

**Министерство науки и высшего образования Российской
Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**КАФЕДРА АНАЛИЗА ДАННЫХ И ТЕХНОЛОГИЙ
ПРОГРАММИРОВАНИЯ**

Направление: 09.03.03 – «Прикладная информатика»

**КУРСОВАЯ РАБОТА
РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ УЧЁТА И АНАЛИЗА ПРОЕКТОВ
КАЗАНСКОГО (ПРИВОЛЖСКОГО) ФЕДЕРАЛЬНОГО
УНИВЕРСИТЕТА**

Студент __ курса

группы _____

«__» _____ 2025 г. _____ Багыманов Р.М.

Научный руководитель

к.т.н., доцент КАДиТП

«__» _____ 2025 г. _____ Бурнашев Р.А.

Казань-2025

Содержание

Введение	2
1. Анализ Предметной области	4
1.1. Особенности предметной области	5
1.2. Обзор существующих решений	6
1.3. Формирование технического задания.	9
2. Проектирование приложения	10
2.1. Проектирование архитектуры приложения	11
2.2. Проектирование сущностей базы данных.	12
2.3. Проектирование архитектуры REST API	14
2.4. Определение стека технологий и инструментов разработки проекта	16
3. Разработка веб-приложения учёта и анализа проектов	18
3.1. Создание архитектуры веб-платформы	19
3.2. Интеграция PostgreSQL базы данных	20
3.3. Реализация пользовательского интерфейса веб-приложения	23
4. Тестирование веб-приложения учета и анализа проектов	36
4.1. Тестирование функциональной части приложения	37
4.2. Тестирование пользовательского интерфейса	42
Заключение	45
Список использованных источников	47

Введение

В современных условиях цифровизации образования особую актуальность приобретают задачи систематизации и визуализации научно-исследовательской деятельности студентов высших учебных заведений. Казанский (Приволжский) федеральный университет ежегодно проводит значительное количество научных исследований силами студентов, магистрантов и аспирантов. Однако существующие подходы к представлению результатов научной работы не обеспечивают должного уровня систематизации и доступности информации о проводимых исследованиях.

Отсутствие единой платформы для демонстрации научных достижений студентов создает ряд проблем: затрудняется поиск и анализ тематически близких работ, ограничивается возможность академического взаимодействия между исследователями, усложняется процесс мониторинга научной активности. Традиционные методы представления научных работ в виде статичных документов не позволяют в полной мере раскрыть содержание и значимость проводимых исследований для широкой аудитории.

Применение современных информационных технологий, включая методы машинного обучения и нейронных сетей, открывает новые возможности для создания инновационных решений в области управления научными проектами. Автоматизация процессов анализа контента и генерации визуального представления исследований позволяет значительно повысить эффективность работы с научной информацией и улучшить пользовательский опыт взаимодействия с системой.

Целью данной работы является разработка и реализация веб-приложения для учёта, систематизации и визуального представления научно-исследовательских проектов студентов, магистрантов и аспирантов Казанского (Приволжского) федерального университета с использованием

технологий машинного обучения для автоматической генерации визуального контента.

В рамках достижения цели необходимо выполнить следующие задачи:

1. Исследовать предметную область и рассмотреть доступные решения — провести исследование современных подходов к управлению научными проектами в образовательных учреждениях, изучить функциональные возможности аналогичных систем и выявить их ограничения.

2. Проектирование архитектуры системы — разработать техническую архитектуру веб-приложения, построить схему базы данных, создать пользовательские интерфейсы и API для коммуникации элементов системы.

3. Внедрение системы входа и управления доступом — сформировать защитный модуль, обеспечивающий ограниченный вход в систему для владельцев университетских учетных записей (@stud.kpfu.ru).

4. Разработка функциональности управления проектами с использованием машинного обучения — реализовать инструменты для добавления, редактирования и категоризации научных работ, создать систему поиска и фильтрации проектов по различным критериям.

5. Тестирование и оптимизация — провести комплексное тестирование разработанного приложения, оценить его производительность и внести необходимые корректировки для обеспечения стабильной работы системы.

Решение поставленных задач позволит создать современную цифровую платформу, способствующую развитию научно-исследовательской деятельности в университете и повышению качества представления результатов студенческих исследований.

1. Анализ Предметной области

1.1. Особенности предметной области

Разработка веб-приложения для учёта и анализа проектов Казанского (Приволжского) федерального университета представляет собой комплексную задачу, требующую учета специфических особенностей академической среды и технологических требований современного образовательного процесса. Предметная область данного исследования характеризуется рядом ключевых аспектов, которые определяют архитектурные решения и функциональные возможности разрабатываемой системы.

Выделим некоторые критически важные особенности для предметной области:

1. Безопасность и контролируемый доступ. Критически важной особенностью предметной области является необходимость обеспечения безопасного доступа к научной информации через корпоративную систему аутентификации. Использование домена @stud.kpfu.ru не только гарантирует принадлежность пользователей к академическому сообществу университета, но и создает доверительную среду для обмена научными идеями. Это исключает возможность несанкционированного доступа внешних пользователей и поддерживает высокие стандарты академической этики при работе с интеллектуальной собственностью.

2. Автоматизация визуализации научного контента. Ключевой технологической особенностью является интеграция методов машинного обучения для автоматического анализа текстового содержания научных работ и генерации соответствующих визуальных образов. Применение нейронных сетей для создания тематических изображений представляет инновационный подход к презентации научной информации, позволяющий преобразовать сложные теоретические концепции в наглядные графические представления.

Такой подход значительно повышает привлекательность научных материалов для широкой аудитории и облегчает процесс первичного ознакомления с содержанием работ.

3. Интуитивность пользовательского интерфейса и навигации. Разрабатываемая система должна обеспечивать максимально простое и понятное взаимодействие с пользователем. Главная страница в формате витрины проектов должна позволять быстрое ознакомление с научными работами через визуальные элементы, а не только через текстовые описания. Личный кабинет пользователя должен предоставлять централизованный доступ ко всем функциям управления проектами, а форма добавления новых работ — обеспечивать простой процесс публикации материалов без необходимости изучения сложных инструкций.

4. Гибкость в представлении разнородного научного контента. Предметная область характеризуется значительным разнообразием типов научных работ — от технических разработок до гуманитарных исследований. Система должна обеспечивать универсальные механизмы обработки различных форматов документов и адаптивное извлечение ключевой информации из текстов различной структуры и тематической направленности. Это требует создания алгоритмов, способных эффективно работать с междисциплинарными исследованиями и проектами на стыке различных научных областей.

5. Система поиска и фильтрации контента. Учитывая потенциально большое количество научных работ в системе, критически важной является реализация эффективных механизмов поиска и категоризации проектов. Пользователи должны иметь возможность быстро находить тематически близкие работы, фильтровать результаты по различным критериям.

1.2. Обзор существующих решений

Была проведена работа по анализу уже существующих витрин научных работ в других университетах. По результату поиска было найдено подходящее по функционалу и требованиям веб-приложение Уральского Федерального университета [1].

УрФУ реализовал на своем сайте витрину лучших проектов, которая содержит в себе работы студентов учебной и внеучебной деятельности (рисунок 1.1). Из преимуществ данного веб-приложения могу выделить:

1. Наличие поисковой строки, а также фильтров по времени, уровню образования и образовательной программе.
2. В визитке проекта указывается вся команда и наименование заказчика (рисунок 1.2).
3. Реализована обратная связь в виде отзывов о данном проекте.

