

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  
« \_\_\_\_ » \_\_\_\_\_ 2024 г.

**БАКАЛАВРСКАЯ РАБОТА**

090301 Информатика и вычислительная техника

Веб-сайт креативного кластера

Руководитель	_____	_____	доцент, канд. физ.-мат. наук	К.В. Коршун
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		В.О. Леушкин
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____		К.В. Коршун
	<i>подпись</i>	<i>дата</i>		

Красноярск 2024

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

« \_\_\_\_ » \_\_\_\_\_ 2024 г.

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

Студенту Леушкину Владиславу Олеговичу

фамилия, имя, отчество

Группа КИ20-06Б Направление (специальность) 090301

номер

код

Информатика и вычислительная техника

наименование

Тема выпускной квалификационной работы: Веб-сайт

креативного кластера

Утверждена приказом по университету № 4858/С от 18.03.2024

Руководитель ВКР: К.В. Коршун, канд. физ.-мат. наук, доцент,

инициалы, фамилия, учёная степень, должность, место работы

НУЛ ТС ИКИТ СФУ

Исходные данные для ВКР:

1) сформулировать цели и задачи;

2) осуществить анализ веб-сайтов аналогов;

3) разработать и развернуть веб-сайт.

Перечень разделов ВКР:

1) анализ предметной области;

2) проектирование;

3) тестирования и развёртывание.

Перечень графического материала: Презентация доклада выступления

Руководитель ВКР

подпись

К.В. Коршун

инициалы, фамилия

Задание принял к исполнению

подпись

В.О. Леушкин

инициалы, фамилия

«    »       Г.

дата

## РЕФЕРАТ

Пояснительная записка выпускной квалификационной работы на тему «Веб-сайт креативного кластера» содержит 45 страниц, 14 рисунков, 2 таблицы, 10 источников.

ВЕБ-САЙТ, КРЕАТИВНЫЙ КЛАСТЕР, MONGODB, FLASK, RECAPTCHA.

Цель выпускной квалификационной работы: разработать веб-сайт для Красноярского креативного кластера.

Задачи, выполненные в ходе выполнения работы:

- проведён обзор существующих аналогов;
- составлено техническое задание на разработку веб-сайта;
- спроектирована архитектура разрабатываемого веб-сайта, основываясь на составленном техническом задании;
- разработан веб-сайт, придерживаясь разработанных ранее архитектурных решений;
- проведено тестирование функциональности сайта.

Практическая значимость разработки состоит в том, что сайт обеспечивает эффективное представление деятельности кластера, привлечение новых участников и партнеров, а также улучшение взаимодействия среди текущих участников. Сайт предоставляет возможность для регистрации и авторизации пользователей, редактирование профиля, загрузки и модерации публикаций, оставления и модерации комментариев.

В результате выполнения выпускной квалификационной работы был разработан функциональный веб-сайт для Красноярского креативного кластера. Задачи выполнены в полном объеме, цель достигнута.

## СОДЕРЖАНИЕ

Введение.....	3
1 Анализ предметной области .....	4
1.1 Обзор аналогов .....	4
1.1.1 Веб-сайт «Awwwards» .....	4
1.1.2 Веб-сайт «Creative at Home» .....	5
1.1.3 Веб-сайт «Дизайн завод».....	6
1.1.4 Вывод по результатам обзора .....	7
1.2 Спецификация требований.....	8
1.3 Выбор технологий разработки.....	9
1.4 Выводы по главе.....	10
2 Проектирование.....	11
2.1 Общая структура .....	11
2.2 Программный код .....	16
2.3 База данных.....	21
2.4 Интерфейс .....	24
2.5 Структура проекта.....	26
2.6 Модуль обработки URL.....	27
2.7 Модуль reCAPTCHA.....	28
2.8 Выводы по главе.....	29
3 Тестирования и развёртывание.....	31
3.1 Развёртывание .....	31
3.2 Методы тестирования .....	35
3.3 Выводы по главе.....	37
Заключение .....	38
Список использованных источников .....	39
ПРИЛОЖЕНИЕ А Результаты тестирования.....	40

## ВВЕДЕНИЕ

Создание веб-сайта для креативного кластера является важной стратегической мерой, необходимой для надлежащего представления деятельности и целей кластера, а также для привлечения внимания заинтересованных сторон и поощрения сотрудничества среди его участников.

Целью выпускной квалификационной работы является разработка веб-сайта креативного кластера Красноярска.

Для выполнения цели были поставлены следующие задачи:

- провести обзор существующих аналогов;
- составить техническое задание на разработку веб-сайта на основании обзора;
- спроектировать архитектуру разрабатываемого веб-сайта, основываясь на составленном техническом задании;
- разработать веб-сайт, придерживаясь разработанных ранее архитектурных решений;
- провести тестирование функциональности сайта.

Основной задачей создания веб-платформы является представление кластера и его активностей широкой аудитории. В контексте управления и организационной эффективности важно отметить, что эргономичный веб-сайт может значительно повысить имидж кластера и привлечь инвесторов, партнеров и новых участников. Кроме того, веб-платформа позволяет участникам кластера эффективно обмениваться информацией и идеями.

Веб-сайт кластера также может работать как платформа для привлечения новых участников и поддержки уже существующих. Он предоставляет возможность демонстрации работ, обсуждения проектов и обмена опытом. Подобные механизмы способствуют развитию кластера и укреплению его сообщества.

Таким образом, создание и поддержка веб-сайта для креативного кластера способствует его развитию и расширению.

## **1 Анализ предметной области**

### **1.1 Обзор аналогов**

В ходе данной работы будет проведена разработка веб-сайта для креативного сообщества, в котором будут реализованы следующие возможности:

- публикация работ пользователей;
- комментирование работ и просмотр комментариев;
- обмен идеями между пользователями;
- сохранение творческих работ.

На данный момент существует большое количество подобных веб-сайтов. Для определения критериев разработки будущего веб-сайта, необходимо рассмотреть существующие аналоги, и выделить основные преимущества и недостатки каждого из них.

#### **1.1.1 Веб-сайт «Awwwards»**

«Awwwards» – веб-сайт креативного кластера, состоящий из дизайнеров, разработчиков и агентств со всего мира, которые продемонстрировали свой талант и опыт в области цифрового дизайна [3]. На рисунке 1 представлен интерфейс сайта.

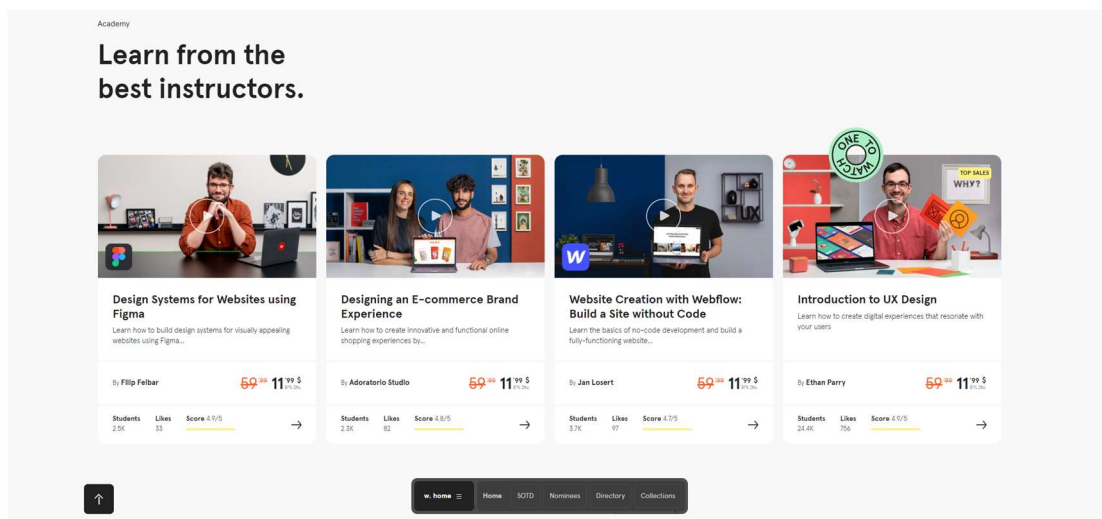


Рисунок 1 – Веб-сайт креативного кластера «Awwwards»

Сайтом предоставляются следующие возможности:

- площадка купли/продажи пользователями своих работ и курсов;
- просмотр всех планируемых кластером событий.

Главным недостатком данной платформы является отсутствие возможности комментировать публикуемый контент.

### 1.1.2 Веб-сайт «Creative at Home»

Проект «Creative at home» был создан для помощи художникам, испытывающим проблемы во время пандемии COVID-19. Целью проекта является разработка инновационной модели онлайн-обучения искусству людей, которые хотят развивать свои творческие способности [2]. На рисунке 2 представлен интерфейс сайта.



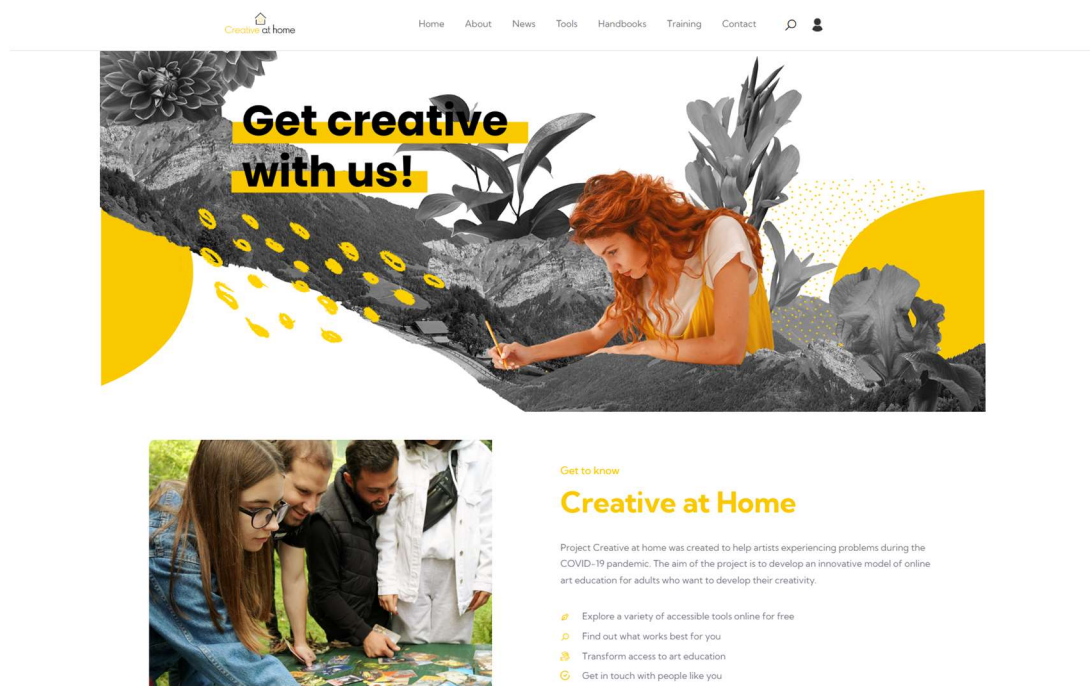


Рисунок 2 – Веб-сайт креативного кластера «Creative at Home»

У сайта удобный интерфейс, имеется возможность демонстрировать свои работы. Есть список избранных товаров и обратная связь. Недостатком сайта является отсутствие инструментов коммуникации для художников, а также отсутствие возможности продавать свои работы на данном веб-сайте

### 1.1.3 Веб-сайт «Дизайн завод»

Этот веб-сайт создан на основе существующего креативного кластера в Москве [4]. «Дизайн завод» – это современное городское пространство, расположенное на территории бывшего Хрустального завода. На рисунке 3 представлен интерфейс сайта.

