «Добавить пользователя» открывается форма для заполнения необходимых учетных данных (Рисунок 15). Чтобы отредактировать или удалить пользователя, необходимо отметить соответствующий «checkbox» напротив пользователя и нажать на кнопку «Редактировать» или «Удалить».

№	ID	Имя пользователя	Электронная почта	
1	1	Guest user	root@localhost	<input type="checkbox"/>
2	2	Admin User	aleksandr_chichkanov@mail.ru	<input type="checkbox"/>
3	3	Олег Шмелёв	oshmelev@mail.ru	<input type="checkbox"/>
4	4	Антон Петров	test120@r-19.ru	<input type="checkbox"/>
5	5	Андрей Тюкалов	test122@r-19.ru	<input type="checkbox"/>
6	6	Федоров Константин	test123@r-19.ru	<input type="checkbox"/>
7	7	Елена Абакумова	test124@r-19.ru	<input type="checkbox"/>
8	8	Test User	testuser@example.com	<input type="checkbox"/>
9	10	alex alex	alex@mail.ru	<input type="checkbox"/>
10	11	sorax tell	tell@r-19.ru	<input type="checkbox"/>
11	12	tell telr	alsd@gf.ru	<input type="checkbox"/>
12	13	Алексей Власов	test135@mail.ru	<input type="checkbox"/>
13	14	Валентина Николаенко	test136@mail.ru	<input type="checkbox"/>

Рисунок 14 – Функциональные кнопки для редактирования пользователей

Имя пользователя:

Пароль:

Имя:

Фамилия:

Email:

Рисунок 15 – Форма добавления пользователя Moodle

При переходе на вкладку «Курсы» главного меню отобразится список существующих курсов в системе (Рисунок 16), с возможностью создания нового курса. Эти данные были получены при помощи API функции «core_course_get_courses» [8].

№	ID	Название курса	Короткое название	ID категории	Описание курса	Формат описания	Дата начала курса	Дата окончания курса	
1	1	moodle.r-19.ru	moodle	0	Тест	1	01.01.1970	01.01.1970	Пользователи в курсе
2	2	Acara Site Administration	shells	1	Тестовый курс для раскрытия работы moodle с языком php протоколов LTE	1	23.11.2022	23.11.2024	Пользователи в курсе
3	3	Компьютерные сети	Сети ЭВМ	1	Курс по компьютерным сетям начального уровня. Видео курса доступно на Youtube бесплатно. Для удобства я создал два отдельных плейлиста для <i>лекций</i> и <i>практик</i> , где видео расположены в логической последовательности. Для некоторых видео есть лекции в текстовом виде. Пока их немного, но я надеюсь, что со временем удастся сделать больше.	1	29.03.2023	28.03.2024	Пользователи в курсе
4	4	sdfgsdfg	sdfgsdfg	1	sdfgsdfg	1	01.01.1970	01.01.1970	Пользователи в курсе
5	5	Критическое мышление	КМ	1	Курс создан на основе его практикующей деятельности	1	01.01.1970	01.01.1970	Пользователи в курсе
6	6	TellMeDoYoublood	TMDYB	1	Очень странное название для данного курса	1	01.01.1970	01.01.1970	Пользователи в курсе

Рисунок 16 – Отображение существующих курсов

Так же можно посмотреть какие элементы включает в себя тот или иной курс, нажав на ссылку в названии курса, после нажатия откроется новая страница (Рисунок 17) где будут отображены соответствующие данные.

General	
lti1	https://moodle.r-19.ru/mod/lti/view.php?id=7
Лабораторная работа №1	https://moodle.r-19.ru/mod/page/view.php?id=10
Topic 1	
getid	https://moodle.r-19.ru/mod/url/view.php?id=3
Лабораторная работа №1	https://moodle.r-19.ru/mod/assign/view.php?id=8
тест по сетям	https://moodle.r-19.ru/mod/quiz/view.php?id=9
Topic 2	
Как добавить место в логический том LVM ubuntu	https://moodle.r-19.ru/mod/lesson/view.php?id=4
Topic 3	
Topic 4	

Рисунок 17 – Просмотр элементов, входящих в состав курса

Доступна функция просмотра какие пользователи записаны на курс, при переходе по гиперссылке «Пользователи в курсе», открывается страница, на которой отображены студенты, записанные на курс в ячейке, которой был осуществлен переход (Рисунок 18). При нажатии на кнопку добавить пользователя откроется форма для выбора студентов, которые еще не записаны на курс. Выбрав студента его можно добавить в этот курс, нажав на кнопку «Записать на курс» (Рисунок 19).

Имя пользователя	Логин	ID пользователя	Email пользователя
Guest user	guest	1	root@localhost
Admin User	admin	2	aleksandr_chichkanov@mail.ru
Олег Шмелёв	oshmelev	3	oshmelev@mail.ru
Антон Петров	test120	4	test120@r-19.ru
Андрей Тюкалов	test122	5	test122@r-19.ru
Федоров Константин	test123	6	test123@r-19.ru
Елена Абакумова	test124	7	test124@r-19.ru
Test User	testuser	8	testuser@example.com
alex alex	test324	10	alex@mail.ru
sorax tell	sorax	11	tell@r-19.ru
tell telp	sick	12	alsd@gf.ru
Алексей Власов	test135	13	test135@mail.ru
Валентина Николаенко	test136	14	test136@mail.ru

Добавить пользователя

Рисунок 18 – Просмотр пользователей, подписанных на курс

Записать на курс

Студент: Курс: Компьютерные сети

- Guest user
- Admin User
- Олег Шмелёв**
- Антон Петров
- Андрей Тюкалов
- Федоров Константин

Рисунок 19 – Форма записи студента на курс

При переходе на вкладку «Чекеры» главного меню, открывается страница, которая отображает список созданных чекеров с возможностью их добавления, редактирования и удаления при помощи элемента «checkbox» [12] (Рисунок 20).

Чекеры

ID	Caption	Lab	base_uri	
1	check_for_script	3	https://lti.behobit.ru/check/check_for_script.php	<input type="checkbox"/>
2	check_for_users	5	https://lti.behobit.ru/check/check_for_users.php	<input type="checkbox"/>
3	check_timezone	3	https://lti.behobit.ru/check/check_for_timezone.php	<input type="checkbox"/>

Рисунок 20 – Список чекеров в платформе разработчика

При нажатии на кнопку «Добавить» появляется скрытая форма с необходимыми для заполнения полями: описанием и выбора лабораторной работы из курсов, к которой он будет прикреплен, после создания Чекера для него будет сформирована уникальная ссылка (по имени), по которой он станет доступен.