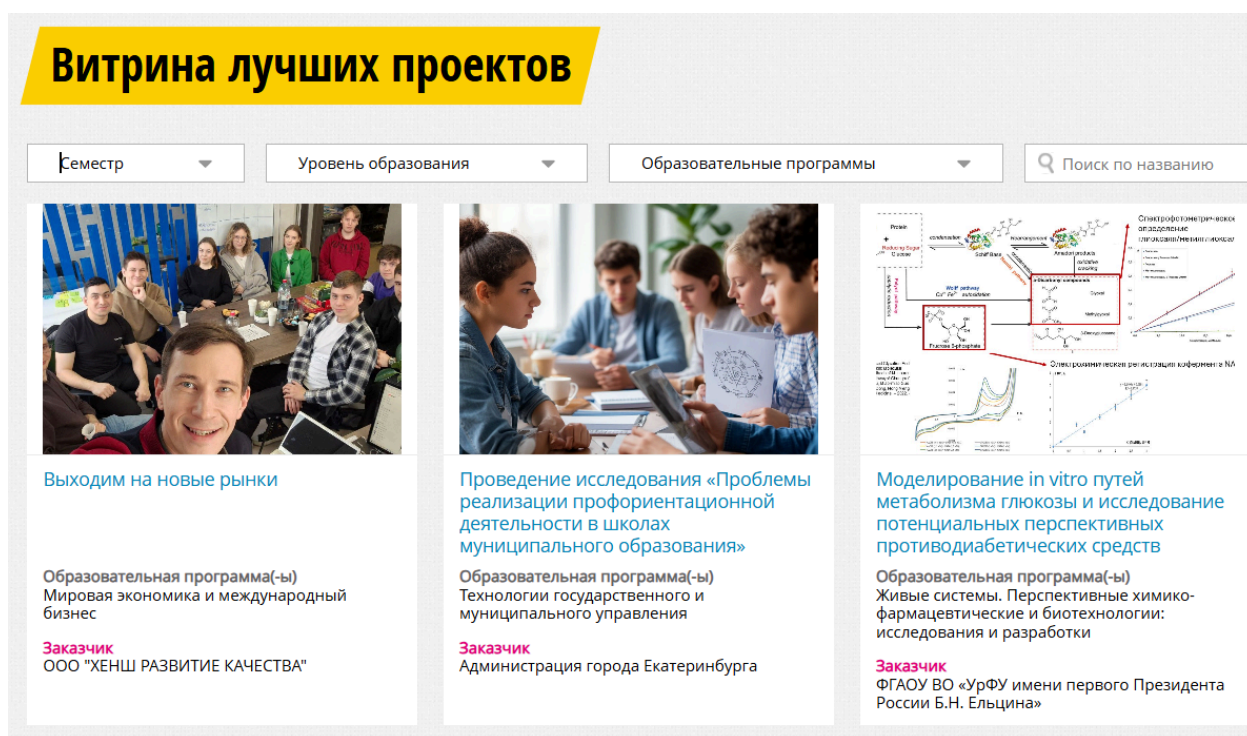


Рисунок 1.1. Основное окно витрины УрФУ.

Несмотря на такие значимые и важные преимущества реализации, были найдены некоторые аспекты, которых может не хватать обычным пользователям:

1. Отсутствие контактов или информации для обратной связи
2. Пользователям может не хватать функции добавления проекта в избранное для ускорения и облегчения поиска нужной им работы.
3. Нехватка подробной информации о проекте и итоговом результате. Пользователь, заинтересовавшись данным проектом, не сможет получить достаточно информации о ходе работы, а также о результате в виде документа, который можно скачать на своё устройство.
4. На данной платформе отсутствует возможность отправлять студентам заявку на добавление своих проектов. То есть данная возможность имеется только у администрации университета, а это значит, что и проекты будут добавляться только те, которые интересны исключительно университету.

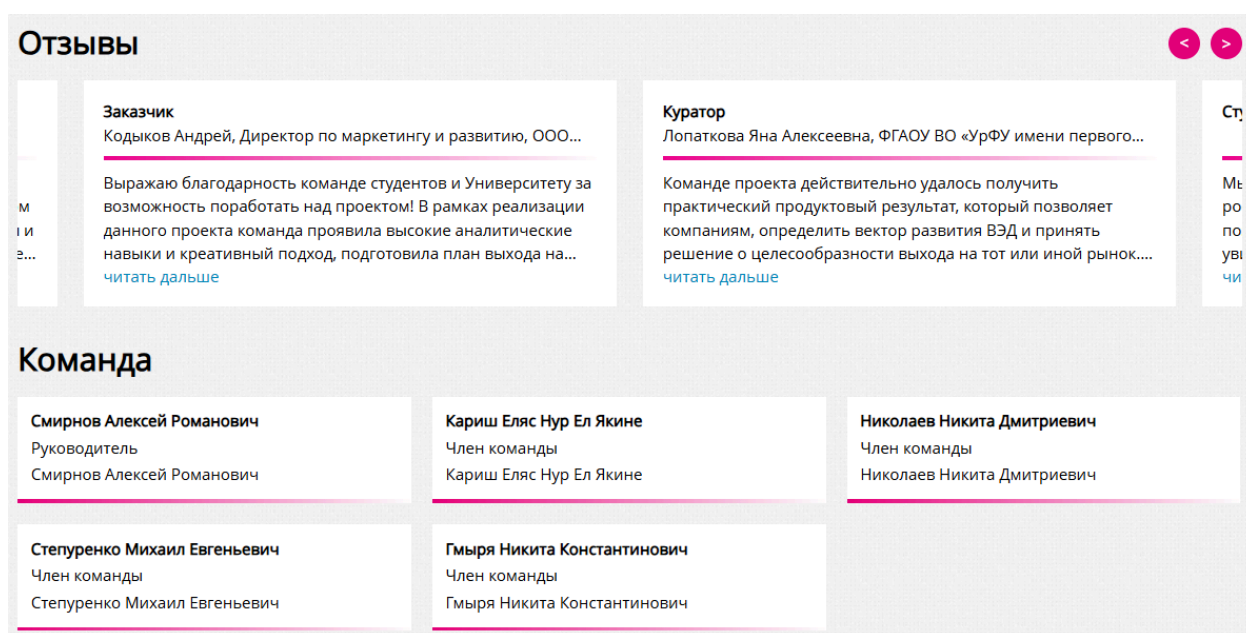


Рисунок 1.2. Указание участников команды.

На основе анализа существующих витрин научных работ в других университетах можно выделить несколько общих недостатков, характерных для большинства подобных платформ:

1. Ограниченная интерактивность и обратная связь. Многие витрины представляют собой статичные каталоги без возможности

полноценного взаимодействия между пользователями и авторами проектов. При разработке собственной платформы стоит учесть этот момент и реализовать систему лайков и поддержку связи с разработчиками проектов, чтобы создать живое сообщество вокруг научных работ.

2. Отсутствие баланса между информативностью и удобством восприятия. В рассмотренной витрине УрФУ наблюдается недостаток детальной информации о проектах, в то время как другие платформы могут перегружать пользователя излишними техническими подробностями.

1.3. Формирование технического задания.

Техническое задание определяет основные требования к разрабатываемому веб-приложению для систематизации научно-исследовательских работ студентов КФУ. Правильное формулирование технических требований является критически важным этапом, обеспечивающим успешную разработку и внедрение информационной системы.

Функциональные требования

- авторизация исключительно через корпоративные учетные записи домена @stud.kpfu.ru;
- автоматическое определение статуса пользователя (студент, магистрант, аспирант) на основе учетной записи;
- добавление новых научных работ через веб-форму с полями: название, аннотация, область исследования, ключевые слова, статус работы;
- загрузка файлов научных работ в форматах PDF, DOC, DOCX;
- редактирование и удаление собственных работ пользователем;

- отображение всех научных работ на главной странице в виде интерактивных карточек;
- полнотекстовый поиск по названию, аннотации, ключевым словам и автору;
- фильтрация работ по типу работы и факультету;
- детальный просмотр информации о работе с возможностью скачивания файла;
- отображение и редактирование профиля пользователя в личном кабинете;
- управление списком собственных научных работ в личном кабинете;
- просмотр статистики: количество работ, общее число просмотров, популярные работы.

Нефункциональные требования

- время отклика системы на стандартные запросы не более 3 секунд;
- время генерации изображения нейронной сетью не более 30 секунд;
- поддержка одновременной работы до 500 пользователей;
- шифрование всех данных при передаче по протоколу HTTP;
- поддержка современных веб-браузеров (Chrome, Firefox, Safari, Edge);

- адаптивный дизайн для корректного отображения на мобильных устройствах;
- интуитивно понятный пользовательский интерфейс;
- поддержка хранения до 100000 научных работ.