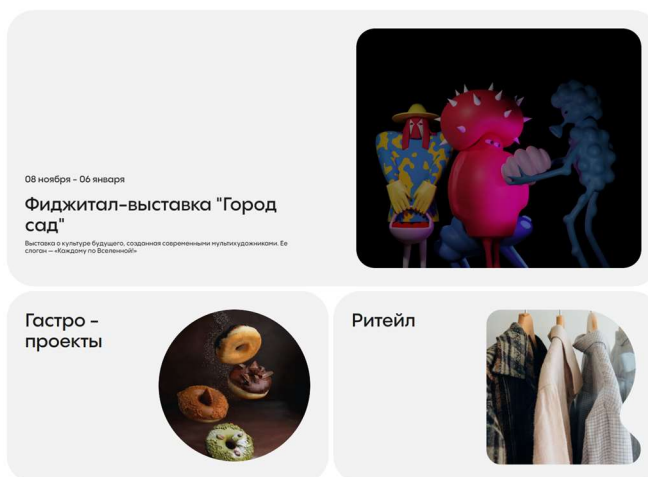


Рисунок 3 – Веб-сайт креативного кластера Дизайн завод

Данный сайт имеет удобный и не загруженный интерфейс. Реализован раздел «Резиденты», где можно посмотреть участников проекта. Из недостатков можно выделить ограниченный функционал сайта для пользователя, отсутствие возможности публиковать и комментировать работы.

#### 1.1.4 Вывод по результатам обзора

После проведенного анализа существующих аналогов были выявлены основные критерии для их сравнения. Эти критерии помогут определить основной функционал веб-сайта и его оформление. Для удобства сравнения, составлена таблица критериев (Таблица 1).

Таблица 1 – Критерии сравнения сайтов

Критерии	Awwwards	Creative at home	Дизайн завод
Публикация контента	+	+	–
Комментирование публикаций	+	–	–
Обратная связь	+	+	+
Дизайн	–	+	+
Эргономичный интерфейс	+	–	+
Раздел объявлений	+	+	+
Удобная галерея	–	+	+
Раздел резидентов	–	–	+

В результате анализа был сделан вывод о необходимости разработки эргономичного интерфейса, добавления возможности комментировать публикации не только другими авторами, но и пользователями, не имеющими роли автора. Также необходимо добавить возможность пользователям напрямую связаться с автором для дальнейшего обсуждения деталей покупки или совместной работы.

Таким образом, были определены основные требования к веб-сайту:

- современный, стильный, эргономичный интерфейс;
- наличие раздела «Галерея»;
- возможность публиковать работы;
- возможность комментировать работы;
- наличие раздела «Объявлений»;
- возможность добавления публикаций в личный список избранных;
- наличие разделов для публикации резидентов.

## **1.2 Спецификация требований**

Для реализации выпускной квалификационной работы по разработке креативного кластера необходимо определить, какие инструменты будут использованы. Для разработки был выбран фреймворк Flask, так как он позволяет с легкостью расширять и корректировать функционал веб-сайта. В качестве СУБД была выбрана NoSQL-система MongoDB.

В ходе проведенного анализа были выделены основные модули разрабатываемого сайта:

- главная страница веб-сайта;
- раздел «Афиша»;
- раздел «Мой профиль»;
- раздел «Галерея»;
- раздел «О нас»;
- модерирование контента, публикующегося на сайте;

- комментирование публикаций;
- раздел «Избранное»;
- раздел «Мои работы».

### 1.3 Выбор технологий разработки

HTML (Hypertext Markup Language) – или гипертекстовый язык разметки, который является основным для создания и отображения веб-страниц и приложений. Данный язык обеспечивает возможность заключать текст в теги – ключевые слова в угловых скобках. Данные ключевые слова не отображаются веб-браузером, а указывают браузеру параметры отображения того или иного компонента. Атрибуты часто используются для указания ссылки на блок, к которому впоследствии применяется язык стилей CSS. Например, атрибут 'href' в теге <a> используется для указания ссылки, по которой можно перейти, а атрибут 'src' в теге <img> используется для указания пути к изображению, которое нужно отобразить.

CSS (Cascading Style Sheets) – это язык стилей, который позволяет добавлять визуальные эффекты к веб-страницам, созданным с помощью HTML. В CSS используются теги, наименования классов и идентификаторы в качестве адресов, которые указывают, какой стиль необходимо применить к данному компоненту. Например, можно использовать тег "h1", чтобы выбрать все заголовки первого уровня на странице и применить к ним определенный стиль. С помощью CSS можно гибко структурировать компоненты и блоки на веб-странице, изменять цвет текста или элементов, их форму, шрифт, положение при наведении мыши и даже создавать анимации.

Flask – это веб-фреймворк, который помогает быстро создавать веб-сайты. Он использует простой и понятный синтаксис Python, что облегчает отладку и тестирование страниц. Кроме того, Flask можно настроить под разные задачи с помощью дополнительных библиотек и пакетов расширения, например,

Flask.Session – модуль, добавляющий поддержку сессий, что позволяет запоминать информацию о пользователе [7].

Python – это универсальный и гибкий язык программирования. Он широко применяется в разработке веб-сайтов, и работы с большим объемом данных. Python предоставляет большой набор модификаций, библиотек, а также собственную гибкую реализацию надстроек плагинов для языка под узкие задачи. Зачастую при разработке сайтов используют именно его, поскольку он имеет обширную документацию и легкий синтаксис, что способствует быстрой разработке веб-сайта [8].

MongoDB – это документоориентированная NoSQL-СУБД. Зачастую именно эту систему используют для хранения, предоставления больших объемов данных, которые собирает веб-сайт. Хранение данных предоставляется в JSON-подобных документах, которыми может обмениваться веб-сайт [6]. При добавлении новых данных при обмене JSON не нарушается схема базы данных. MongoDB поддерживает масштабирование на кластер из нескольких узлов. При большой нагрузке, манипуляциях и масштабировании система имеет высокую отказоустойчивость.

#### **1.4 Выводы по главе**

В данной главе был осуществлен разбор существующих аналогов, были выявлены основные критерии для проектирования веб-сайта, а также определены инструменты, которые будут использоваться в разработке.

## 2 Проектирование

### 2.1 Общая структура

Flask – это веб-фреймворк Python, который используется для разработки простых и масштабируемых веб-сайтов с использованием шаблонизатора Jinja2. Главным преимуществом этого фреймворка является его модульность и расширяемость.

Для обеспечения необходимой функциональности сайта кластера необходимо использовать такие модули фреймворка, как модули маршрутизации, обработки запросов, шаблонов, взаимодействия с базой данных. Благодаря этому обеспечивается упорядоченность исходного кода и возможность расширения функционала веб-сайта.

На рисунке 4 изображена клиент – серверная структура сайта.

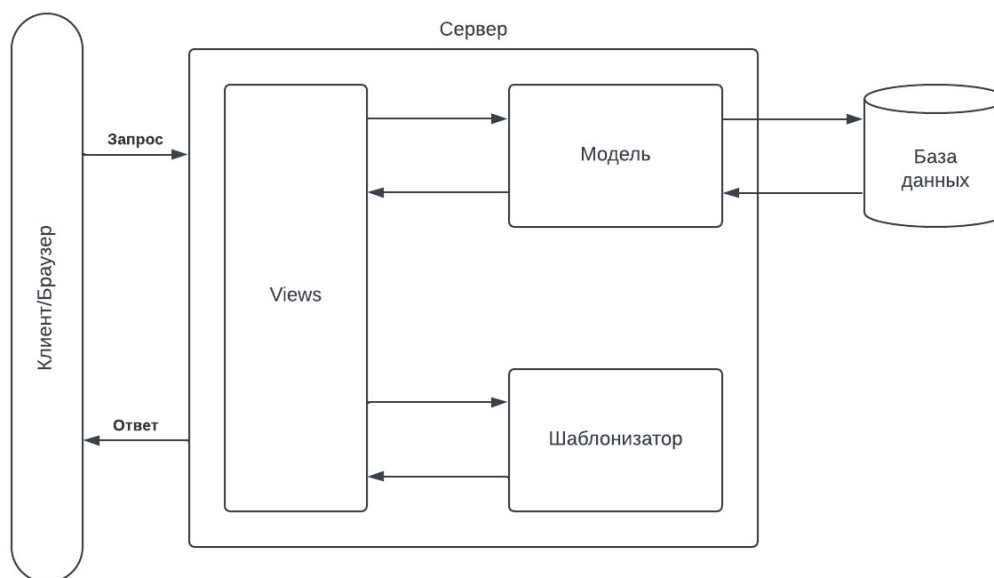


Рисунок 4 – Клиент – серверная структура веб-сайта

Основной функционал и описание взаимодействия пользователя с веб-сайтом опишем с помощью диаграммы прецедентов (Рисунок 5).



Рисунок 5 – Диаграмма прецедентов для пользователя и возможных для него ролей

Далее опишем все возможные прецеденты, предусловия их выполнения и их сценарии.

**Название: «Регистрация пользователя».**

Предусловие: пользователь не зарегистрирован на сайте.

Основной сценарий:

1. Пользователь заходит на страницу «Зарегистрироваться».
2. Пользователь заполняет предлагаемые формы необходимой информацией, а именно: имя пользователя, пароль, подтверждение пароля, загружает фотографию и проходит проверку reCAPTCHA.
3. Пользователь нажимает кнопку «Зарегистрироваться».