Описание:

Выбор лабораторной работы:

Файл не выбран

Рисунок 21 – Форма создания чекера

Выполнение чекера можно посмотреть, нажав на его название в виде ссылки которая отображена в столбце «caption». При нажатии на ссылку откроется страница с записями из БД с информацией о том, кто запускал данный чекер (id пользователя из таблицы БД «user»), сколько раз, в какое время и результат его выполнения в виде оценки (Рисунок 22).

Работа чекер - "check_for_script"

ID	Checker	Created	Grade	User
1	check_for_script	2023-05-18 09:17:35	67	3
1	check_for_script	2023-05-18 09:34:15	0	3
1	check_for_script	2023-05-18 09:41:11	67	3
1	check_for_script	2023-05-18 09:45:45	67	3
1	check_for_script	2023-05-18 10:23:21	84	3
1	check_for_script	2023-05-18 12:11:43	67	3
1	check_for_script	2023-05-18 15:15:15	67	3
1	check_for_script	2023-05-18 21:12:33	67	3
1	check_for_script	2023-05-18 22:32:54	84	3

Рисунок 22 – Результаты выполнения чекера «check_for_script»

При переходе на вкладку «Отчёты» главного меню, открывается страница на которой можно настроить и сформировать отчёт согласно заданным параметрам, по студентам, по алгоритму вычисления оценок и по курсам на которые зачислены студенты (Рисунок 23).

Отчет

Студент: Оценка: Курс:

Рисунок 23 – Страница формирования отчёта по заданным параметрам

После выбора необходимых атрибутов для формирования отчёта, нужно нажать на кнопку «Сформировать отчёт», сработает скрипт reports.php и отчёт автоматически загрузится в браузере в формате xlsx [12]. Для примера был сформирован отчёт с параметрами, указанными на рисунке 24.

В результате на выходе мы получили Excel файл с отчётом о выполнении проверок лабораторных работ по заранее выставленному курсу и алгоритмом вычисления оценки (Рисунок 25).

Отчет

Студент: Оценка: Курс:

Рисунок 24 – Результаты выполнения чекера «check_for_script»

	A	B	C	D	E	F	G	H	I
1	ID	Checker	Created	Grade	User	Name			
2	1	check_for_sc	2023-05-18 09:17:35	67	3	Олег Шмелёв			
3	1	check_for_sc	2023-05-18 09:34:15	0	3	Олег Шмелёв			
4	1	check_for_sc	2023-05-18 09:41:11	67	3	Олег Шмелёв			
5	1	check_for_sc	2023-05-18 09:45:45	67	3	Олег Шмелёв			
6	1	check_for_sc	2023-05-18 10:23:21	84	3	Олег Шмелёв			
7	1	check_for_sc	2023-05-18 12:11:43	67	3	Олег Шмелёв			
8	1	check_for_sc	2023-05-18 15:15:15	67	3	Олег Шмелёв			
9	1	check_for_sc	2023-05-18 21:12:33	67	3	Олег Шмелёв			
10	1	check_for_sc	2023-05-18 22:32:54	84	3	Олег Шмелёв			
11	max	84							

Рисунок 25 – Сформированный отчет по заданным параметрам в формате xlsx

5 Тестирование разработанной системы

В процессе разработки серверной части были использованы следующие технологии и инструменты: язык программирования PHP, фреймворк Laravel, СУБД MySQL PhpMyadmin, веб-сервер Apache2 на базе ОС Ubuntu 20.04. Были реализованы функциональные возможности по регистрации пользователей и модулей, ведению профилей, журнала П-Чеков и П-Оценок, а также возможность вычисления и регистрации П-Оценок.

Для обеспечения безопасности данных на сервере были приняты меры по защите от несанкционированного доступа и хакерских атак. Это включает в себя использование SSL-сертификата для шифрования передачи данных, проверку входных данных на наличие SQL-инъекций и XSS-атак, а также использование хэш-функций для хранения паролей пользователей.

Процесс тестирования серверной части включал юнит-тестирование и функциональное тестирование. Были обнаружены ошибки и проблемы с авторизацией пользователей и вычислением П-Оценок, которые были исправлены.

6 Внедрение и оценка качества надежности функционирования ПО

Разрабатываемая система интегрирована с сервером e.sfu-kras.ru для передачи результатов выполнения работ. Эта интеграция позволяет обеспечить бесперебойную передачу данных между разрабатываемой системой и сервером e.sfu-kras.ru. После обработки результатов выполнения скриптов, они отправляются на сервер e.sfu-kras.ru через API. Сервер e.sfu-kras.ru получает данные и заносит их в базу данных [11]. Отображение результатов выполнения работ осуществляется в ведомости у преподавателя.

Интеграция с сервером e.sfu-kras.ru реализована с использованием API, секретного ключа, внешнего инструмента LTI и токена пользователя, созданного специально для работы с внешней системой и наделенным правами необходимыми для выполнения операций [3]. На рисунке 26 изображено создание специального токена на сервере Moodle.

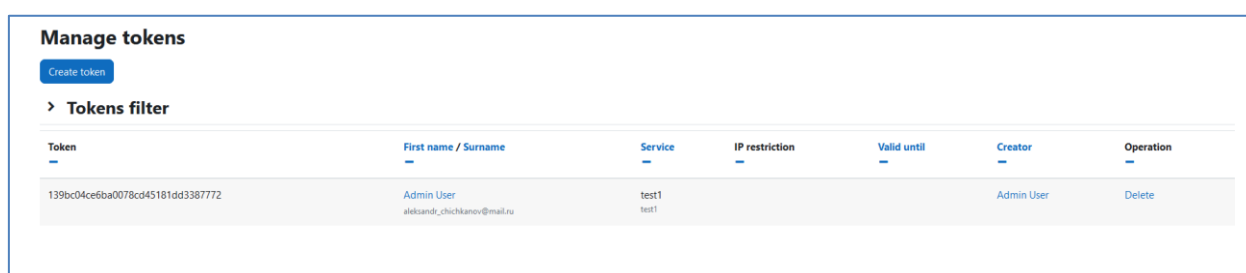


Рисунок 26 – Создание токена на сервере Moodle

В курсе Moodle добавляется специальный компонент, который называется «Внешний инструмент» который выделен красной рамкой на рисунке 27.

После добавления и настройки внешнего инструмента (Рисунок 28) он должен будет отображаться в курсе, в котором создавался и при нажатии на него студент будет взаимодействовать с разрабатываемой системой (Рисунок 29).