2. Проектирование приложения

2.1. Проектирование архитектуры приложения

Имея при себе готовое техническое задание можно начать процесс проектирования нашей работы. Так как проект представляет из себя веб-приложение, то самым удачным вариантом окажется использование клиент-серверной архитектуры, [2] поскольку она разделяет основную логику на две части: клиент (графическое представление сайта и витрины) и сервер (работа с базой данных и API). Данная структура приложения позволяет придерживаться идеологии чистого кода, которая будет легко масштабироваться, тестироваться и в дальнейшем мониториться. Так как данная архитектура разделяет весь проект на отдельные части, то и отлавливание ошибок не будет из себя представлять что-то трудозатратное. И если мы говорим о клиент-серверной части, то всю логику можно разбить на 4 части:

1. Client (Клиент). Представляет из себя визуальную составляющую веб-приложения, то есть HTML-страницы, которые, благодаря стилям CSS и скриптам JavaScript, удовлетворяют потребность пользователей в удобном и дружелюбном интерфейсе. Все действия пользователей, модерации и администрирования проходят именно на стороне клиента.

2. Database (База данных). Базы данных являются неотъемлемой частью любого веб-приложения, особенно если оно заточено на работу с большими потоками информации в виде файлов. В моем проекте была использована реляционная база данных PostgreSQL [3], которая славится своей надежностью и простотой миграции.

3. API (интерфейс программного приложения). Содержит описание всех путей взаимодействия, которые соединяют клиентскую часть с системой хранения данных. API функционирует как соединительный элемент: принимает обращения пользователей, производит их обработку по заданным

алгоритмам работы системы и возвращает требуемую информацию. Благодаря RESTful [4] архитектуре API становится интуитивно понятным для разработчиков и легко масштабируемым при добавлении новых функций. Все операции CRUD [5] (создание, чтение, обновление, удаление) с данными проходят через API, что обеспечивает единообразие обработки запросов и централизованный контроль над доступом к информации.

4. Server (Сервер). Серверная часть представляет собой основу всего веб-приложения, которая объединяет в себе API и логику работы с базой данных. Сервер отвечает за обработку входящих HTTP-запросов, аутентификацию пользователей, валидацию данных и выполнение бизнес-логики приложения. Именно на сервере происходит основная вычислительная работа: обработка файлов, формирование ответов для клиента и поддержание сессий пользователей. Серверная часть веб-приложения была построена так, что можно было обеспечить отказоустойчивость, быстроедействие и надежность. Помимо этого, также появляется возможность обеспечить горизонтальное масштабирование [6] при повышении нагрузки на проект.

2.2. Проектирование сущностей базы данных.

Создание цифровой экосистемы для демонстрации студенческих исследовательских проектов требует комплексного подхода к организации информационной архитектуры. Платформа должна обеспечивать:

- авторизацию обучающихся посредством корпоративного домена университета (@stud.kpfu.ru);
- размещение исследовательских материалов с интеллектуальной визуализацией содержания;

- контроль качества публикуемого контента через многоуровневую систему рецензирования;
- социальное взаимодействие участников академического сообщества;
- мониторинг активности и формирование аналитических отчетов.

Архитектурное решение базируется на шести таблицах (рисунок 2.1):

Таблица “Студенты” (Students) - Профили участников академического сообщества. Базовый элемент платформы, аккумулирующий персональные характеристики пользователей. Включает идентификационные данные, академическую принадлежность (институт, образовательная программа) и криптографические параметры доступа. Система поддерживает дифференциацию по образовательным циклам: первая ступень высшего образования, магистерская подготовка, научно-исследовательская деятельность аспирантов.

Таблица “Работы” (Works) - Репозиторий исследовательских материалов. Ключевой контентный модуль системы. Агрегирует описательную информацию проектов (наименование, аннотация, тематическая область), авторские связи, машинно-генерируемые графические представления, метрики вовлеченности аудитории и файловые ресурсы. Обеспечивает полнотекстовый поиск и тематическую категоризацию материалов.

Таблица “Представления работы” (Work_Submissions) - Буферная зона предварительной обработки. Транзитный компонент для реализации процедур валидации контента. Содержит неопубликованные материалы на стадии экспертной оценки, статусную информацию о ходе рассмотрения и

экспертные заключения. Успешно верифицированные работы мигрируют в основной репозиторий.

Таблица “Модераторы” (Moderators) - Подсистема распределения административных привилегий. Регулирует права доступа к функциям контент-менеджмента. Фиксирует пользователей с экспертными полномочиями, определяет границы их компетенций (валидация материалов, отклонение заявок, доступ к аналитическим панелям) и поддерживает иерархическую структуру назначений.

Таблица “Лайки работы” (Work_Likes) - Механизм социальной оценки контента. Реляционная структура для учета пользовательских предпочтений. Реализует связь типа "многие-ко-многим" между участниками и исследовательскими работами, исключая возможность множественного голосования от одного аккаунта.

Таблица “Логи модерации” (Moderation_Logs) - Система аудита экспертной деятельности. Хронологический реестр модераторских решений. Документирует все действия по рассмотрению заявок с указанием временных рамок, обоснований и ответственных лиц, что гарантирует прозрачность и подотчетность процесса контроля качества.

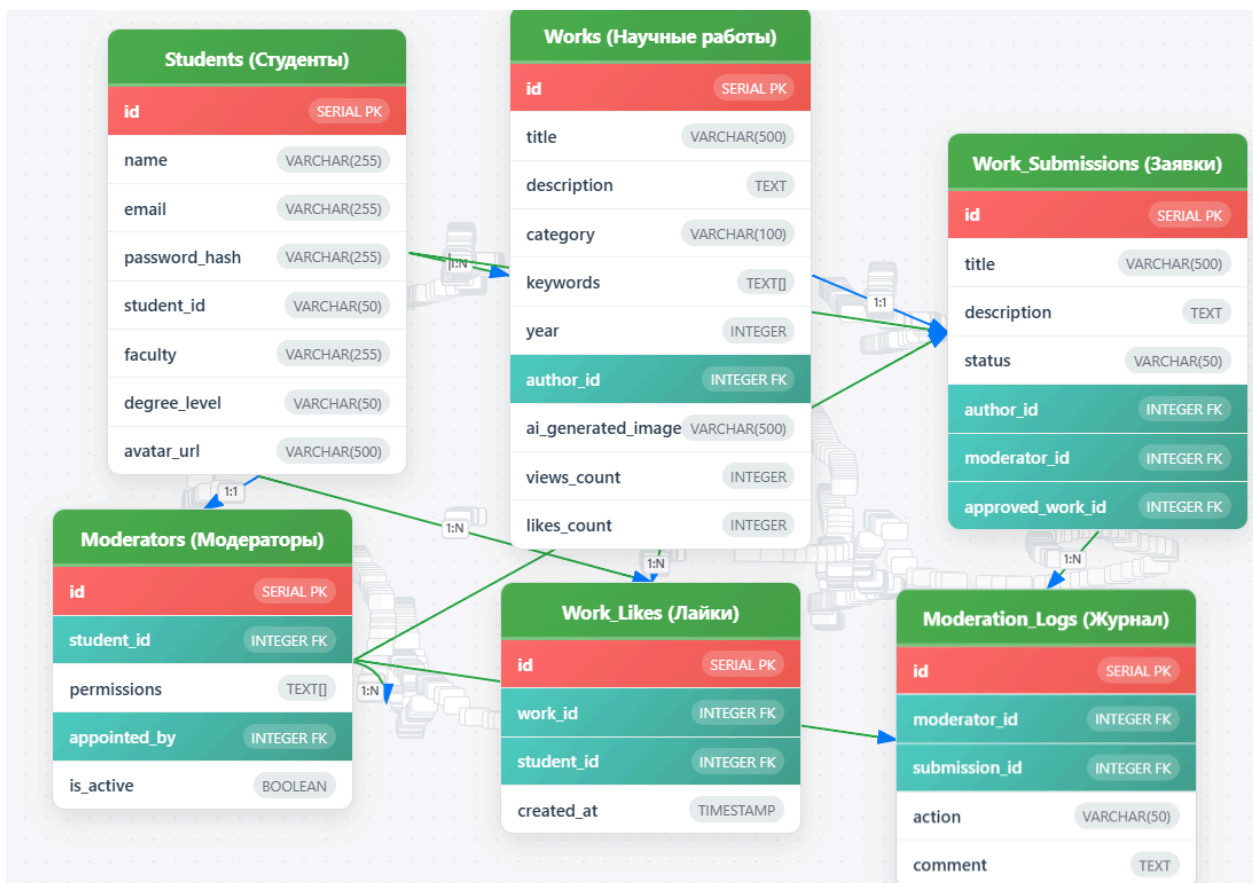


Рисунок 2.1. База данных веб-приложения учёта и анализа проектов КФУ

2.3. Проектирование архитектуры REST API

API предоставляет структурированный доступ к функциональным возможностям платформы через стандартизированные HTTP-методы и JSON-форматированные данные.