4. Система проверяет введенные данные, создает новый аккаунт пользователя и перенаправляет его на страницу авторизации.

**Название: «Авторизация пользователя».**

Предусловие: пользователь существует, но еще не был успешно авторизован на сайте.

Основной сценарий:

1. Пользователь заходит на страницу «Войти».
2. Пользователь заполняет необходимые данные: логин и пароль.
3. Пользователь нажимает кнопку «Авторизоваться».
4. Система проверяет правильность данных и дает пользователю права.

**Название: «Просмотр профиля».**

Предусловие: пользователь существует и успешно авторизован на сайте.

Основной сценарий:

1. Пользователь заходит в свой профиль.
2. Система загружает данные профиля пользователя.
3. Пользователь получает доступ к своим данным и произведениям.

**Название: «Редактирование профиля».**

Предусловие: пользователь уже успешно авторизован на сайте.

Основной сценарий:

1. Пользователь заходит на страницу редактирования профиля.
2. Пользователь изменяет формы с данными (например, фото, описание).
3. Пользователь нажимает кнопку «Сохранить».
4. Система сохраняет внесенные изменения и обновляет профиль пользователя.

**Название: «Просмотр публикаций».**

Предусловие: пользователь зашел на главную страницу сайта.

Основной сценарий:

1. Пользователь просматривает галерею публикаций.



2. Пользователь заходит на интересующую его публикацию.
3. Система отображает всю информацию о публикации, включая ее описание, автора и комментарии.

**Название: «Публикация комментария».**

Предусловие:

1. Пользователь уже успешно авторизован на сайте.
2. Пользователь зашел на публикацию, которую хочет прокомментировать.

Основной сценарий:

1. Пользователь видит блок с комментариями других пользователей.
2. Пользователь пишет свой комментарий.
3. Пользователь нажимает на кнопку «Публиковать».
4. Система обновляет информацию о публикации.

**Название: «Добавление публикации».**

Предусловие: пользователь авторизован на сайте и имеет роль «Автор».

Основной сценарий:

1. Пользователь переходит на страницу профиля.
2. Пользователь при нажатии на кнопку «Загрузить работу» видит окно для загрузки публикации.
3. Пользователь заполняет формы необходимой информацией и нажимает кнопку «Отправить на проверку».
4. Публикация отправляется на проверку модераторами.

**Название: «Удаление публикации».**

Предусловие:

1. Пользователь авторизован на сайте и имеет роль «Автор».
2. Пользователь выбрал публикацию, которую хочет удалить, и является ее автором.

Основной сценарий:

1. Пользователь заходит на страницу профиля.
2. Пользователь видит свои публикации в виде списка в разделе «Мои работы».

3. Пользователь нажимает значок удаления публикации.
4. После подтверждения система удаляет публикацию из каталога публикаций и профиля пользователя.

**Название: «Редактирование публикации».**

Предусловие:

1. Пользователь авторизован на сайте и имеет роль «Автор».
2. Пользователь выбрал публикацию, которую хочет редактировать, и является ее автором.

Основной сценарий:

1. Пользователь переходит на страницу профиля.
2. Пользователь видит свои публикации в виде списка в разделе «Мои работы».
3. Пользователь переходит на страницу публикации.
4. Пользователь нажимает на кнопку «Редактировать публикацию».
5. Пользователь редактирует формы с данными публикации.
6. Пользователь нажимает кнопку «Сохранить».
7. Система обновляет информацию о публикации.

**Название: «Просмотр публикаций для модерации».**

Предусловие: Пользователь авторизован на сайте и имеет роль «Модератор».

Основной сценарий:

1. Пользователь переходит на страницу профиля.
2. На странице пользователя в разделе «Модерация», он видит список всех публикаций, которые должны пройти проверку.
3. Если публикация соответствует правилам, модератор публикует ее, нажав на кнопку «Модерация пройдена».

Альтернативный сценарий:

1. Пользователь переходит на страницу профиля.
2. На странице пользователя в разделе «Модерация», он видит список всех публикаций, которые должны пройти проверку.

3. Если публикация не соответствует правилам, модератор описывает причину отклонения запроса на публикацию и отправляет ее на доработку, нажав на кнопку «Отклонить запрос на публикацию».

**Название: «Удаление комментариев».**

Предусловие: Пользователь авторизован на сайте и имеет роль «Модератор».

Основной сценарий:

1. Пользователь переходит на страницу профиля.
2. На странице пользователя в разделе «Модерация», он видит список всех комментариев, на которые были отправлены жалобы.
3. При нажатии на жалобу из списка, модератор переходит на страницу с публикацией, в которой нужно удалить комментарий.
4. Пользователь выбирает комментарий и нажимает кнопку «Удалить комментарий».
5. Система обновляет информацию о публикации.

## **2.2 Программный код**

Для визуализации структуры и различных связей объектов разрабатываемого сайта была создана следующая диаграмма классов (Рисунок 6).



Класс User представляет пользователя сайта, который имеет уникальный идентификатор, параметры профиля, роль и статус автора.

Класс User содержит следующие атрибуты:

- user\_id: int – уникальный идентификатор пользователя;
- user\_params: UserProfile – объект, содержащий параметры профиля пользователя;
- role: int – роль пользователя в системе (пользователь, модератор, администратор);
- author: bool – статус автора (истина, если пользователь является автором).

Класс User содержит следующие методы:

- editProfile(new\_info: UserProfile): void – метод, редактирующий профиль пользователя на основании объекта с новой информацией;
- viewAuthorProfile(user\_id: int): void – метод, реализующий просмотр профиля автора;
- addToFavorites(publication\_id: int): void – метод, добавляющий публикацию с соответствующим идентификатором в избранное;
- addComment(text: str, publication\_id: int): void – метод, добавляющий комментарий к публикации с соответствующим идентификатором;
- addReportComment(publication\_id: int): void – метод, реализующий функционал подачи жалобы на комментарий.

Класс UserProfile представляет профиль пользователя на сайте. Профиль должен содержать логин, пароль и аватар.

Класс UserProfile содержит следующие атрибуты:

- login: str – логин пользователя;
- password: str – хэш пароля пользователя;
- salt: str – соль для хэша;
- avatar: string – ссылка на файл аватара пользователя.

Класс Author наследуется от User и представляет автора публикаций на сайте. Автор имеет описание и может добавлять и редактировать свои публикации.

Класс Author содержит следующие атрибуты:

- description: str – описание автора.

Класс Author содержит следующие методы:

- addPublication(publication\_id: int): void – метод, отвечающий за добавление публикации;

- editDescription(new\_description: str): void – метод, отвечающий за редактирование описания профиля.

Класс Publication представляет публикацию на сайте. Публикация должна содержать идентификатор, фото, стоимость, список авторов, идентификатор автора, идентификатор модератора и описание.

Класс Publication содержит следующие атрибуты:

- id: int – уникальный идентификатор публикации;

- cost: int – стоимость публикации для покупки;

- authors\_id: List[User] – список авторов публикации;

- author\_upload\_id: int – идентификатор автора, загрузившего публикацию;

- moderator\_id: int – идентификатор модератора, проверившего публикацию;

- description: string – описание публикации.

Класс Event представляет мероприятие на сайте. Мероприятие должно содержать идентификатор, идентификатор модератора, название, описание и фото.

Класс Event содержит следующие атрибуты:

- event\_id: int – уникальный идентификатор мероприятия;

- moderator\_id: int – идентификатор модератора, добавившего мероприятие;

- name: str – название мероприятия;

- description: str – описание мероприятия;

– photo: string – фото мероприятия.

Класс Moderator наследуется от User и представляет модератора на сайте. Модератор может назначать авторов, проверять публикации и комментарии, а также добавлять мероприятия. Класс Moderator содержит следующие методы.

Класс Moderator содержит следующие методы:

– assignAuthor(user: User): void – метод, реализующий назначение пользователя автором;

– reviewPublication(publication\_id: int): void – метод, реализующий проверку публикации;

– reviewComment(comment\_id: int): void – метод, реализующий проверку комментария;

– addEvent(event: Event): void – метод, реализующий добавление мероприятия.

Класс Administrator наследуется от Moderator и представляет администратора на сайте. Администратор может добавлять модераторов.

Класс Administrator содержит следующие методы:

– addModerator(moderator: Moderator): void – метод, добавляющий модератора.

Класс Comment представляет комментарий, оставленный пользователем к публикации на сайте. Комментарий должен содержать идентификатор, идентификатор пользователя, идентификатор публикации, текст и идентификатор ответа.

Класс Comment содержит следующие атрибуты:

– comment\_id: int – уникальный идентификатор комментария;

– user\_id: int – идентификатор пользователя, оставившего комментарий;

– publication\_id: int – идентификатор публикации, к которой оставлен комментарий;

– text: str – текст комментария;

– reply\_id: int – идентификатор комментария, на который дан ответ.

Класс `CommentReport` представляет жалобу на комментарий. Жалоба должна содержать идентификатор пользователя, подавшего жалобу, и комментарий.

Класс `CommentReport` содержит следующие атрибуты:

- `id: int` – уникальный идентификатор жалобы;
- `user: User` – пользователь, подавший жалобу;
- `comment: Comment` – объект, представляющий комментарий, на который подана жалоба.

### **2.3 База данных**

В качестве СУБД для использования в веб-сайте креативного кластера нами была выбрана `MongoDB` – СУБД общего назначения, которая использует распределенную структуру. Эта СУБД реализована с использованием модели `NoSQL` (`Not Only SQL`). Данные в такой БД хранятся в виде документов или коллекций.

Для визуализации базы данных веб-сайта нами использована диаграмма «сущность – связь», которая представляет собой графическую визуализацию структуры базы данных и отображает отношения между наборами элементов в ней.

Диаграмма «сущность – связь» БД, использованной в веб-сайте, приведена на рисунке 7.