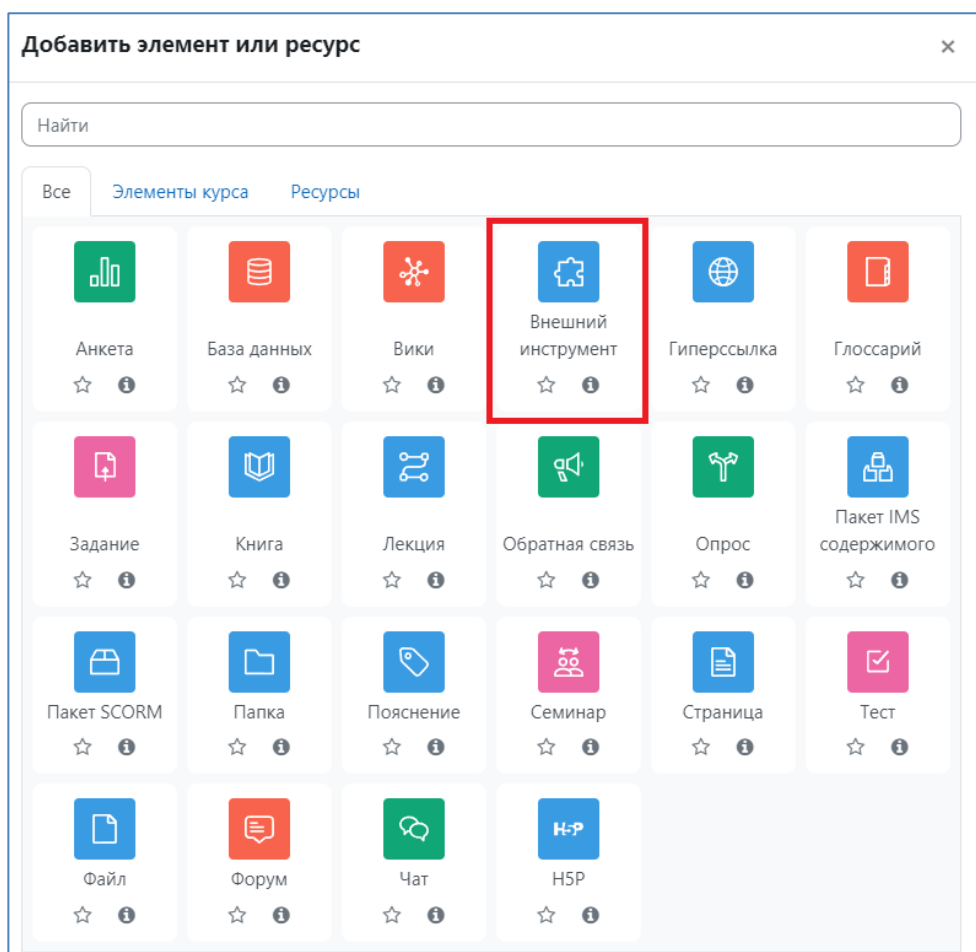


Рисунок 27 – Создание внешнего инструмента на сервере Moodle

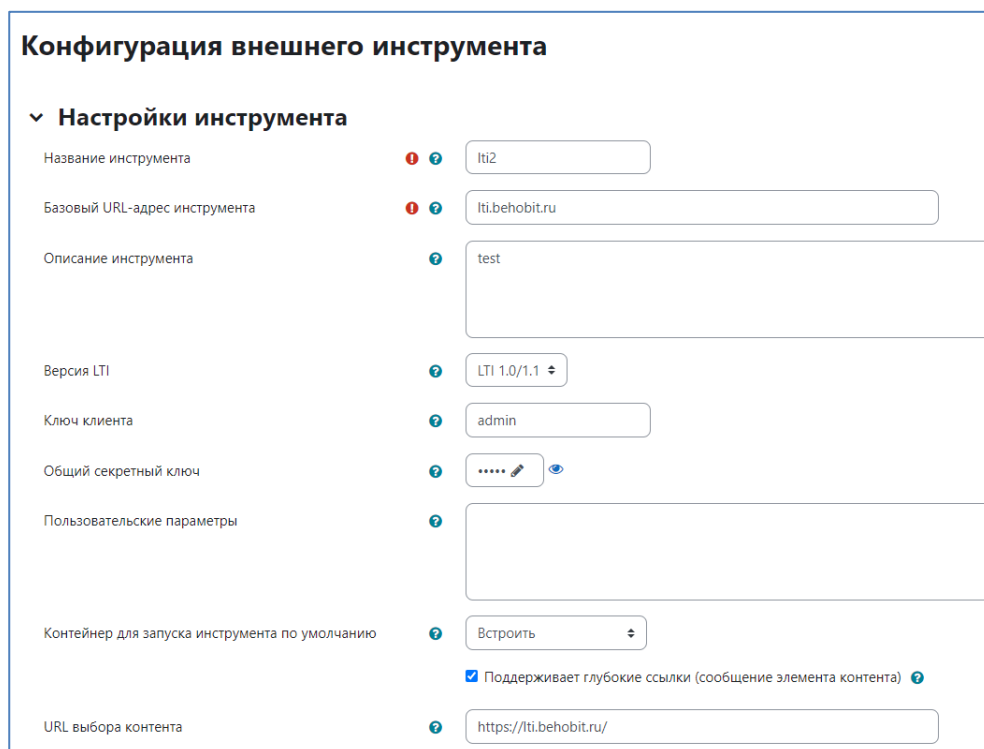


Рисунок 28 – Настройка LTI как элемента в Moodle

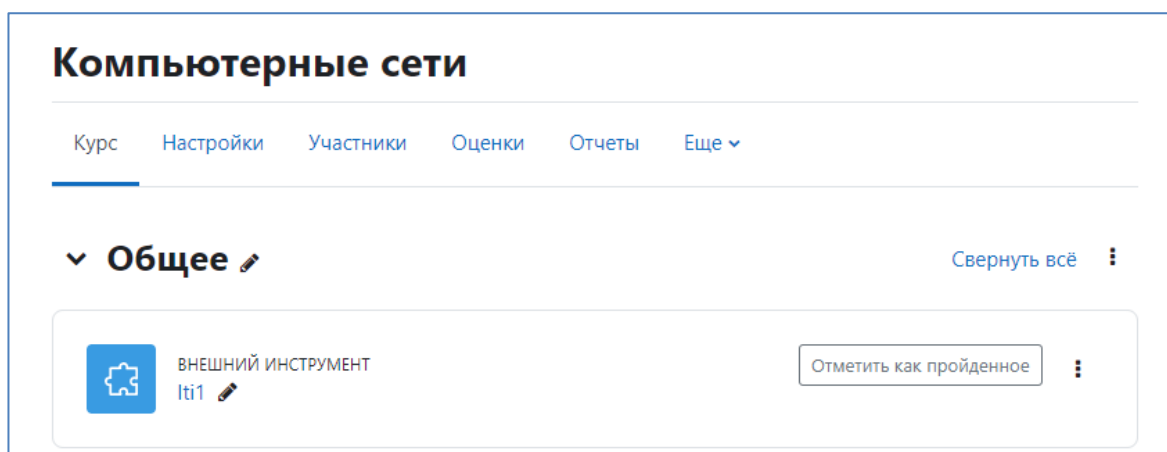


Рисунок 29 – Отображение созданного элемента в курсе

При нажатии студентом на созданный элемент LTI, при помощи встроенных в библиотеку классов и методов, был реализован механизм, который записывал в БД данные студента из Moodle в разрабатываемую платформу, а на выходе выдавал пользователю ссылку для запуска Чекера сформированную из таблицы «check_module» при помощи функции «doContent» (Рисунок 30).