Архитектурное решение интерфейса включает шесть основных функциональных групп маршрутов, реализующих полный жизненный цикл работы с академическим контентом (рисунок 2.2):

Маршруты аутентификации (Authentication Endpoints) - Подсистема аутентификации и авторизации

POST (отправление) /api/auth/register - Регистрация новых участников. Критически важный маршрут, обеспечивающий первичную верификацию пользователей через корпоративную электронную почту университета. Реализует валидацию входных данных: проверку принадлежности к домену @stud.kpfu.ru, уникальность студенческого билета. Применяет криптографическое хеширование [7] паролей с использованием алгоритма bcrypt.

POST /api/auth/login - Авторизация зарегистрированных пользователей. Обеспечивает безопасный вход в систему через верификацию учетных данных и генерацию сессионных токенов. Возвращает расширенную информацию о профиле пользователя, включая количество опубликованных работ и академический статус.

Менеджер заявок (Submission Management) - Система управления заявками на публикацию

POST /api/submissions - Подача заявок на размещение исследовательских материалов. Центральный маршрут для инициации процесса публикации работ. Поддерживает мультимедийную загрузку файлов, автоматическую генерацию метаданных и парсинг (отлов) [8]

ключевых слов. Реализует файловую систему с организованной структурой каталогов.

GET (получение) /api/submissions/my/{author_id} - Персональный кабинет отслеживания заявок. Предоставляет авторам доступ к истории поданных материалов с детализацией статусов рассмотрения, комментариев экспертов и временных рамок обработки.

Репозитории контента (Content Repository) - Основной репозиторий академического контента

GET /api/works - Каталог опубликованных исследовательских работ. Высокопроизводительный маршрут для отображения верифицированного контента с расширенными возможностями фильтрации и сортировки. Реализует полнотекстовый поиск с поддержкой русскоязычной морфологии, категориальную группировку, факультетскую принадлежность и временные интервалы публикации.

GET /api/works/{work_id} - Детальное представление исследовательских проектов. Обеспечивает углубленный просмотр материалов с автоматическим учетом метрик популярности и отображением авторской информации. Включает механизм отслеживания просмотров для аналитических целей.

POST	/api/auth/register	Register Student	▼
POST	/api/auth/login	Login Student	▼
GET	/api/profile/{student_id}	Get Student Profile	▼
POST	/api/works	Create Work Direct	▼
GET	/api/works	Get All Works	▼
GET	/api/works/{work_id}	Get Work Detail	▼
POST	/api/works/{work_id}/like	Toggle Like Work	▼
GET	/api/stats/dashboard	Get Dashboard Stats	▼
GET	/api/search	Search Works	▼
POST	/api/submissions	Submit Work	▼
GET	/api/submissions/my/{author_id}	Get My Submissions	▼
GET	/api/moderation/submissions	Get Pending Submissions	▼
POST	/api/moderation/submissions/{submission_id}/action	Moderate Submission	▼
POST	/api/admin/moderators	Appoint Moderator	▼
GET	/api/moderation/stats/{moderator_id}	Get Moderation Stats	▼
GET	/api/files/download/{work_id}	Download Work File	▼
GET	/api/health	Health Check	▼
GET	/	Root	▼
GET	/register	Register Page	▼
GET	/login	Login Page	▼

Рисунок 2.2. Архитектура REST API

2.4. Определение стека технологий и инструментов разработки проекта

При создании веб-платформы для демонстрации научных и исследовательских работ студентов были проанализированы современные технологические решения, способные обеспечить высокую производительность, масштабируемость и удобство использования. Факторы, которые сыграли важную роль при выборе технологий и инструментов:

- производительность и скорость обработки запросов для комфортной работы пользователей;
- надежность хранения и обработки научных данных;
- возможность интеграции с системами машинного обучения для автоматической генерации контента;

- простота развертывания и поддержки веб-сервиса;

Архитектурные решения и используемые технологии

1. Язык программирования Python [9]

Python выбран в качестве основного языка разработки благодаря своей универсальности и богатой экосистеме библиотек. Данный язык программирования обладает читаемым синтаксисом, что упрощает разработку и последующее сопровождение кода. Особенно важным преимуществом является наличие развитых библиотек для работы с машинным обучением и обработкой естественного языка, что критично для реализации функционала автоматической генерации изображений на основе содержимого научных работ.

2. Веб-фреймворк FastAPI [10]

Для построения веб-сервиса была выбрана современная библиотека FastAPI, которая обеспечивает высокую производительность асинхронной обработки запросов. Фреймворк автоматически генерирует документацию API, поддерживает валидацию данных через Pydantic и обеспечивает типизацию, что значительно повышает надежность разрабатываемого решения. FastAPI также предоставляет встроенную поддержку загрузки файлов, что очень полезно для нашего случая с научными работами.

3. Сервер Uvicorn [11]

Для развертывания веб-приложения, который работает асинхронно, используется инструмент Uvicorn, обеспечивающий высокую производительность. Этот сервер оптимизирован для работы с FastAPI и обеспечивает эффективную обработку множественных одновременных подключений, что важно для платформы с потенциально большим количеством пользователей.

4. Система управления базами данных PostgreSQL

В качестве основного хранилища данных выбрана реляционная СУБД PostgreSQL. Данная система обеспечивает надежное хранение структурированной информации о пользователях, их научных работах и метаданных. PostgreSQL поддерживает расширенные типы данных, полнотекстовый поиск и обладает высокой производительностью при работе с большими объемами информации. Связь с базой данных реализована через asynсpg - асинхронный драйвер, благодаря которому запросы к БД не замедляют работу остальных частей системы.

5. Библиотека Pydantic для валидации данных [12]

Pydantic применяется для создания схем данных и их автоматической валидации. Это обеспечивает контроль корректности поступающих данных и автоматическое преобразование типов, что повышает стабильность работы системы и упрощает отладку.

6. Управление зависимостями через Poetry [13]

Управление зависимостями проекта осуществляется с помощью современного инструмента Poetry, который обеспечивает точное воспроизведение окружения разработки и упрощает развертывание приложения в различных средах.

3. Разработка веб-приложения учёта и анализа проектов

3.1. Создание архитектуры веб-платформы

При разработке современной веб-платформы критически важно обеспечить грамотное распределение функциональных обязанностей между различными модулями системы. Каждый элемент архитектуры должен решать конкретные задачи, не пересекаясь с компетенциями других компонентов. Создаваемая платформа представления студенческих научных работ построена по модульному принципу, где каждый блок отвечает за свою специфическую область функционирования.

Основные архитектурные блоки системы (Рисунок 3.1):

1. Модуль управления данными. Данный сегмент системы координирует все операции, связанные с хранением и извлечением информации. В его состав входят модели данных, описывающие структуру пользовательских профилей и научных публикаций, интерфейсы доступа к данным, обеспечивающие CRUD-операции для каждой таблицы, центральный класс управления базой данных и слой абстракции, который изолирует бизнес-логику от прямого взаимодействия с хранилищем данных.

2. Серверная инфраструктура. Этот компонент обеспечивает обработку HTTP-запросов и управление сессиями пользователей. Включает в себя:

- RESTful API для взаимодействия с клиентской частью;
- систему аутентификации через корпоративную почту университета;
- обработчики загрузки файлов и документов.

3. Пользовательский интерфейс. Фронтенд-составляющая представлена набором взаимосвязанных веб-страниц:

- портал авторизации для входа через университетскую учетную запись;
- главная витрина с каталогом исследовательских проектов;
- персональный кабинет для управления публикациями;
- форма подачи новых научных материалов;
- административная панель контроля контента.

Принципы взаимодействия компонентов. Архитектура построена на принципе слабой связанности, где каждый модуль взаимодействует с остальными через четко определенные интерфейсы. Это обеспечивает:

- возможность независимого развития отдельных частей системы;
- масштабируемость решения при росте нагрузки;
- простоту внедрения новых функциональных возможностей.

Подобная организация кодовой базы способствует созданию устойчивой и расширяемой платформы, которая может эффективно адаптироваться к изменяющимся требованиям академической среды и растущим потребностям пользователей.