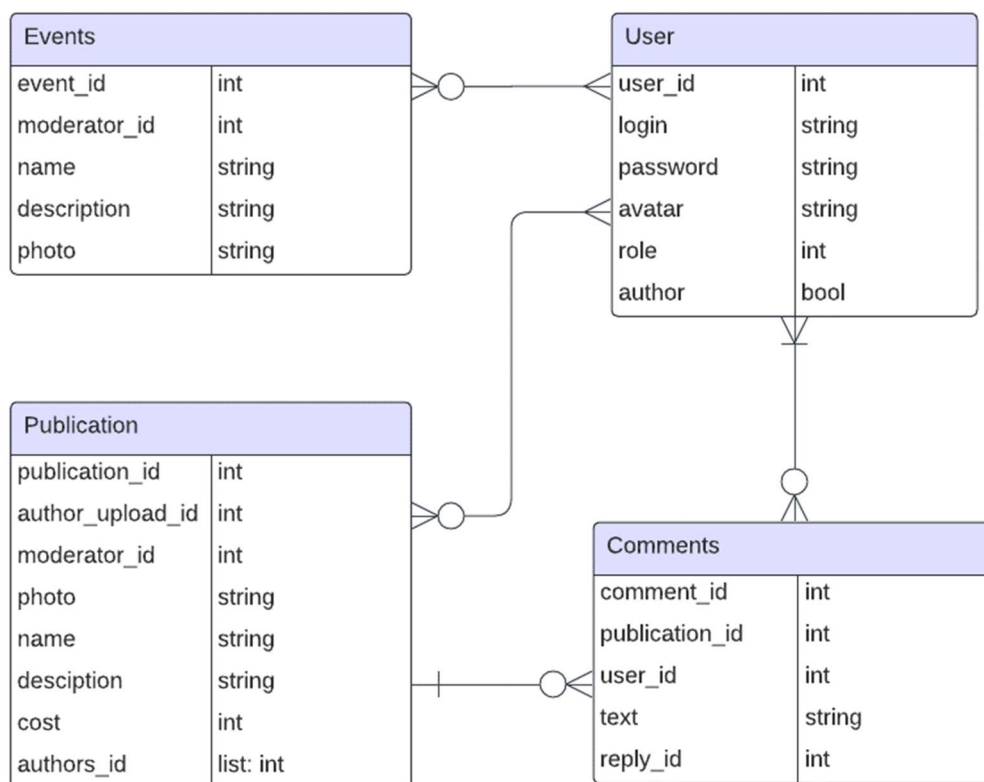


Рисунок 7 – Диаграмма базы данных «сущность-связь»

Описание коллекций базы данных веб-сайта приведено ниже:

#### User:

- user\_id:int – основной ключ, идентификатор пользователя;
- login: string – логин пользователя;
- password: string – хэш пароля пользователя;
- salt: string – соль для хэша;
- avatar: string – путь к аватара профиля;
- role: int – роль пользователя в системе;
- author: bool – статус автора.

#### Publication:

- publication\_id:int – основной ключ, идентификатор публикации;
- author\_upload\_id: int – идентификатор автора публикации;
- moderator\_id: int – идентификатор модератора;
- photo: string – путь к изображению публикации;

- name: string – название публикации;
- description: string – описание публикации;
- cost: int – цена произведения при продаже;
- authors\_id: list:int – список, хранящий идентификаторы соавторов публикации.

**Events:**

- event\_id:int – основной ключ, идентификатор мероприятия;
- moderator\_id: int – идентификатор модератора, ответственного за новость о мероприятии;
- name: string – название мероприятия;
- photo: string – путь к изображению, соответствующему данному мероприятию;
- description: string – описание мероприятия.

**Comments:**

- comment\_id:int – основной ключ, идентификатор комментария;
- publication\_id: int – идентификатор публикации;
- user\_id:int – основной ключ, идентификатор пользователя;
- text: string – содержание комментария;
- reply\_id:int – основной ключ, идентификатор ответа на комментарий.

## 2.4 Интерфейс

Разрабатываемый веб-сайт является интернет – пространством для участников кластера, где у них есть возможность публиковать свои работы, смотреть работы других участников, комментировать их, добавлять в избранное, взаимодействовать друг с другом. На веб-сайте реализован набор ролей, которые могут получить участники или простые посетители сайта. В зависимости от роли, функционал веб-сайта может отличаться.

На главной странице сайта расположены ссылки, ведущие на основные информационные разделы, такие как раздел «Резиденты», раздел «Афиши» и другие разделы сайта. Далее приведем описание этих разделов.

«Главная» – это основной раздел веб-сайта, на который пользователь попадает после авторизации или регистрации.

«Галерея» – раздел, в котором отображаются опубликованные работы авторов. Любой пользователь, вне зависимости от роли, может перейти на страницу публикации, ознакомиться с подробной информацией, связаться с автором или оставить комментарий.

«Резиденты» – это раздел, в котором указаны заслуженные авторы кластера, а также их популярные работы.

«Профиль» – раздел сайта, в котором пользователь в зависимости от его роли может редактировать личную информацию, просмотреть список понравившихся работ, список своих работ или работ, которые должны пройти модерацию. Также, если пользователь имеет роль «Модератор», в этом разделе отображается список комментариев, на которые поступили жалобы.

Диаграмма структуры разделов и окон веб-сайта продемонстрирована на рисунке 8.

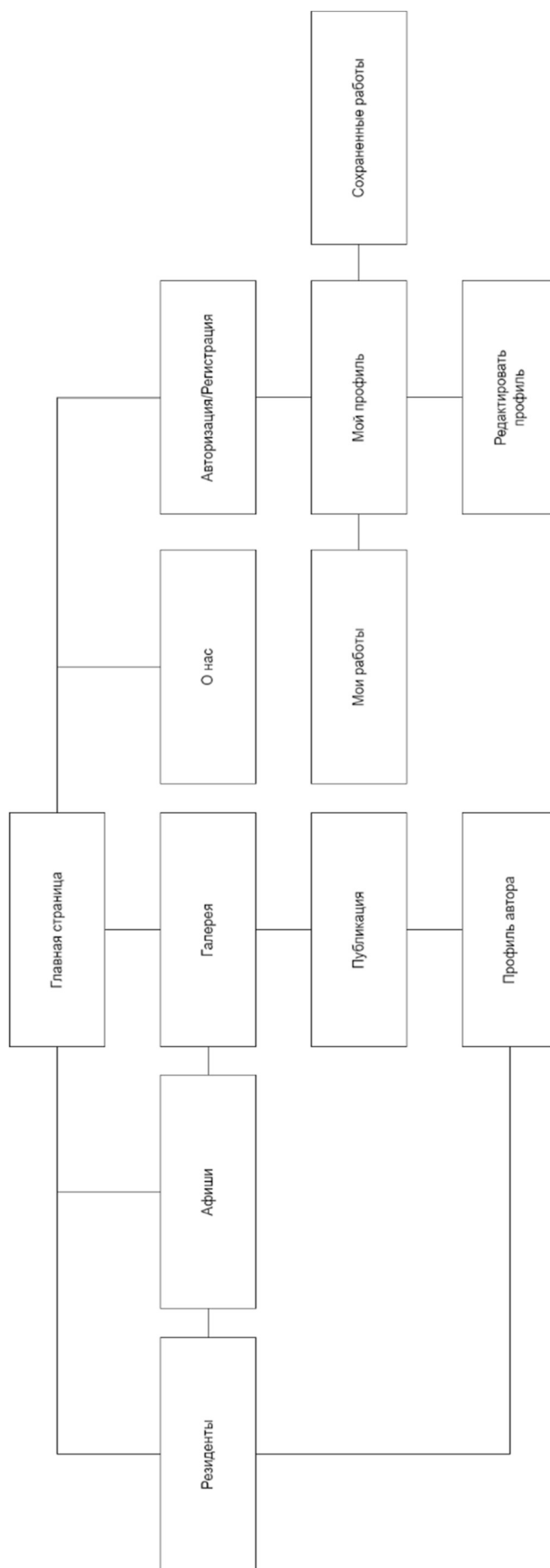


Рисунок 8 – Структура разделов и окон веб-сайта

## 2.5 Структура проекта

Структура проекта веб-сайта изображена на рисунке 9.

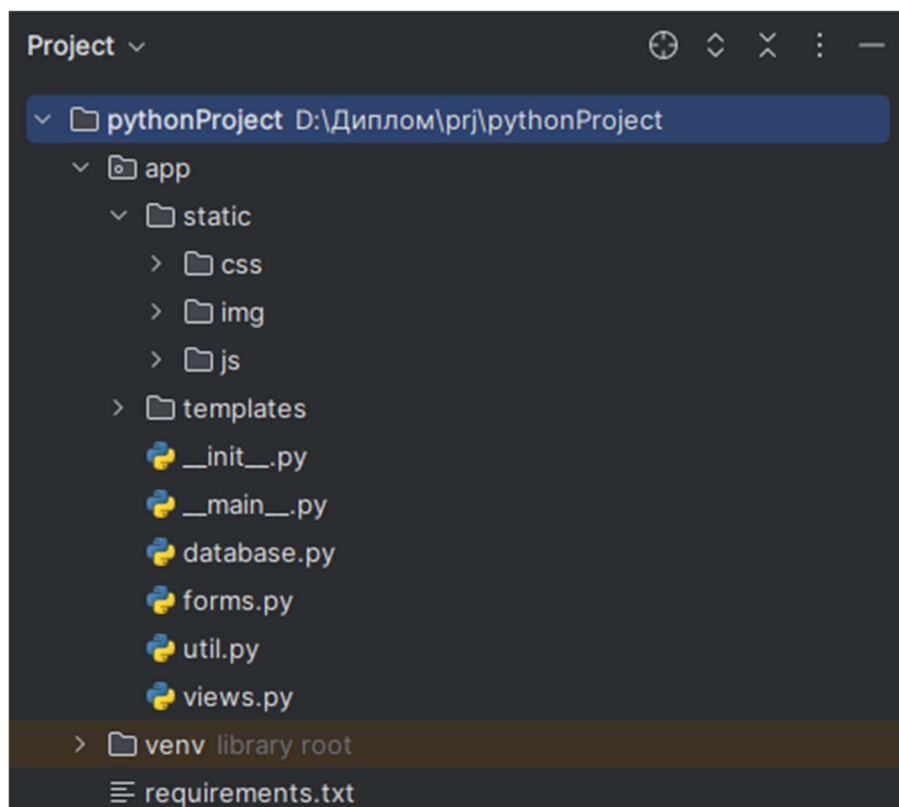


Рисунок 9 – Структура проекта веб-сайта

Опишем структуру данного проекта:

- директория «app» – содержит весь исходный код программы;
- директория «venv» – содержит виртуальное окружение проекта;
- файл «requirements.txt» – в данном файле описаны необходимые для работы модули, которые должны быть установлены в виртуальное окружение;
- директория «app/static» – содержит файлы, используемые для работы сайта и участвующие в отрисовке компонентов в шаблоне;
- директория «app/templates» – содержит шаблоны страниц сайта, используемые шаблонизатором Jinja2;

– файлы «`__init__.py`» и «`__main__.py`» – обязательные файлы для формирования цельного пакета сайта. В файле «`__init__.py`» происходит создание объекта Flask и его настройка;

– файл «`database.py`» представляет собой обертку для взаимодействия с СУБД и заполнения моделей, используемых при отрисовке страниц;

– файл «`forms.py`» содержит в себе описание структур, которые используются при заполнении в файле «`database.py`» и при отрисовке страниц;

– файл «`util.py`» содержит в себе вспомогательные функции, используемые при обработке полученных данных и подготовке к отправке данных. Примером функции из этого файла является проверка сессии на успешность авторизации пользователя;

– файл «`views.py`» содержит в себе функции, используемые для обработки входящих запросов и формирования ответа на них при обращении клиентской части к серверу.