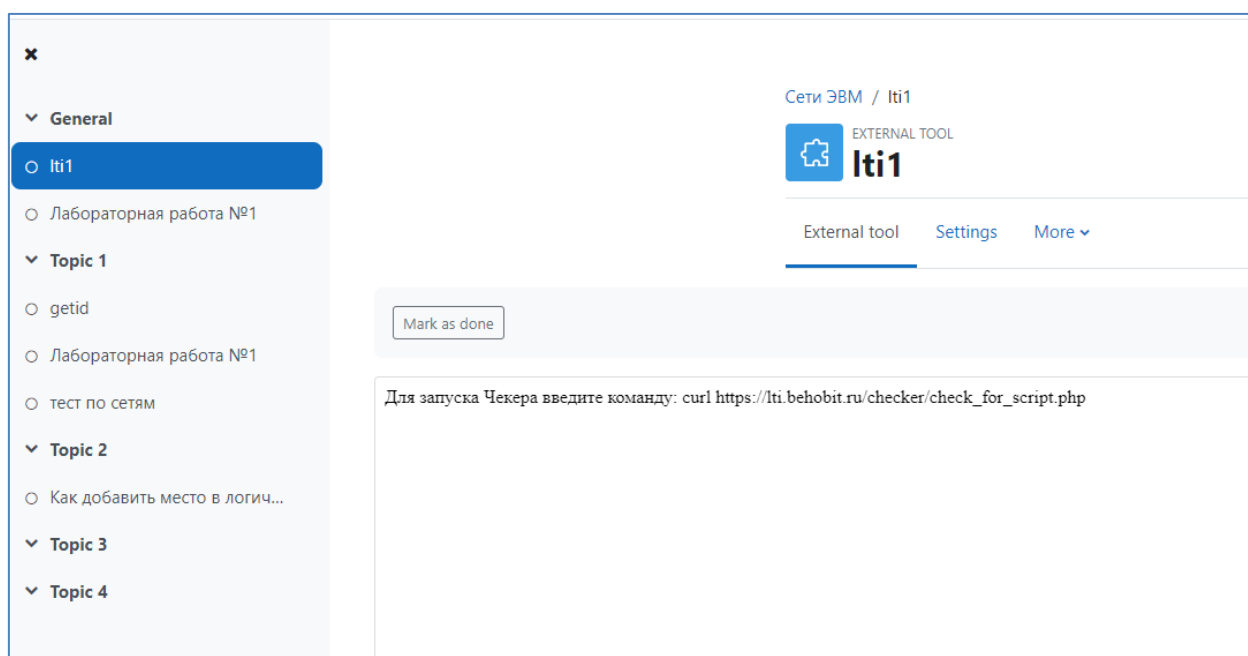


Рисунок 30 – Отображение необходимой информации студенту при запуске LTI инструмента

API предоставляет набор методов для обмена данными между системами. На рисунке 31 изображен пример добавления различных функций для работы с API Moodle. Разрабатываемая система использует эти методы для отправки результатов выполнения скриптов на сервер e.sfu-kras.ru. Сервер e.sfu-kras.ru обрабатывает полученные данные и заносит их в базу данных [3].

Add functions to the service "test1"			
Function	Description	Required capabilities	Edit
core_course_create_courses	Create new courses	moodle/coursecreate, moodle/coursevisibility	Remove
core_course_get_contents	Get course contents	moodle/courseupdate, moodle/courseviewhiddencourses	Remove
core_course_get_course_module	Return information about a course module		Remove
core_course_get_course_module_by_instance	Return information about a given module name and instance id		Remove
core_course_get_courses	Return course details	moodle/courseview, moodle/courseupdate, moodle/courseviewhiddencourses	Remove
core_enrol_get_enrolled_users	Get enrolled users by course id.	moodle/userviewdetails, moodle/userviewhiddendetails, moodle/courseuseremail, moodle/userupdate, moodle/siteaccessallgroups	Remove
core_user_create_users	Create users.	moodle/usercreate	Remove
core_user_get_users	search for users matching the parameters	moodle/userviewdetails, moodle/userviewhiddendetails, moodle/courseuseremail, moodle/userupdate	Remove
enrol_manual_enrol_users	Manual enrol users	enrol/manualenrol	Remove
enrol_manual_unenrol_users	Manual unenrol users	enrol/manualunenrol	Remove

[Add functions](#)

Рисунок 31 – Добавление функций во внешний инструмент Moodle

Отображение результатов выполнения работ осуществляется в ведомости у преподавателя. Преподаватель может просматривать результаты выполнения работ студентами в удобном для него формате. Это позволяет преподавателю быстро оценивать качество выполненных работ.

На рисунке 32 изображена схема интеграции веб-службы платформы разработчика средств внешнего оценивания для e-курсов СФУ с e-курсами СФУ. Посредством протокола LTI и взаимодействия с системой moodle по предоставленному API.

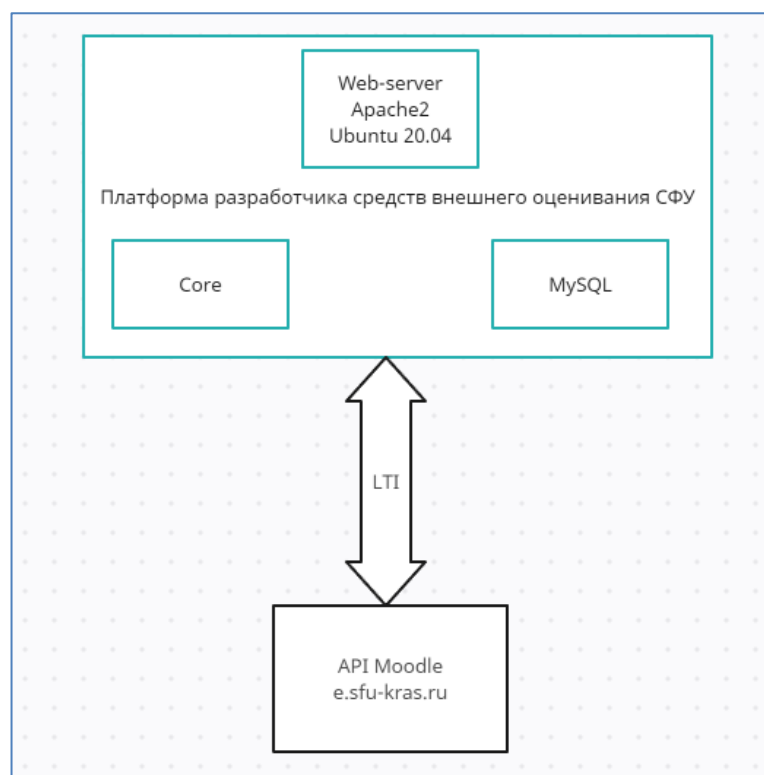


Рисунок 32 – Схема работы веб-службы «Платформы разработчика средств внешнего оценивания СФУ»