Архитектура веб-платформы студенческих научных работ



Рисунок 3.1. Основная инфраструктура приложения

3.2. Интеграция PostgreSQL базы данных

В проекте академической витрины для хранения научных работ мы остановились на PostgreSQL — это надежная реляционная база данных, которая хорошо справляется с большими объемами информации и сложными запросами. Чтобы работать с этой БД, выбрали библиотеку `asynsrg` - она позволяет обращаться к PostgreSQL асинхронно, то есть система может заниматься другими делами, пока идет обработка запроса к базе.

Архитектура подключения к базе данных

Система управления подключениями к базе данных реализована через пул соединений, что позволяет эффективно управлять ресурсами и обеспечивать высокую производительность при множественных запросах.

Параметры подключения загружаются из переменных окружения через класс `Settings` в модуле конфигурации, что обеспечивает гибкость развертывания в различных средах (код представлен ниже). База данных поддерживает настройку минимального и максимального количества соединений в пуле (от 1 до 10), таймауты команд и специфические настройки сервера.

```
class Settings:
    """Класс для управления настройками приложения"""

    # Настройки базы данных
    DB_USER: str = os.getenv("DB_USER", "postgres")
    DB_PASSWORD: str = os.getenv("DB_PASSWORD", "password")
    DB_HOST: str = os.getenv("DB_HOST", "localhost")
    DB_PORT: str = os.getenv("DB_PORT", "5432")
    DB_NAME: str = os.getenv("DB_NAME", "unisite_db")

    # Настройки сервера
    HOST: str = os.getenv("HOST", "127.0.0.1")
    PORT: int = int(os.getenv("PORT", "8000"))
    DEBUG: bool = os.getenv("DEBUG", "true").lower() in ("true", "1",
"yes")
```

```

# Настройки безопасности
SECRET_KEY: str = os.getenv("SECRET_KEY",
"your-secret-key-change-in-production")
ALGORITHM: str = os.getenv("ALGORITHM", "HS256")
ACCESS_TOKEN_EXPIRE_MINUTES: int =
int(os.getenv("ACCESS_TOKEN_EXPIRE_MINUTES", "30"))

# Настройки логирования
LOG_LEVEL: str = os.getenv("LOG_LEVEL", "INFO")

CORS_ORIGINS: list = os.getenv("CORS_ORIGINS",
"http://localhost:3000,http://127.0.0.1:3000").split(",")

@property
def database_url(self) -> str:
    """Получить URL для подключения к базе данных"""
    return
f"postgresql://{self.DB_USER}:{self.DB_PASSWORD}@{self.DB_HOST}:{self.
DB_PORT}/{self.DB_NAME}"

def __repr__(self) -> str:
    return f"Settings(host={self.HOST}, port={self.PORT},
db_name={self.DB_NAME})"

```

Оптимизация производительности. Для обеспечения быстрой работы системы создана система индексов по критическим полям - email пользователей, датам создания записей, статусам заявок на модерацию. Это значительно ускоряет выполнение часто используемых запросов поиска и фильтрации.

Реализованы автоматические триггеры для обновления временных меток, что обеспечивает актуальность метаданных без дополнительной логики в приложении.

Управление транзакциями и безопасность Система предоставляет набор утилитных функций для выполнения различных типов запросов - `execute_query()` для команд изменения данных, `fetch_query()` для получения

множественных результатов и `fetchrow_query()` для единичных записей. Все операции выполняются в контексте управляемых соединений из пула, что гарантирует правильное освобождение ресурсов.

Безопасность обеспечивается через параметризованные запросы, что предотвращает SQL-инъекции [14], и систему внешних ключей с каскадным удалением для поддержания целостности данных.

Мониторинг и диагностика Встроенная система логирования отслеживает все критические операции - создание пула соединений, выполнение миграций, ошибки подключения (пример показан ниже). Функция `check_database_connection()` позволяет проводить регулярную диагностику состояния подключения для обеспечения отказоустойчивости системы.

2025-06-09 12:53:57,513 - *server.server* - INFO - 🚀 Запуск UniSite API...

2025-06-09 12:53:57,594 - *server.database.db_con* - INFO - ✅ Пул соединений с базой данных создан

2025-06-09 12:53:57,609 - *server.database.db_con* - INFO - ✅ Таблицы и индексы созданы успешно

2025-06-09 12:53:57,610 - *server.server* - INFO - ✅ База данных успешно инициализирована

Такой вид имеют логи серверной части.

3.3. Реализация пользовательского интерфейса веб-приложения

При загрузке веб-приложения первым делом открывается главная страница академической витрины. Она имеет адаптивный дизайн, который корректно отображается на различных устройствах - от настольных компьютеров до мобильных телефонов. На момент первого посещения пользователь видит полнофункциональную страницу с демонстрацией возможностей платформы (рисунок 3.2).

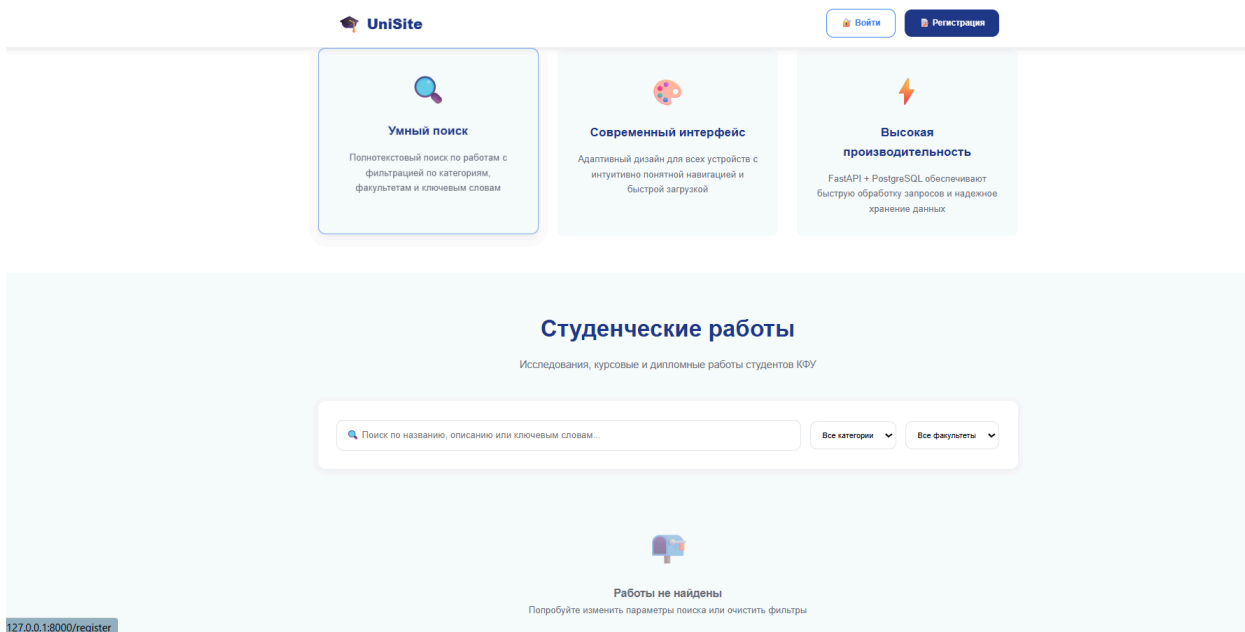


Рисунок 3.2. Главная страница веб-приложения с секциями возможностей и студенческих работ.

Страница состоит из нескольких ключевых компонентов, которые обеспечивают удобную навигацию и информирование пользователей:

Секция возможностей платформы. Следующий блок представляет возможности системы через сетку из шести карточек (рисунок 3.3). Каждая карточка содержит:

- иконку-эмодзи для визуальной идентификации функции;
- заголовок возможности;
- детальное описание функционала.