## **2.6 Модуль обработки URL**

Модуль обработки URL в Flask предоставляет механизм привязки URL-адресов к представлениям (`views`) на веб-сайте. Этот модуль использует файл «`views.py`», в котором определена таблица маршрутизации URL, указывающая на представления, соответствующие определенным адресам.

Пример добавления маршрутов приведен в листинге 1.

## Листинг 1 – Файл конфигурации модуля views.py

```
app.add_url_rule('/', 'main_page', main_page)
app.add_url_rule('/gallery', 'gallery_page', gallery_page)
app.add_url_rule('/map', 'maps_page', maps_page)
app.add_url_rule('/posters', 'posters_page', posters_page)
app.add_url_rule('/about', 'about_page', about_page)
app.add_url_rule('/profile', 'profile_page', profile_page)
app.add_url_rule('/profile/<username>', 'profile_page',
profile_page)
app.add_url_rule('/register', 'register_page', register_page)
app.add_url_rule('/login', 'login_page', login_page)
```

Приведем таблицу маршрутизации URL, определяемую данным исходным кодом (Таблица 2).

Таблица 2 – Описание связи маршрутов и функций

Маршрут	Функция	Описание
«/»	main_page	Отображение главной страницы сайта
«/gallery»	gallery_page	Отображение раздела «Галерея»
«/posters»	posters_page	Отображение раздела «Афиши»
«/about»	about_page	Отображение раздела «О нас»
«/profile»	profile_page	Отображение профиля пользователя
«/profile/<username>»	profile_page	Отображение профиля конкретного пользователя, указанного в параметре <username>
«/register»	register_page	Отображение страницы регистрации
«/login»	login_page	Отображение страницы входа

## 2.7 Модуль reCAPTCHA

В целях защиты сайта от DDoS-атак было принято решение использовать систему защиты веб-сайта reCAPTCHA от Google. Причиной выбора данного способа защиты ресурса от атак является популярность данного продукта, а также большой лимит бесплатных запросов (один миллион запросов доступны бесплатно), что оптимально подходит под количество пользователей веб-сайта.

На рисунке 10 представлена временная диаграмма использования сервиса reCAPTCHA. Во время регистрации пользователю необходимо подтвердить то,

что он не робот. Для этого в конце формы регистрации находится форма подтверждения с надписью «Я не робот». Пользователю необходимо нажать на квадратное поле внутри блока. После этого начнется процесс подтверждения.

Пользователю предоставляется два типа подтверждения:

- среди изображений выбрать определенный объект;
- среди частей изображения выбрать те, что относятся к объекту.

После нажатия кнопки подтвердить, сервис reCAPTCHA проверяет ответ пользователя на корректность. В случае неправильного выбора пользователю предоставляются попытки до тех пор, пока он не ответит правильно.

Статус выполнения проверки пользователем записывается в форме. Сервер считывает его перед выполнением действия, запрашиваемым пользователем.

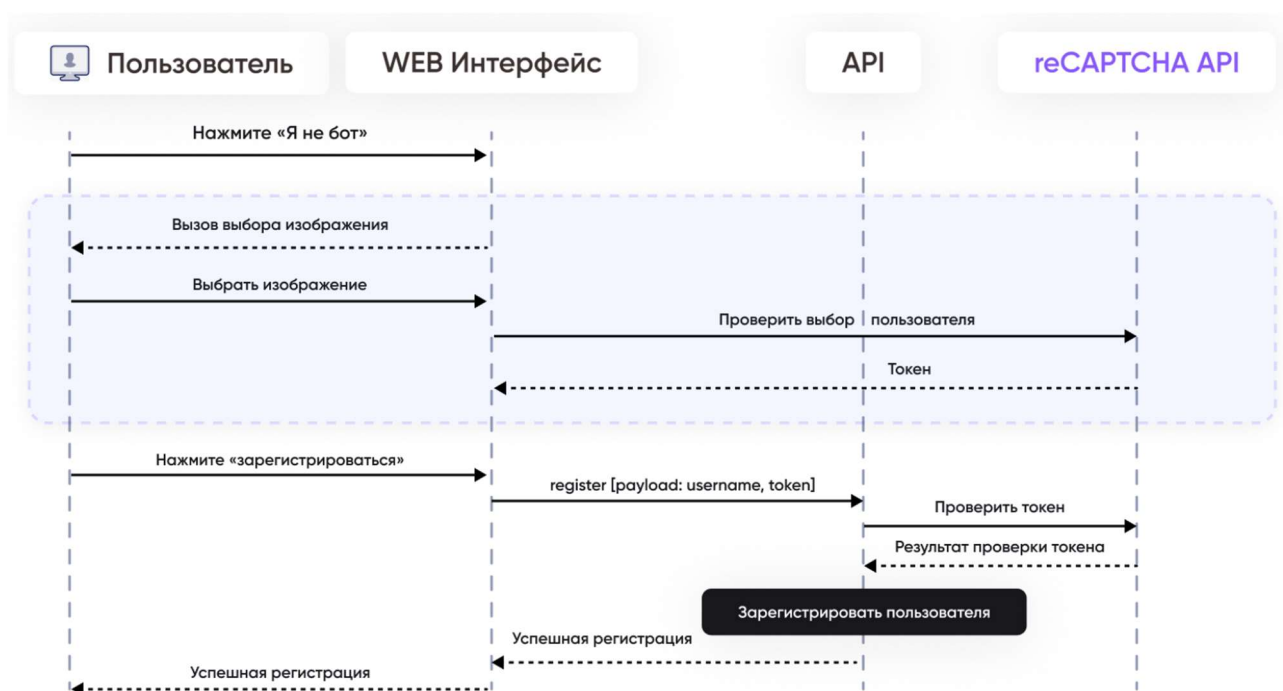


Рисунок 10 – Результат создания домена для веб-сайта

## 2.8 Выводы по главе

В данной главе была спроектирована общая структура веб-сайта, которая содержит в себе программный код, структуру базы данных, описание интерфейса



для всех предусмотренных ролей пользователя, описание использования модуля защиты веб-сайта reCAPTCHA, а также был описан модуль обработки URL. Таким образом, функционал сайта полностью реализован и готов к развёртыванию и тестированию.

## **3 Тестирования и развёртывание**

### **3.1 Развёртывание**

В качестве хостинга для развёртывания веб-сайта креативного кластера был выбран хостинг «Timeweb Cloud».

Развёртывание проведено через при помощи личного кабинета на сайте хостинга. Процесс включает в себя полную настройку серверной инфраструктуры, перенос данных веб-сайта, создание и настройку базы данных, а также настройку окружения.

Для начала нами был создан аккаунт на хостинге Timeweb Cloud. Затем в разделе управления сайтами был выбран и куплен домен для веб-сайта креативного кластера.

На рисунке 11 приведен результат создания домена для веб-сайта.

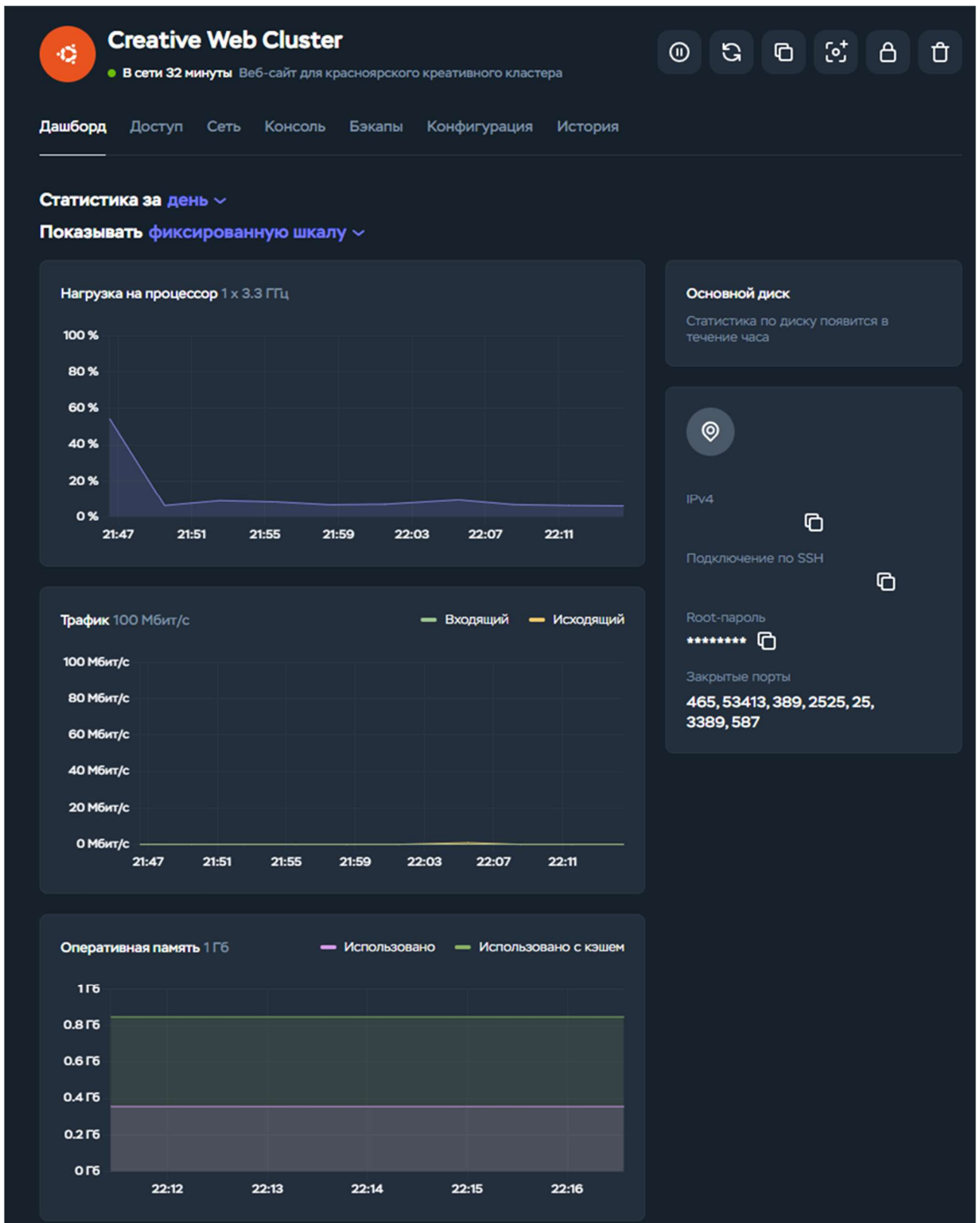


Рисунок 11 – Дашборд, предоставляемый хостингом после создания домена

Далее нами был создан SSH-ключ. Это необходимо для удобного и безопасного подключения к удаленному серверу. После подключения по SSH-