В заключение можно сказать, что интеграция с сервером e.sfu-kras.ru является важным компонентом разрабатываемой системы. Она позволяет обеспечить бесперебойную передачу данных между системами и упрощает работу преподавателей. Благодаря этой интеграции преподаватели могут просматривать результаты выполнения работ студентами в удобном для них формате и быстро оценивать работу.

ЗАКЛЮЧЕНИЕ

Предметная область разрабатываемого проекта была полностью изучена, концептуально рассмотрена, была спроектирована база данных для будущего приложения отражающая максимальные функциональные возможности. Разработка платформы будет вестись при помощи фреймворка Laravel версии не ниже 9.0 на языке PHP, данное web-приложение должно иметь максимальный функционал, а также простой и эргономичный дизайн для удобства пользователей.

В результате выполнения дипломной работы была разработана платформа разработчика средств внешнего оценивания для e-курсов СФУ для упрощения оценивания работ студентов преподавателями и интеграции внешнего инструмента с платформой на базе moodle.

В ходе выполнения работы были учтены требования самих сотрудников предприятия к разрабатываемому модулю.

Произведен выбор инструментальных средств разработки проекта. Для создания модуля был использован локальный сервер – Apache2, в качестве СУБД был использован MySQL, а языком разработки был выбран PHP.

В результате реализации данной дипломной работы были решены следующие задачи:

- исследование предметной области;
- построение логической и концептуальной модели информационной системы;
- выбор программных средств, для реализации модуля информационной системы;
- разработка платформы разработчика средств внешнего оценивания для e-курсов СФУ;
- тестирование платформы разработчика средств внешнего оценивания для e-курсов СФУ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон ; пер. Ю.Н. Артеменко. – Москва : Диалектика, 2017. – 768 с.
2. Документация JavaScript : сайт. – URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения: 24.03.2023).
3. Документация LMS Moodle External Tool : сайт. – URL: https://docs.moodle.org/402/en/External_tool (дата обращения: 15.03.2023).
4. Документация LMS Moodle LTI : сайт. – URL: https://docs.moodle.org/402/en/LTI_and_Moodle (дата обращения: 24.03.2023).
5. Документация MySQL : сайт. – URL: <https://dev.mysql.com/doc/> (дата обращения: 24.03.2023).
6. Документация PHP : сайт. – URL: <https://www.php.net/manual/ru/index.php> (дата обращения: 24.03.2023).
7. Документация к библиотеке PHP-LTI : сайт. – URL: <https://github.com/celtic-project/LTI-PHP/wiki> (дата обращения: 24.03.2023).
8. Документация по протоколу LTI : сайт. – URL: <https://www.imsglobal.org/спец/Lti/v1p3> (дата обращения: 24.03.2023).
9. Зандстра, М. PHP. Объекты, шаблоны и методики программирования / М. Зандстра. – Москва : Вильямс. – 2014. – 560 с.
10. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с.
11. Руководство LMS Moodle API : сайт. – URL: <https://moodledev.io/docs/apis> (дата обращения: 04.04.2023).
12. Справочные материалы по протоколу LTI : сайт. – URL: <https://docs.anthology.com/lti/getting-started-with-lti> (дата обращения: 24.03.2023).

13. Учебные материалы по JavaScript : сайт. – URL:
<https://learn.javascript.ru/> (дата обращения: 24.03.2023).

ПРИЛОЖЕНИЕ А

Листинг программы

Add_user_course.php

```
<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24 ми-
нуты
{
header('Location: /');
exit();
}

//Добавление пользователя в курс - форма
include "get_user_get_courses.php"; // подключаем модуль который получает в нашу форму данные с мудла
?>

<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Test</title>
<script src="scripts/headerFooter.js"></script>
</head>
<body>
<form action="enrol_user.php" method="post">
<label for="userid">Выберите пользователя:</label><br>
<select id="userid" name="userid">
<?php foreach ($users as $user) { ?>
<option value="<?php echo $user->id; ?>"><?php echo $user->firstname . ' ' . $user->lastname; ?></option>
<?php } ?>
</select><br>
<label for="courseid">Выберите курс:</label><br>
<select id="courseid" name="courseid">
<?php foreach ($courses as $course) { ?>
<option value="<?php echo $course->id; ?>"><?php echo $course->fullname; ?></option>
<?php } ?>
```

```
</select><br>
<input type="submit" value="Записать на курс">
</form>
</body>
</html>
```

Addcourses.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Test</title>
```

```
</head>
<body>
<form action="create_course.php" method="post">
<label for="fullname">Название курса:</label><br>
<input type="text" id="fullname" name="fullname"><br>
<label for="shortname">Краткое название:</label><br>
<input type="text" id="shortname" name="shortname"><br>
<label for="categoryid">ID категории:</label><br>
<input type="text" id="categoryid" name="categoryid"><br>
<label for="summary">Описание курса:</label><br>
<textarea id="summary" name="summary"></textarea><br>
<input type="submit" value="Создать курс">
</form>
</body>
</html>
```

Adduser.php

```
<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24
минуты
{
header('Location: /');
```

```

exit();
}
if (isset($_POST['submit'])) {
// Обработка отправки формы
$username = $_POST['username'];
$password = $_POST['password'];
$firstname = $_POST['firstname'];
$lastname = $_POST['lastname'];
$email = $_POST['email'];

// Проверка введенных данных
if (empty($username) || empty($password) || empty($firstname) || empty($lastname) || empty($email)) {
echo "Пожалуйста, заполните все поля формы";
} else {
// Добавление нового пользователя в Moodle
$token = '139bc04ce6ba0078cd45181dd3387772';
$domainname = 'https://moodle.r-19.ru';

$functionname = 'core_user_create_users';

$restformat = 'json';
// Параметры нового пользователя
$user = new stdClass();
$user->username = $username;
$user->password = $password;
$user->firstname = $firstname;
$user->lastname = $lastname;
$user->email = $email;
$user->auth = 'manual';
$user->idnumber = "";
$user->lang = 'en';
$user->timezone = 'Pacific/Port_Moresby';
$user->mailformat = 0;
$user->description = "";
$user->city = "";
$user->country = "";

```