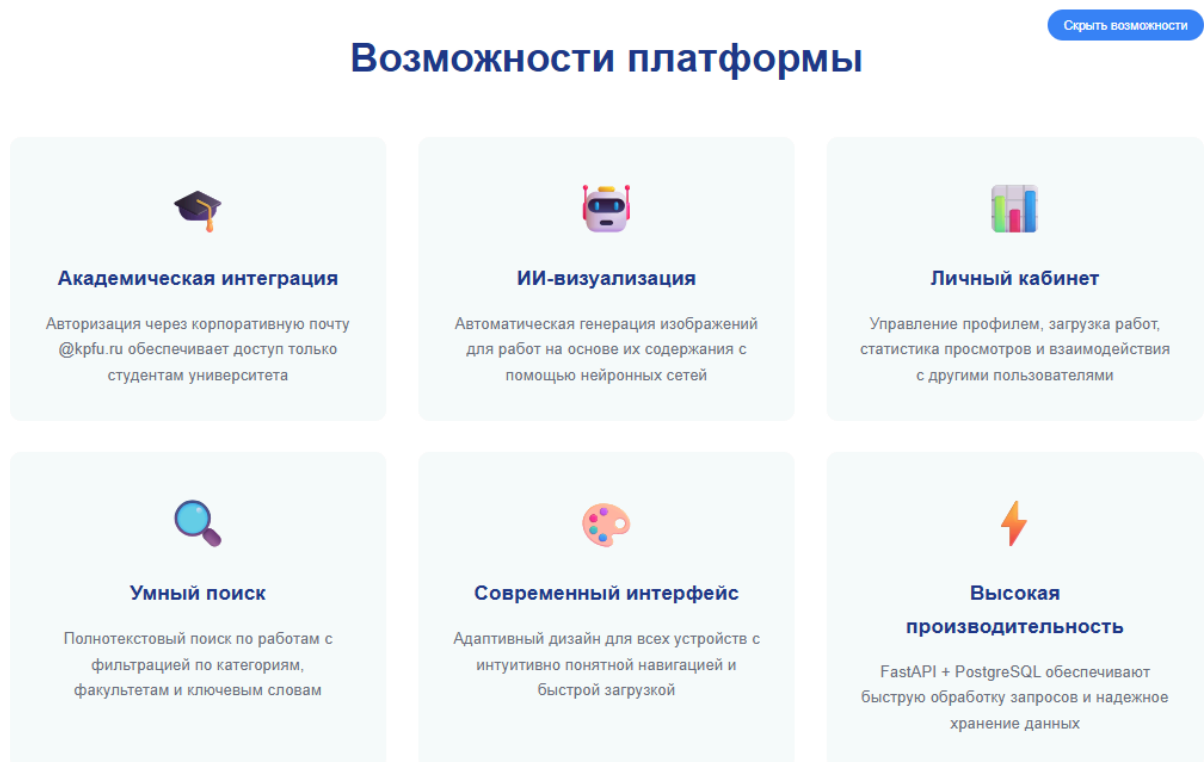


Рисунок 3.3. Интерактивная секция возможностей платформы с функцией сворачивания.

Особенностью данной секции является интерактивная кнопка "Скрыть возможности", которая позволяет пользователю сворачивать и разворачивать содержимое для экономии места на странице и улучшения пользовательского опыта.

Витрина студенческих работ. Основная функциональная секция страницы представляет собой витрину академических работ студентов КФУ (рисунок 3.4). Данная область включает несколько компонентов:

Система поиска и фильтрации

- поле полнотекстового поиска с поддержкой поиска по названию, описанию и ключевым словам;

- выпадающий список категорий (НИР, ВКР, Курсовая, Диссертация, Статья);
- фильтр по факультетам (ИВМиИТ, Физический, Химический и др.).

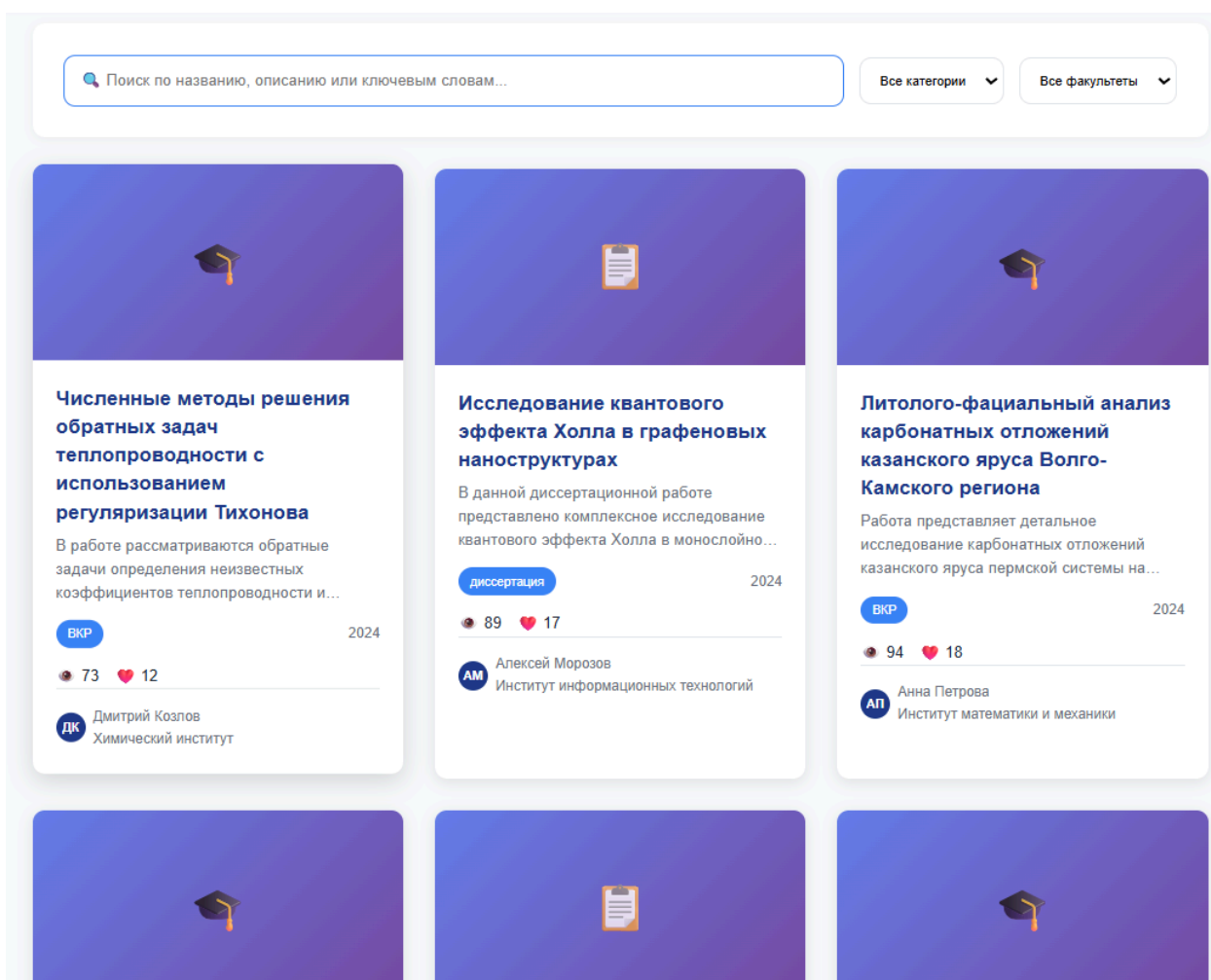


Рисунок 3.4. Интерфейс поиска и фильтрации академических работ с адаптивной сеткой карточек.

Сетка карточек работ. Результаты отображаются в виде адаптивной сетки карточек, где каждая карточка содержит:

- визуальную область с тематической иконкой, соответствующей категории работы;
- заголовок работы;
- краткое описание с ограничением по количеству строк;
- статистику взаимодействия: количество просмотров и лайков;
- информацию об авторе с аватаром, инициалами, именем и факультетом.

Переход к детальному просмотру работы. При клике на любую карточку работы пользователь переходит на страницу детального просмотра (рисунок 3.5). Данная страница предоставляет исчерпывающую информацию о выбранной академической работе.