протоколу, были сделаны публичный и приватный ключи. Далее через утилиту «scp» был передан исходный код проекта с локального компьютера на удаленный сервер. Структура каталога с проектом на сервере изображена на рисунке 12.

```
root@3047123-so47386:~# tree -L 3 pythonProject/
pythonProject/
├── app
│   ├── database.py
│   ├── forms.py
│   ├── __init__.py
│   ├── __main__.py
│   ├── static
│   │   ├── css
│   │   ├── img
│   │   └── js
│   └── templates
│       ├── about.html
│       ├── base.html
│       ├── gallery.html
│       ├── gallery_id.html
│       ├── index.html
│       ├── login.html
│       ├── map.html
│       ├── page_404.html
│       ├── posters.html
│       ├── profile.html
│       ├── profile_id.html
│       ├── profile_old.html
│       └── signup.html
├── util.py
├── views.py
├── requirements.txt
└── venv
```

Рисунок 12 – Структура проекта

После того, как проект был передан на удаленный сервер, были выполнены следующие команды (Рисунок 13):

```
1. apt install mongodb
2. systemctl start mongodb
3. systemctl status mongodb
4. cd pythonProject/
5. rm -rf venv
6. apt install python3.8-venv
7. python -m venv venv
8. source venv/bin/activate
9. pip install -r requirements.txt
10. uwsgi -shared-socket :443 -https =0,server.crt,server.key -w app:app
```

### Рисунок 13 – Результат создания домена для веб-сайта

Описание выполненных команд в соответствии с рисунком 13:

- 1) установка MongoDB на сервер при помощи пакетного менеджера;
- 2) запуск демона MongoDB;
- 3) проверка успешного запуска MongoDB;
- 4) переход в папку с проектом;
- 5) удаление виртуального окружения, использовавшегося на локальном компьютере;
- 6) установка пакета для настройки окружения Python;
- 7) создание виртуального окружения Python;
- 8) использование виртуального окружения;
- 9) установка зависимостей проекта в локальное окружение;
- 10) запуск сервера с ключами сертификата.

В результате проведенных мероприятий, веб-сайт был успешно развернут на хостинге «Timeweb Cloud» и теперь доступен по следующему URL-адресу: <https://147.45.158.247:5000/>.

На рисунке 15 изображен результат развертывания веб-сайта.

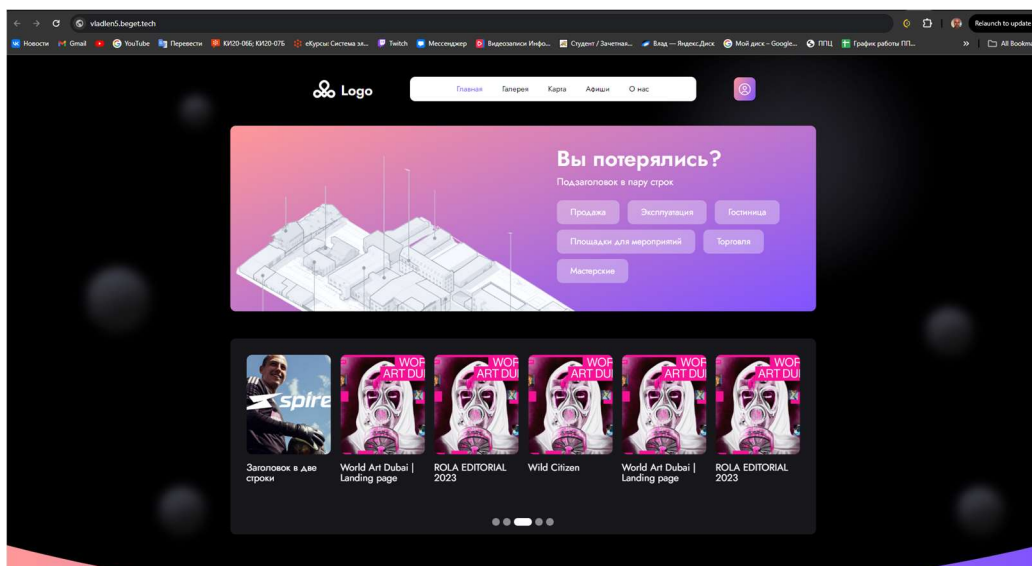


Рисунок 14 – Результат развертывания веб-сайта

## 3.2 Методы тестирования

Чтобы убедиться, что сайт работает исправно, необходимо провести ручное тестирование с различными вариантами использования. На сайте реализован функционал для трех типов пользователей: обычный пользователь, автор и модератор. Обычные пользователи могут просматривать публикации в разделе «Галерея», писать комментарии под публикациями и изменять свой профиль. Авторы помимо действий, доступных обычным пользователям, могут размещать, изменять и удалять свои работы. Модераторы могут делать все вышеперечисленное, а также модерировать и удалять комментарии и публикации.

Для ручного тестирования веб-сайта, был разработан набор тестов, который охватывает весь функционал сайта с учетом пользовательских ролей. Далее опишем набор ручных тестов для разрабатываемого веб-сайта.

### **Тестирование регистрации и аутентификации:**

- попытка зарегистрировать нового пользователя с корректными данными;
- попытка зарегистрировать пользователя с неправильными данными (например, некорректным адресом электронной почты);

- попытка аутентификации с корректными учетными данными;
- попытка аутентификации с неправильными учетными данными.

#### **Тестирование редактирования профиля:**

- изменение данных профиля пользователя;
- попытка изменить данные профиля на недопустимые (например, сменить пароль на слишком короткий).

#### **Тестирование просмотра и взаимодействия с контентом:**

- просмотр галереи с публикациями;
- комментирование публикации;
- просмотр разделов «Афиши» и «О нас»;
- просмотр списка сохраненных публикаций в профиле.

#### **Тестирование функционала автора:**

- загрузка новой публикации;
- редактирование существующей публикации;
- попытка удаления публикации.

#### **Тестирование функционала модератора:**

- просмотр списка публикаций для модерации;
- просмотр жалоб на комментарии;
- модерация публикаций (т.е. утверждение или отклонение работ).

#### **Тестирование взаимодействия между пользователями:**

- комментирование публикаций другими пользователями;
- попытка пожаловаться на комментарий.

### **Тестирование общего поведения веб-сайта:**

- проверка корректности работы переходов между страницами;
- проверка работы функциональных элементов, таких как кнопки, ссылки и формы.
- проверка адаптивности и отзывчивости интерфейса на различных устройствах и разрешениях экрана;

Далее в таблице 3 будут приведены результаты тестирования для всех возможных сценариев использования веб-сайта.

Таблица 3 – Результаты тестирования

<b>Название теста</b>	<b>Статус</b>	<b>Расположение результатов</b>
Авторизация	Успешно	Приложение А, Рисунок А.1
Регистрация	Успешно	Приложение А, Рисунок А.2
Редактирование профиля	Успешно	Приложение А, Рисунок А.3
Просмотр галереи с публикациями	Успешно	Приложение А, Рисунок А.4
Просмотр раздела «Афиши»	Успешно	Приложение А, Рисунок А.5
Просмотр раздела «О нас»	Успешно	Приложение А, Рисунок А.6
Просмотр списка сохраненных публикаций	Успешно	Приложение А, Рисунок А.7
Выгрузка новой публикации	Успешно	Приложение А, Рисунок А.8
Попытка удалить публикацию	Успешно	Приложение А, Рисунок А.9
Просмотр жалоб на комментарии	Успешно	Приложение А, Рисунок А.10
Отображение комментариев	Успешно	Приложение А, Рисунок А.11

### **3.3 Выводы по главе**

В данной главе было проведено успешное развертывание сайта на хостинге Timeweb Cloud и проведено ручное тестирование с учетом всевозможных вариантов использования веб-сайта.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения данной ВКР был разработан полностью функциональный веб-сайт для креативного Красноярского кластера.

Все поставленные задачи были выполнены в полном объёме. Проведен анализ аналогичных решений, выявлены основные критерии для проектирования веб-сайта.

Реализованы весь функционал, включая интерфейс для пользователей с учетом их ролей, модальные окна, базу данных, клиент-серверную архитектуру и программный код.

Разработанный сайт имеет весь необходимый функционал для привлечения новой аудитории к креативному кластеру, авторов, о которых раньше было неизвестно, а также инвесторов, которые смогут поддерживать деятельность авторов. На веб-сайте пользователи смогут делиться новыми идеями, создавать совместные работы и находить вдохновение.