```

// Отправка запроса на создание пользователя
header('Content-Type: text/plain');

$serverurl = "$domain-
name/webservice/rest/server.php?wstoken=$token&wsfunction=$functionname&moodlewsrestformat=$restformat"
;

require_once('./curl.php');

$curl = new curl;

// Отправка запроса на создание пользователя
if ($restformat == 'json') {
$resp = json_decode($curl->post($serverurl, array('users' => array($user))));
if (isset($resp->exception)) {
echo "Ошибка: " . $resp->message;
} else {
echo "Пользователь успешно создан с ID: " . reset($resp->id);
$new_url = 'users.php';
header('Location: '.$new_url);
}
} else {
echo "Формат ответа не поддерживается";
}
} else {
// Отображение формы для ввода данных нового пользователя
echo '<form method="post">
<label for="username">Имя пользователя:</label><br>
<input type="text" id="username" name="username"><br>
<label for="password">Пароль:</label><br>
<input type="password" id="password" name="password"><br>
<label for="firstname">Имя:</label><br>
<input type="text" id="firstname" name="firstname"><br>
<label for="lastname">Фамилия:</label><br>
<input type="text" id="lastname" name="lastname"><br>
<label for="email">Email:</label><br>
<input type="email" id="email" name="email"><br><br>
<input type="submit" name="submit" value="Создать пользователя">
</form>;
}

```

Auth.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Autorization</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhITQ8iRABdZL16O3oVMWStkQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">
<link rel="stylesheet" href="style/style_auth.css">
</head>
<body>
<div class="container mt-4">
<h1 class="display-1">Metascape</h1>
<form action="auth.php" method="post">
<input type="text" class="form-control" name="login" id="login" placeholder="Введите логин">
<br>
<input type="password" class="form-control" name="password" id="password" placeholder="Введите пароль">
<br>
<button class="btn btn-success" type="submit">Autorization</button>
</form>
</div>
</body>
</html>
```

Auth.php

```
<?php
// обработка авторизации с формы auth.html
session_start();
$login = filter_var(trim($_POST['login']),
FILTER_SANITIZE_STRING);
$pass = filter_var(trim($_POST['password']),
FILTER_SANITIZE_STRING);

if($login == "" || $pass == "")
echo "Введите логин или пароль";
```

```

$mysql = new mysqli('localhost','oracle','Treant123.','diplom');

$result = $mysql->query("SELECT * FROM `users_service` WHERE `login` = '$login' AND `password` = '$pass'");
$user = $result->fetch_assoc();

if(count((array)$user) == 0 )
{
echo "Такой пользователь не найден";
exit();
}

$_SESSION['password'] = $user['password']; // записываем данные авторизации в массив сессии
$_SESSION['login'] = $user['login'];

setcookie("user", $user['login'], time()+3600, "/");
$mysql->close();
header('Location: index.php');

?>

```

Checker.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Создание объекта</title>
</head>
<body>
<form action="create.php" method="post">
<label for="description">Описание:</label>
<input type="text" name="description" id="description"><br><br>
<label for="lab_work">Выбор лабораторной работы:</label>
<select name="lab_work" id="lab_work">
<option value="1">Лабораторная работа 1</option>
<option value="2">Лабораторная работа 2</option>
<option value="3">Лабораторная работа 3</option>
</select><br><br>

```

```

<input type="submit" value="Создать">
</form>
</body>
</html>

```

Cources.php

```

<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24
минуты
{
header('Location: /');
exit();
}

$token = '139bc04ce6ba0078cd45181dd3387772';
$domainname = 'https://moodle.r-19.ru';
$functionname = 'core_course_get_courses';
$restformat = 'json';
$params = array();
$serverurl = $domainname . '/webservice/rest/server.php' . '?wstoken=' . $token . '&wsfunction='.$functionname;
require_once('./curl.php');

$curl = new curl;
$restformat = ($restformat == 'json')?'&moodlewsrestformat=' . $restformat . ':';
$response = $curl->post($serverurl . $restformat, $params);
$courses = json_decode($response);

//print_r($courses);

// Предположим, что $courses содержит массив курсов, полученных из API Moodle
echo '<table border=3px;>';

echo '<tr><th>№</th><th>ID</th><th>Название курса</th><th>Короткое название</th><th>ID
категории</th><th>Описание курса</th><th>Формат описания</th><th>Дата начала курса</th><th>Дата
окончания курса</th></tr>';

$i = 1;
foreach ($courses as $course) {
echo '<tr>';
echo '<td>' . $i . '</td>';
echo '<td>' . $course->id . '</td>';
echo '<td><p><a href="get_courses_mods.php?courseid=' . $course->id . '">' . $course->fullname . '</a></p></td>';
//ссылка на получение всех лабораторных с этого курса
echo '<td>' . $course->shortname . '</td>';
echo '<td>' . $course->categoryid . '</td>';

```

```

echo '<td>' . $course->summary . '</td>';
echo '<td>' . $course->summaryformat . '</td>';
echo '<td>' . date('d.m.Y', $course->startdate) . '</td>';
echo '<td>' . date('d.m.Y', $course->enddate) . '</td>';
echo '<td><a href="get_enrolled_users.php?courseid=' . $course->id . '">Пользователи в курсе</a></td>';
echo '</tr>';
$i++;
}
echo '</table>';
echo '</form>';
echo '<form action="addcourses.html" method="post">';
echo '<input type="submit" name="button" value="Создать курс">';
echo '</form>';
?>

```

Create_course.php

```

<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24 ми-
нуты
{
header('Location: /');
exit();
}

// Получение данных из формы
$fullname = $_POST['fullname'];
$shortname = $_POST['shortname'];
$categoryid = $_POST['categoryid'];
$summary = $_POST['summary'];

$token = '139bc04ce6ba0078cd45181dd3387772';
$domainname = 'https://moodle.r-19.ru';
$functionname = 'core_course_create_courses';
$restformat = 'json';

// Параметры курса
$course = new stdClass();
$course->fullname = $fullname;
$course->shortname = $shortname;

```

```

$course->categoryid = $categoryid;
$course->summary = $summary;
//$course->summaryformat = 'FORMAT_HTML';
$course->format = 'topics'; // Формат курса
$course->visible = 1; // Видимость курса

// Параметры запроса
$params = array('courses' => array($course));

// Формирование URL-адреса для отправки запроса
$serverurl = $domainname . '/webservice/rest/server.php' . '?wstoken=' . $token . '&wsfunction=' . $functionname;
require_once('./curl.php');
$curl = new curl;