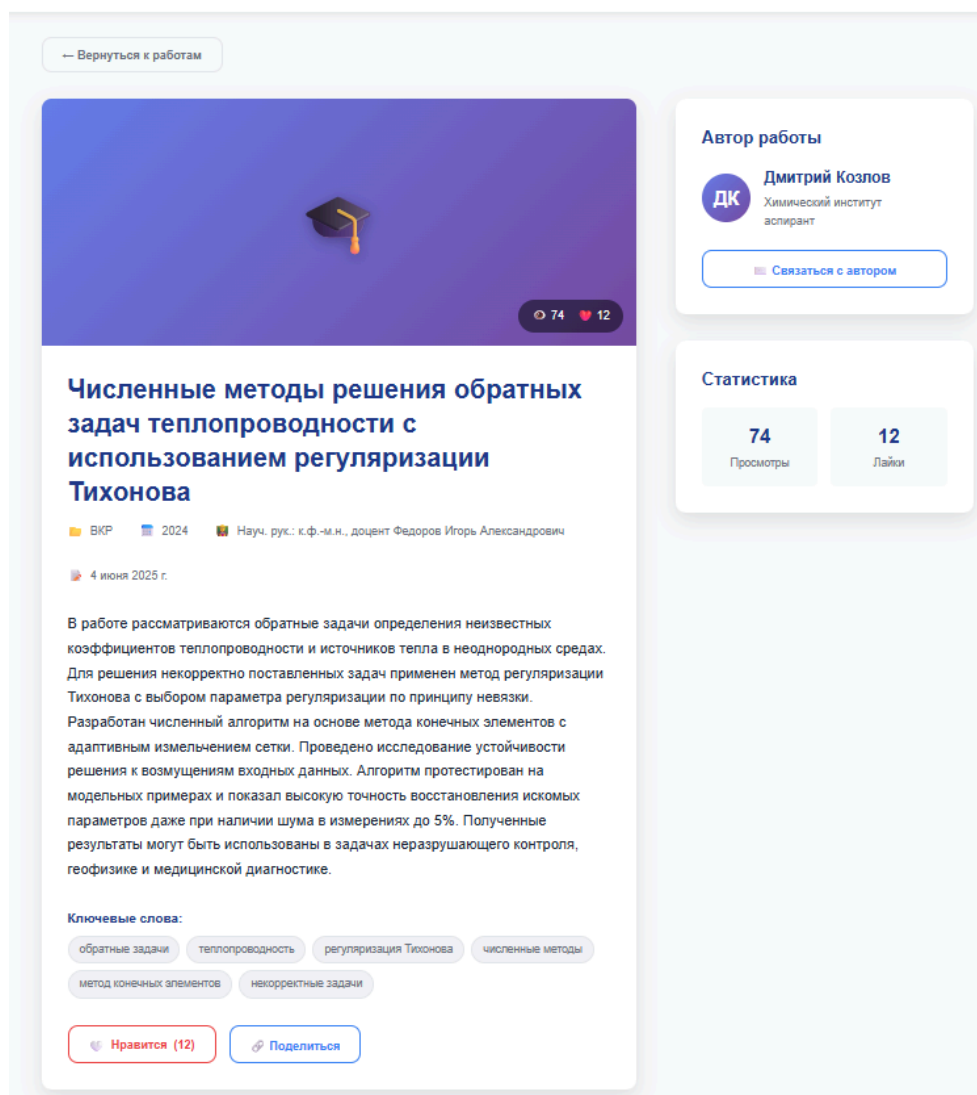


Рисунок 3.5. Страница детального просмотра академической работы с двухколоночной структурой.

Страница детального просмотра работы построена на основе современных принципов UX/UI дизайна и обеспечивает пользователям полный доступ к информации об академической работе. Архитектура страницы включает несколько ключевых компонентов:

Навигационный блок содержит кнопку возврата "Вернуться к работам", которая позволяет пользователю легко вернуться к списку работ или предыдущей странице в браузерной истории.

Основная область контента представляет детальную информацию о работе через несколько взаимосвязанных секций:

Героическая секция с визуальным представлением работы включает градиентный фон, крупную тематическую иконку категории и наложенную статистику просмотров и лайков. Данный подход обеспечивает мгновенное визуальное восприятие типа и популярности работы.

Информационный блок структурированно представляет:

- полный заголовок работы с академическим форматированием;
- расширенные метаданные: категорию, год создания, научного руководителя, дату публикации;
- полное описание работы с сохранением авторского форматирования;
- систему ключевых слов в виде интерактивных тегов для улучшения поисковых возможностей.

Панель интерактивных действий предоставляет пользователям возможности для взаимодействия с контентом:

Система лайков реализована с динамическим обновлением состояния - кнопка изменяет визуальное представление (от белого контурного сердца до красного заполненного) и автоматически обновляет счетчик лайков при взаимодействии пользователя.

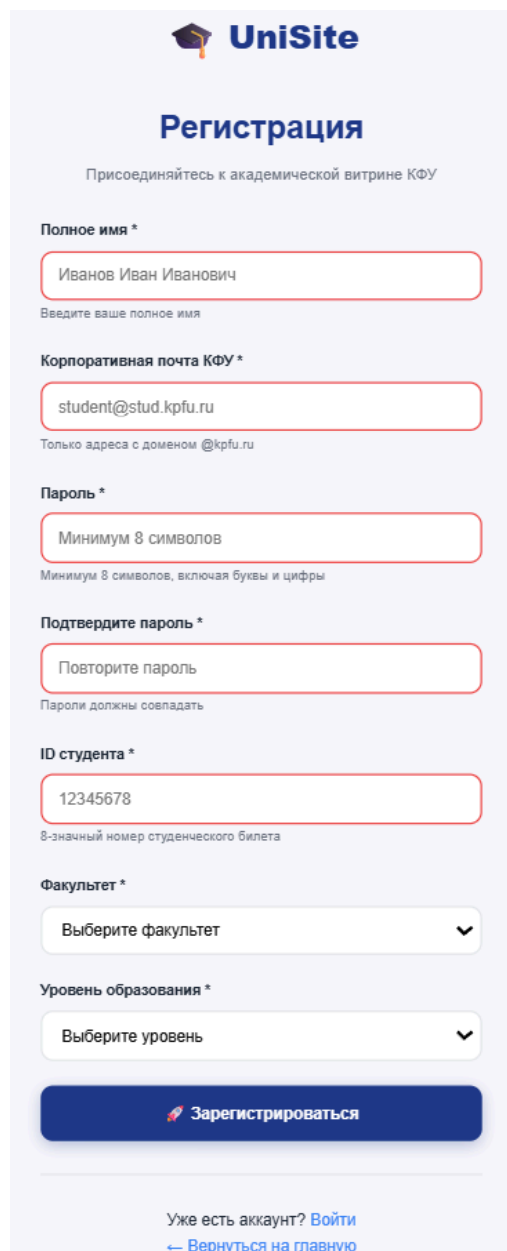
Функция скачивания файлов активируется при наличии прикрепленного к работе документа и обеспечивает безопасную загрузку через защищенные API-маршрутов.

Механизм "Поделиться" использует современные веб-технологии: нативный Web Share API для поддерживаемых браузеров или копирование ссылки в буфер обмена с соответствующими уведомлениями.

Боковая информационная панель дополняет основной контент двумя специализированными карточками:

Карточка автора включает визуальный аватар с инициалами, полную информацию об авторе (имя, факультет, уровень образования) и интерактивную кнопку для связи через электронную почту с предзаполненной темой и шаблоном письма.

Страница регистрации пользователей. Система регистрации представляет собой важный компонент веб-приложения, обеспечивающий контролируемый доступ к функционалу академической витрины. Страница регистрации реализована с учетом современных принципов безопасности и пользовательского опыта (рисунок 3.6).



The image shows a registration form for UniSite. At the top is the UniSite logo, which includes a graduation cap icon. Below the logo is the title "Регистрация" (Registration) and a subtitle "Присоединяйтесь к академической витрине КФУ" (Join the academic showcase of KFU). The form contains several input fields: "Полное имя *" (Full name) with the example "Иванов Иван Иванович"; "Корпоративная почта КФУ *" (Corporate email) with the example "student@stud.kpfu.ru"; "Пароль *" (Password) with a placeholder "Минимум 8 символов"; "Подтвердите пароль *" (Confirm password) with a placeholder "Повторите пароль"; and "ID студента *" (Student ID) with the example "12345678". There are also two dropdown menus for "Факультет *" (Faculty) and "Уровень образования *" (Education level), both with placeholder text "Выберите..." and a downward arrow. At the bottom of the form is a blue button with a red flame icon and the text "Зарегистрироваться" (Register). Below the button, there are two links: "Уже есть аккаунт? Войти" (Already have an account? Log in) and "← Вернуться на главную" (← Return to the main page).