Выбранные инструменты для разработки позволят беспрепятственно расширять функционал сайта, что планируется в будущем.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СТУ 7.5-07-2021. Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности.
2. Create At Home: сайт. – URL: <https://www.creativitycluster.eu/> (дата обращения: 05.01.24)
3. Awwwards: сайт. – URL: <https://www.awwwards.com/> (дата обращения: 05.01.24)
4. Дизайн завод: сайт. – URL: <https://design-zavod.ru/> (дата обращения: 05.01.24)
5. Timeweb Cloud: сайт. – URL: <https://timeweb.cloud/> (дата обращения: 21.03.24)
6. MongoDB: сайт. – URL: <https://www.mongodb.com/> (дата обращения: 22.03.24)
7. Flask: сайт. – URL: <https://flask.palletsprojects.com/en/3.0.x/> (дата обращения: 25.04.24)
8. Discuss.python: сайт. – URL: <https://discuss.python.org/> (дата обращения: 25.04.24)
9. StackExchange: сайт. – URL: <https://askubuntu.com/> (дата обращения: 25.04.24)
10. Devhops: сайт. – URL: <https://devhops.ru/> (дата обращения: 25.04.24)

# ПРИЛОЖЕНИЕ А

## Результаты тестирования

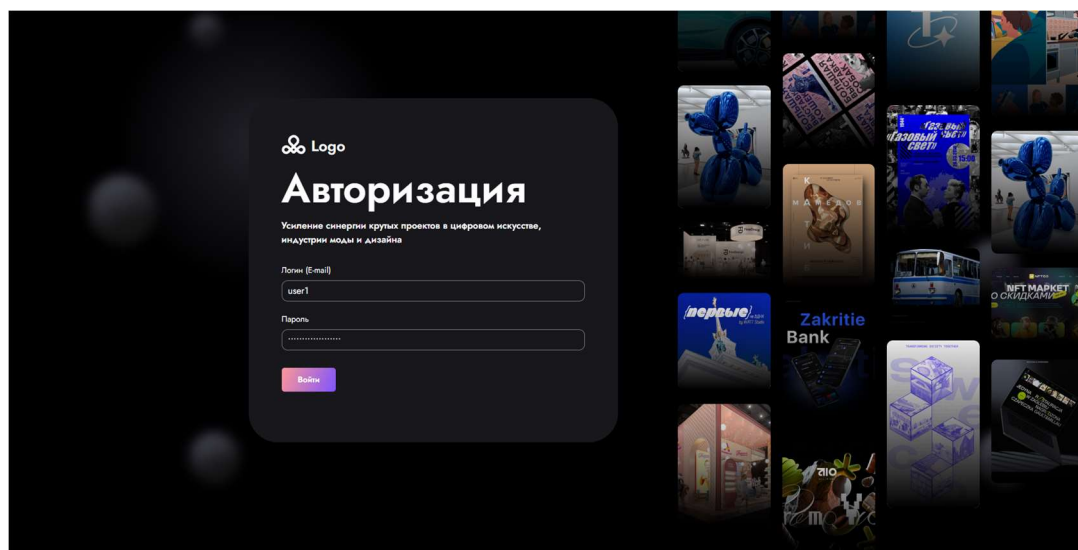


Рисунок А.1 – Результат теста авторизации

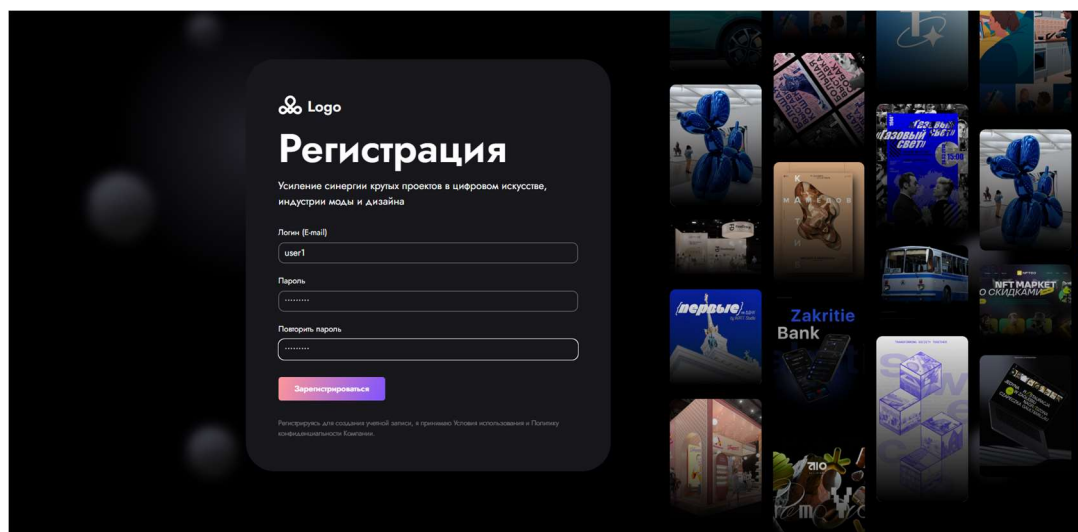


Рисунок А.2 – Результат теста регистрации

## Продолжение приложения А

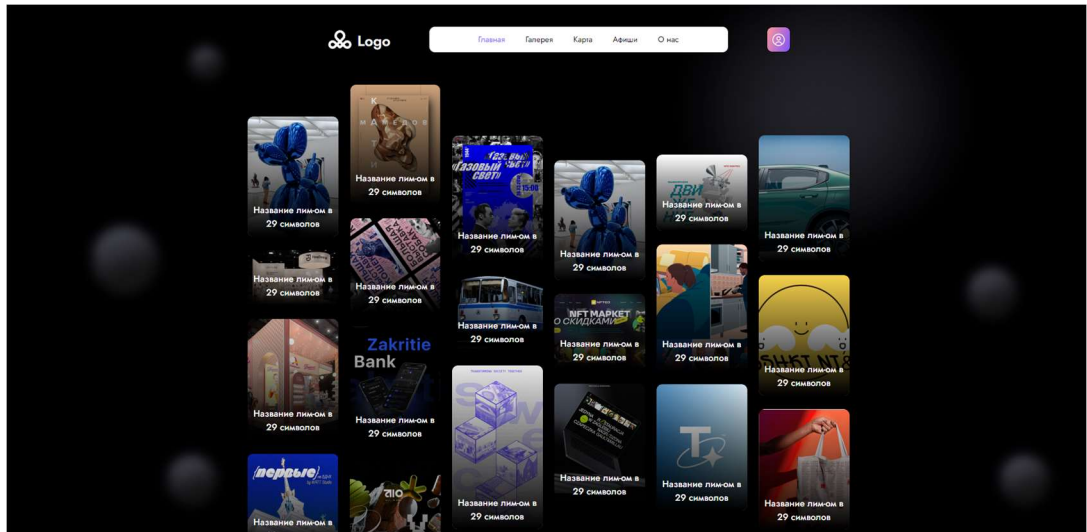


Рисунок А.3 – Результат теста на отображение галереи

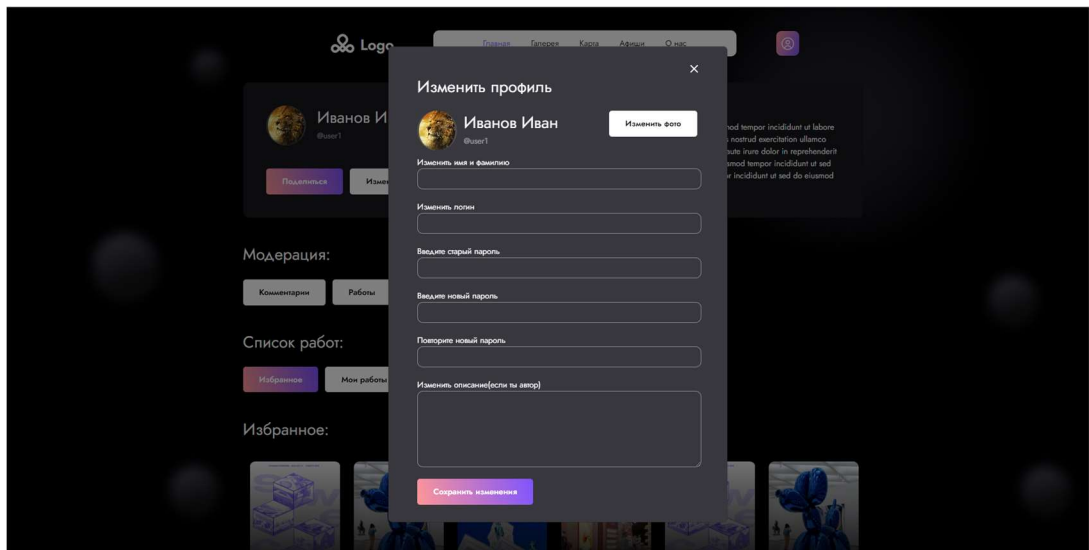