// Отправка запроса
if ($restformat == 'json') {
    $resp = json_decode($curl->post($serverurl . '&moodlewsrestformat=' . $restformat, $params));
} else {
    $resp = xmlrpc_decode($curl->post($serverurl, xmlrpc_encode_request($functionname, $params)));
}

// Обработка ответа
if (isset($resp->exception)) {
    // Ошибка при выполнении запроса
    echo "Ошибка: " . $resp->message;
} else {
    // Вывод информации о созданном курсе
    echo "Курс создан: ID=" . $resp[0]->id;
    $new_url = 'courses.php';
    header('Location: '.$new_url);
}

```

Curl.php

```

<?php
//библиотека для удобной работы с передачей данных по ссылкам
class curl {

public $options;

```

```

public function __construct($options = array()) {
    $this->options = $options;
}

```

```

public function get($url, $params = array()) {
    $options = array(
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_HEADER          => false,
        CURLOPT_FOLLOWLOCATION    => true,
        CURLOPT_USERAGENT       => 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)',
        CURLOPT_AUTOREFERER     => true,
        CURLOPT_CONNECTTIMEOUT  => 120,
        CURLOPT_TIMEOUT          => 120,
        CURLOPT_MAXREDIRS       => 10,
    );
    $options = $options + $this->options;
    $ch = curl_init($url . '?' . http_build_query($params));
    curl_setopt_array($ch, $options);
    $content = curl_exec($ch);
    curl_close($ch);
    return $content;
}

```

```

public function post($url, $params = array()) {
    $options = array(
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_HEADER          => false,
        CURLOPT_FOLLOWLOCATION    => true,
        CURLOPT_USERAGENT       => 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)',
        CURLOPT_AUTOREFERER     => true,
        CURLOPT_CONNECTTIMEOUT  => 120,
        CURLOPT_TIMEOUT          => 120,
        CURLOPT_MAXREDIRS       => 10,
        CURLOPT_POST             => 1,
        CURLOPT_POSTFIELDS       => http_build_query($params),
    );
}

```

```

$options = $options + $this->options;
$ch = curl_init($url);
curl_setopt_array($ch, $options);
$content = curl_exec($ch);
curl_close($ch);
return $content;
}
}

```

Enrol_user.php

```

<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24 ми-
нуты
{
header('Location: /');
exit();
}

// запись студента на курс , принимает данные из формы add_user_course.php
// Получение данных из формы
$userid = $_POST['userid'];
$courseid = $_POST['courseid'];

$token = '139bc04ce6ba0078cd45181dd3387772';
$domainname = 'https://moodle.r-19.ru';
$functionname = 'enrol_manual_enrol_users';
$restformat = 'json';

// Параметры записи на курс
$enrolment = new stdClass();
$enrolment->roleid = 5; // ID роли студента
$enrolment->userid = $userid;
$enrolment->courseid = $courseid;

// Параметры запроса
$params = array('enrolments' => array($enrolment));

// Формирование URL-адреса для отправки запроса

```



```

$serverurl = $domainname . '/webservice/rest/server.php' . '?wstoken=' . $token . '&wsfunction=' . $functionname;
require_once('./curl.php');
$curl = new curl;

// Отправка запроса
if ($restformat == 'json') {
$resp = json_decode($curl->post($serverurl . '&moodlewsrestformat=' . $restformat, $params));
} else {
$resp = xmlrpc_decode($curl->post($serverurl, xmlrpc_encode_request($functionname, $params)));
}

// Обработка ответа
if (isset($resp->exception)) {
// Ошибка при выполнении запроса
echo "Ошибка: " . $resp->message;
} else {
// Вывод информации об успешной записи на курс
echo "Пользователь успешно записан на курс";
}

```

Get_courses_mods.php

```

<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24 ми-
нуты
{
header('Location: /');
exit();
}

// этот скрипт предназначен для получения данных о содержащихся в курсе элементах, таких как страни-
цы/гиперссылки и т.д
require_once('./curl.php');

$token = '139bc04ce6ba0078cd45181dd3387772';
$domainname = 'https://moodle.r-19.ru';
$functionname = 'core_course_get_contents';
$serverurl = $domainname . '/webservice/rest/server.php' . '?wstoken=' . $token . '&wsfunction='.$functionname;
$restformat = 'json';
$courseid = $_GET['courseid']; // принимаем значение с sources.php через GET запрос
//$courseid = 3; // ID курса

```

```

$curl = new curl;
$restformat = ($restformat == 'json')?'&moodlewsrestformat=' . $restformat."";
$params = array('courseid' => $courseid);
$resp = $curl->post($serverurl . $restformat, $params);
$data = json_decode($resp);
echo '<table>';
foreach ($data as $section) {
echo '<tr><td>' . $section->name . '</td></tr>';
foreach ($section->modules as $module) {
echo '<tr><td>' . $module->name . '</td><td>' . $module->url . '</td></tr>';
}
}
echo '</table>';

```

Get_enrolled_users.php

```

<?php
// этот скрипт отображает какие студенты записаны на определенный курс
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24 ми-
нуты
{
header('Location: /');
exit();
}

// Получение идентификатора курса из параметров запроса
$courseid = $_GET['courseid'];

$token = '139bc04ce6ba0078cd45181dd3387772';
$domainname = 'https://moodle.r-19.ru';
$functionname = 'core_enrol_get_enrolled_users';
$restformat = 'json';

// Параметры запроса
$params = array('courseid' => $courseid);

// Формирование URL-адреса для отправки запроса
$serverurl = $domainname . '/webservice/rest/server.php' . '?wstoken=' . $token . '&wsfunction=' . $functionname;
require_once('./curl.php');
$curl = new curl;

```

```

// Отправка запроса
if ($restformat == 'json') {
$resp = json_decode($curl->post($serverurl . '&moodlewsrestformat=' . $restformat, $params));
} else {
$resp = xmlrpc_decode($curl->post($serverurl, xmlrpc_encode_request($functionname, $params)));
}
// выводим табличкой
echo "<table border='1'>";
echo "<tr><th>Имя пользователя</th><th>ID пользователя</th><th>Email пользователя</th></tr>";
foreach ($resp as $user) {
echo "<tr>";
echo "<td>" . $user->username . "</td>";
echo "<td>" . $user->id . "</td>";
echo "<td>" . $user->email . "</td>";
echo "</tr>";
}
echo "</table>";
echo '<input type="submit" name="button" value="Добавить пользователя">';

```

Get_user_get_courses.php

```

<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24 ми-
нуты
{
header('Location: /');
exit();
}

// скрипт к которому обращается add_user_course.php для получения данных в форму (студенты и пользова-
тели)
$token = '139bc04ce6ba0078cd45181dd3387772';
$domainname = 'https://moodle.r-19.ru';
$restformat = 'json';
require_once('./curl.php');
$curl = new curl;

// Получение списка пользователей
$functionname = 'core_user_get_users';

```

```

$params = array('criteria' => array(array('key' => 'email', 'value' => '%')));
$serverurl = $domainname . '/webservice/rest/server.php' . '?wstoken=' . $token . '&wsfunction=' . $functionname;
if ($restformat == 'json') {
    $resp = json_decode($curl->post($serverurl . '&moodlewsrestformat=' . $restformat, $params));
} else {
    $resp = xmlrpc_decode($curl->post($serverurl, xmlrpc_encode_request($functionname, $params)));
}
$users = $resp->users;

// Получение списка курсов
$functionname = 'core_course_get_courses';
$params = array();
$serverurl = $domainname . '/webservice/rest/server.php' . '?wstoken=' . $token . '&wsfunction=' . $functionname;
if ($restformat == 'json') {
    $resp = json_decode($curl->post($serverurl . '&moodlewsrestformat=' . $restformat, $params));
} else {
    $resp = xmlrpc_decode($curl->post($serverurl, xmlrpc_encode_request($functionname, $params)));
}
$courses = $resp;

```

Index.php

```

<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24
минуты
{
    header('Location: /');
    exit();
}
?>

<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Main Page</title>

```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhITQ8iRABdZL16O3oVMWSktQOp6b7In1Z13/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">
<link rel="stylesheet" href="style/style_index.css">
</head>
<body>
<header class="header">
<div class="container_1">
<h1 class="display-1">Metascape</h1>
</div>
<div class="icon">
<?php
echo $_COOKIE['user'];
?>
<a href="out.php">Выход</a>
</div>
</header>
<main class="main">
<div>
<form>
<ul class="menu">
<li><a href="#">Пользователи</a></li>
<ul class="submenu">
<li><a href="users.php">Moodle</a></li>
<li><a href="users_service.php">Пользователи сервиса</a></li>
</ul>
<li><a href="courses.php">Курсы</a></li>
<li><a href="checker.html">Чекеры</a></li>
<li><a href="#">Отчеты</a></li>
<li><a href="design/example.html">Пример</a></li>
</ul>
</form>
</div>
<div class="content">
А тут будет выводиться какой то контент
</div>
</main>
</body>

```

```
<footer class="footer">
<p>©Авторские права никому не принадлежат, 2050. Все права защищены.</p>
</footer>
</html>
```

Out.php

```
<?php
// окончание сессии, разлогиниться
setcookie('user', ((array)$user)['name'], time() - 3600, "/");
header('Location: /');
?>
```

Users_service.php

```
<?php
// отображает список пользователей зареганных на моей платформе
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24
минуты
{
header('Location: /');
exit();
}
?>

<!DOCTYPE html>
<html>
<head>
<title>Пользователи платформы</title>
<link rel="stylesheet" type="text/css" href="style/style_users_service.css">
</head>
<body>
<?php
// добавление пользователей разрабатываемого сервиса для авторизации
$servername = "localhost";
$username = "oracle";
$password = "Treant123.";
$dbname = "diplom";

// Создаем соединение с базой данных
$conn = mysqli_connect($servername, $username, $password, $dbname);
```

```

// Проверяем соединение
if (!$conn) {
die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT login, name, created FROM users_service";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
// Выводим данные каждой строки
while($row = mysqli_fetch_assoc($result)) {
echo "login: " . $row["login"]. " - Name: " . $row["name"]. " - Created: " . $row["created"]. "<br>";
}
} else {
echo "0 results";
}

// Проверяем, была ли отправлена форма
if ($_SERVER["REQUEST_METHOD"] == "POST") {
// Получаем значения полей формы
$username = $_POST["login"];
$password = $_POST["password"];
$name = $_POST["name"];

// Создаем SQL-запрос для добавления нового пользователя
$sql = "INSERT INTO users_service (login, password, name) VALUES ('$username', '$password', '$name')";

if (mysqli_query($conn, $sql)) {
echo "Новый пользователь успешно добавлен";
} else {
echo "Ошибка: " . $sql . "<br>" . mysqli_error($conn);
}
}
}

```

```

mysqli_close($conn);

?>

<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" class="form"
style="display: none;">

<label for="login">Login:</label>

<input type="text" id="login" name="login"><br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password"><br><br>

<label for="name">Name:</label>

<input type="text" id="name" name="name"><br><br>

<input type="submit" value="Сохранить">

</form>

<button class="add-user-button">Добавить пользователя</button>

<script src="scripts/script.js"></script>

</body>

```

Users.php

```

<?php
session_start();

if(!isset($_SESSION['password'])) // проверка сессии на пароль, сессия хранит данные по умолчанию 24 ми-
нуты
{
header('Location: /');
exit();
}

// Получение списка пользователей
$token = '139bc04ce6ba0078cd45181dd3387772';
$domainname = 'https://moodle.r-19.ru';

$functionname = 'core_user_get_users';

$restformat = 'json';

// Получаем список пользователей с типом аутентификации 'manual'
$params = array('criteria' => array(array('key' => 'auth', 'value' => 'manual')));

$serverurl = $domainname . '/webservice/rest/server.php'. '?wstoken=' . $token . '&wsfunction='.$functionname;
require_once('./curl.php');

```



```

$curl = new curl;
$restformat = ($restformat == 'json')?'&moodlewsrestformat=' . $restformat:"";
$resp = $curl->post($serverurl . $restformat, $params);
$result = json_decode($resp);
$users = $result->users;
echo '<table>';
echo '<tr><th>№</th><th>ID</th><th>Имя пользователя</th><th>Электронная почта</th></tr>';
$i = 1;
foreach ($users as $user) {
echo '<tr>';
echo '<td>' . $i . '</td>';
echo '<td>' . $user->id . '</td>';
echo '<td>' . $user->fullname . '</td>';
echo '<td>' . $user->email . '</td>';
echo '</tr>';
$i++;
}
echo '</table>';
echo '<form action="adduser.php" method="post">';
echo '<input type="submit" name="button" value="Добавить пользователя">';
echo '</form>';

```

Министерство науки и высшего образования РФ

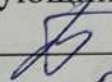
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

 О.В. Непомнящий
"20" 06 2023 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Платформа разработчика средств внешнего оценивания
для e-курсов СФУ

Руководитель


подпись


20.06.23
дата

ст. преподаватель

должность, ученая степень

О.В. Шмелёв

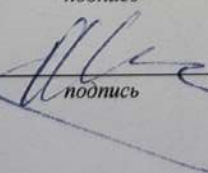
Выпускник


подпись

20.06.23
дата

А.Э. Чичканов

Нормоконтролёр


подпись

20.06.23
дата

ст. преподаватель

должность, ученая степень

О.В. Шмелёв

Красноярск 2023