Рисунок А.4 – Результат теста редактирования профиля

## Продолжение приложения А

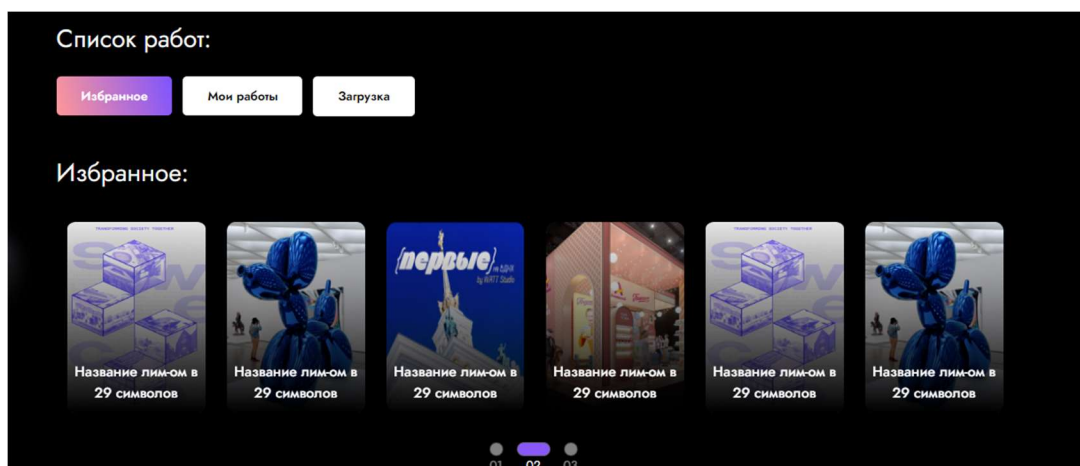


Рисунок А.5 – Результат отображения сохраненных публикаций

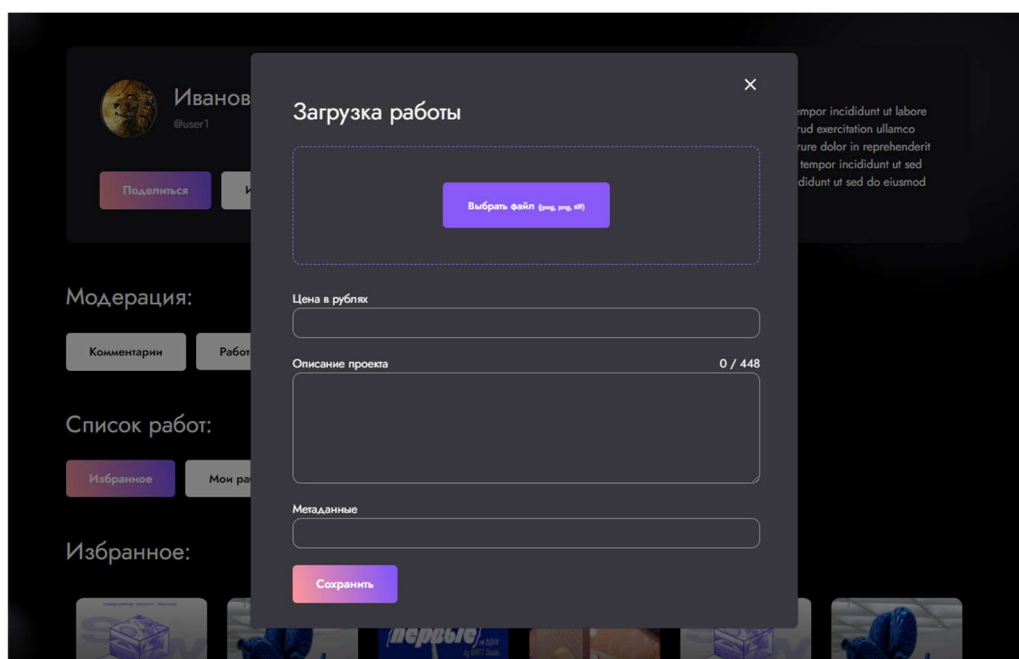


Рисунок А.6 – Результат выгрузки публикации

## Продолжение приложения А

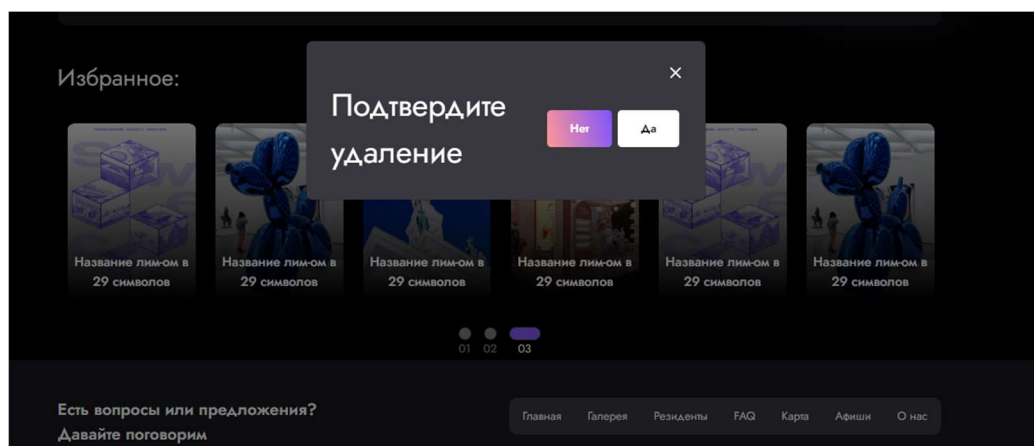


Рисунок А.7 – Результат теста на удаление публикации

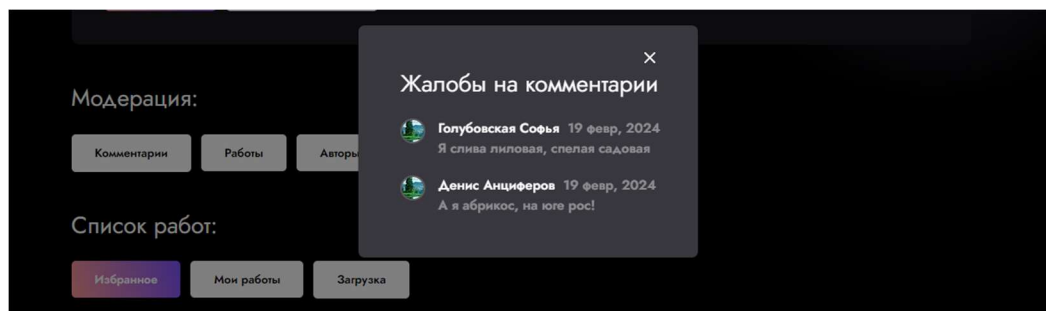


Рисунок А.8 – Результат теста на отображение жалоб на комментарии

## Продолжение приложения А

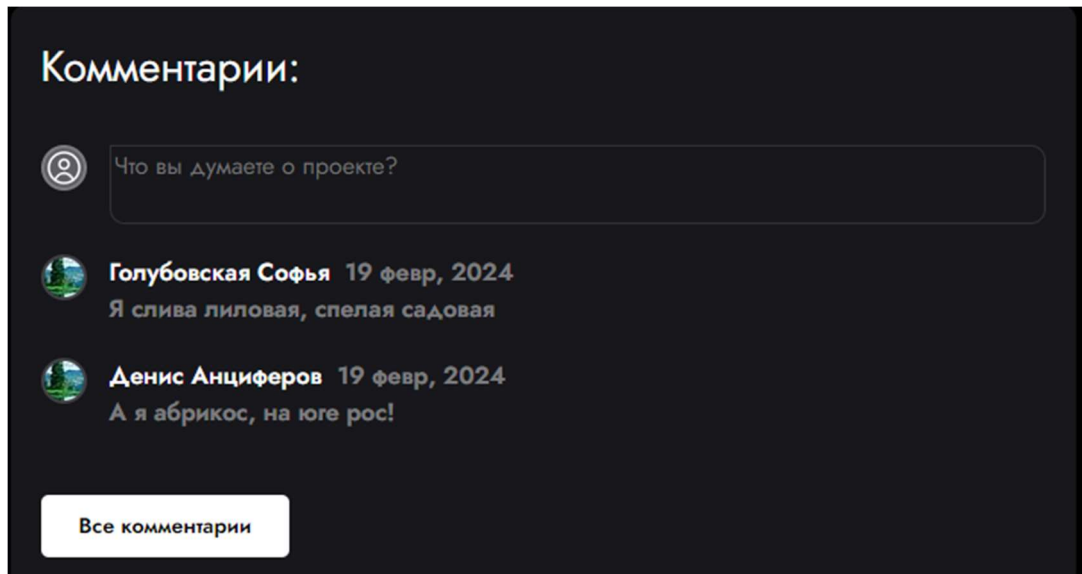


Рисунок А.9 – Результат теста на регистрацию

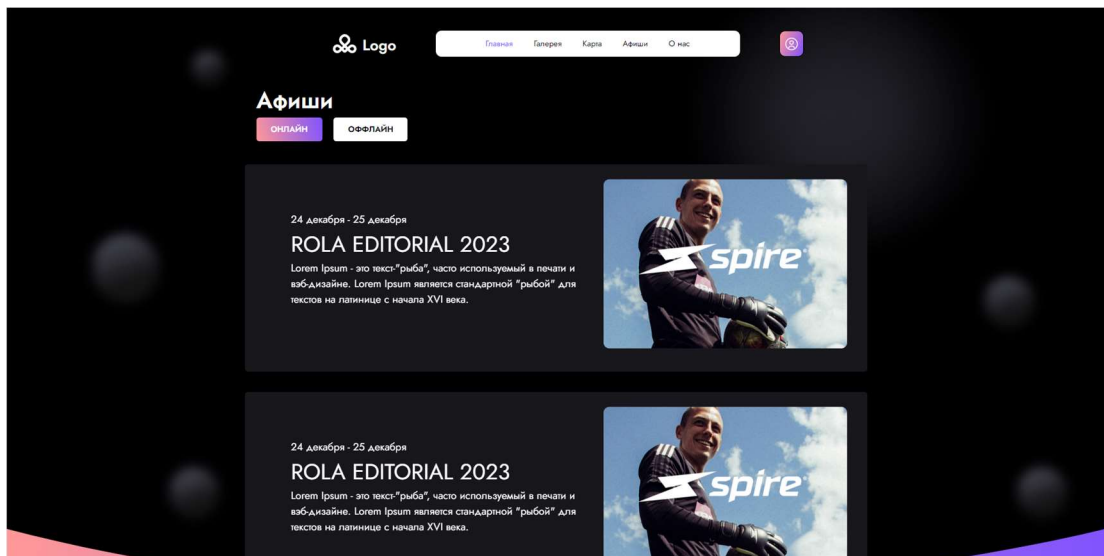


Рисунок А.10 – Результат теста отображения раздела «Афиши»

## Окончание приложения А

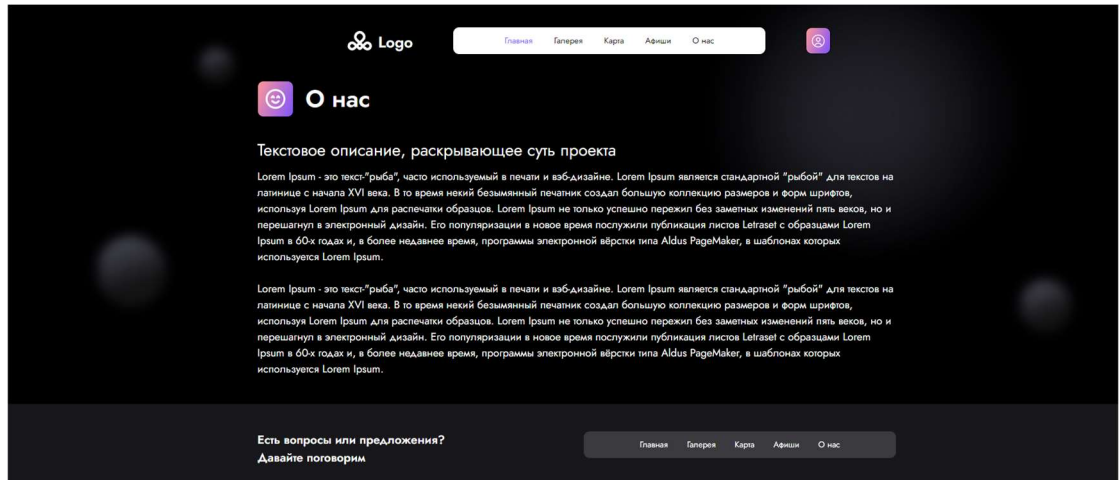


Рисунок А.11 – Результат отображения раздела «О нас»



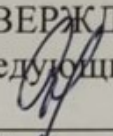
Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

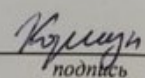
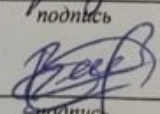
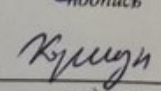
  
О.В. Непомнящий

«13» 06 2024 г.

**БАКАЛАВРСКАЯ РАБОТА**

090301 Информатика и вычислительная техника

Веб-сайт креативного кластера

Руководитель	 подпись	13.06.2024 дата	доцент, канд. физ.-мат. наук должность, ученая степень	К.В. Коршун
Выпускник	 подпись	13.06.2024 дата		В.О. Леушкин
Нормоконтролёр	 подпись	13.06.2024 дата		К.В. Коршун

Красноярск 2024