UniSite

Регистрация

Присоединяйтесь к академической витрине КФУ

Полное имя *

Иванов Иван Иванович

Введите ваше полное имя

Корпоративная почта КФУ *

student@stud.kpfu.ru

Только адреса с доменом @kpfu.ru

Пароль *

Минимум 8 символов

Минимум 8 символов, включая буквы и цифры

Подтвердите пароль *

Повторите пароль

Пароли должны совпадать

ID студента *

12345678

8-значный номер студенческого билета

Факультет *

Выберите факультет

Уровень образования *

Выберите уровень

Зарегистрироваться

Уже есть аккаунт? [Войти](#)

[← Вернуться на главную](#)

Рисунок 3.6. Окно регистрации пользователей

Основные компоненты страницы регистрации:

Навигационный заголовок содержит логотип системы UniSite с эмодзи-иконкой академической шапочки, обеспечивающий визуальную идентификацию платформы и возможность возврата на главную страницу.

Форма регистрации структурирована следующим образом:

Поле полного имени с валидацией минимальной длины и обязательным заполнением, обеспечивающее корректную идентификацию пользователя в системе.

Поле корпоративной электронной почты с жесткой валидацией домена @stud.kpfu.ru, гарантирующее принадлежность пользователя к академическому сообществу КФУ. Система проверяет соответствие введенного адреса корпоративному стандарту в режиме реального времени.

Поля пароля и подтверждения пароля с криптографическими требованиями: минимальная длина 8 символов, обязательное включение букв и цифр. Система автоматически сравнивает введенные пароли и предоставляет мгновенную обратную связь о их соответствии.

Поле ID студента с форматной валидацией точно 8-значного числового значения, соответствующего формату студенческих билетов КФУ.

Выпадающий список факультетов, включающий все основные структурные подразделения университета: от Института вычислительной математики и информационных технологий до Филологического факультета, обеспечивая точную институциональную привязку пользователя.

Селектор уровня образования с вариантами от бакалавра до аспиранта, позволяющий системе адаптировать функционал под академический статус пользователя.

Система валидации и обратной связи реализована на нескольких уровнях:

Клиентская валидация в режиме реального времени обеспечивает немедленную проверку корректности вводимых данных с визуальными индикаторами ошибок и подсказками.

Серверная валидация гарантирует безопасность и целостность данных на уровне API, предотвращая возможные попытки обхода клиентских проверок.

Страница личного кабинета пользователя. После успешной регистрации и авторизации пользователь получает доступ к персонализированному интерфейсу управления - личному кабинету. Данная страница представляет собой центральную панель управления всеми функциями системы и персональными данными пользователя (рисунок 3.7).

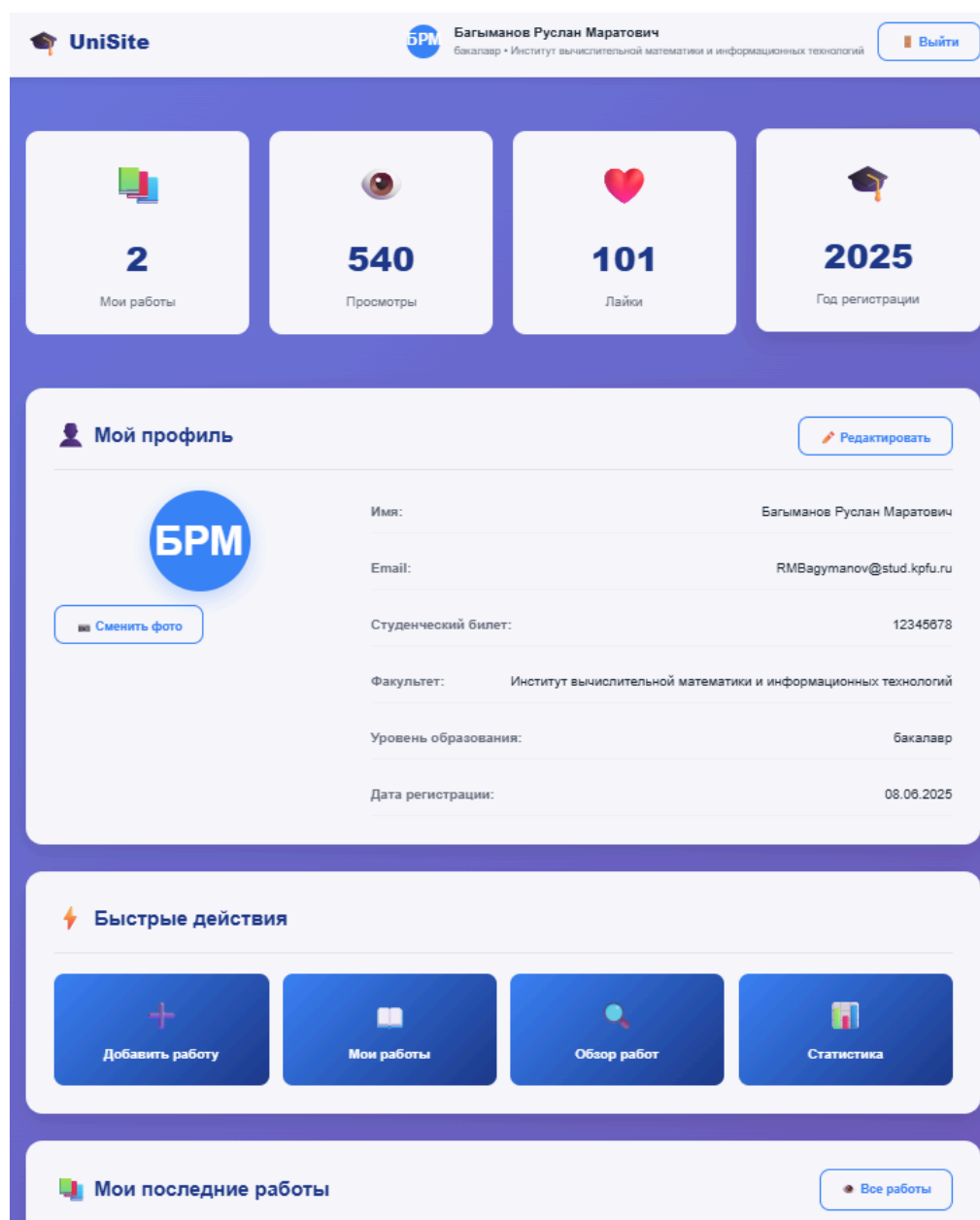


Рисунок 3.7. Интерфейс личного кабинета пользователя со статистическими показателями.

Архитектура личного кабинета построена на основе современного подхода к дизайну досок информации и включает несколько функциональных зон:

Динамический заголовок страницы обеспечивает персонализированную навигацию и идентификацию пользователя:

Брендинговый логотип UniSite с интерактивными эффектами при наведении курсора, обеспечивающий возможность возврата на главную страницу системы.

Персонализированный блок пользователя, включающий круглый аватар с инициалами, генерируемый автоматически на основе полного имени, полное имя пользователя и академическую информацию в формате "Уровень образования • Факультет".

Кнопка безопасного выхода из системы с подтверждающим диалогом, предотвращающим случайную деавторизацию.

Интерактивная панель статистики представляет ключевые метрики активности пользователя через адаптивную сетку из четырех информационных карточек:

Карточка "Мои работы" отображает общее количество опубликованных академических работ пользователя с тематической иконкой книг.

Карточка "Просмотры" агрегирует суммарное количество просмотров всех работ пользователя, демонстрируя популярность контента.

Карточка "Лайки" показывает общее количество положительных оценок, полученных работами пользователя от других участников системы.

Карточка "Год регистрации" содержит информацию о дате присоединения пользователя к академическому сообществу платформы.

Секция профиля пользователя структурирована в двухколоночном формате и обеспечивает комплексное представление персональной информации:

Левая колонка содержит визуальный блок с увеличенным аватаром пользователя (120x120 пикселей) и кнопкой для будущего функционала смены фотографии профиля.

Правая колонка представляет детализированную информацию в виде структурированного списка пар "параметр-значение":

- полное имя пользователя;
- корпоративный email-адрес;
- номер студенческого билета;
- принадлежность к факультету;
- уровень образования;
- дата регистрации в системе с локализованным форматированием.

Панель быстрых действий обеспечивает оперативный доступ к основным функциям системы через сетку интерактивных кнопок с градиентным дизайном:

"Добавить работу" - переход к форме публикации новой академической работы;

"Мои работы" - доступ к полному каталогу работ пользователя;

"Обзор работ" - переход к общему каталогу всех работ в системе;

"Статистика" - доступ к детализированной аналитике активности пользователя.

Секция последних работ представляет краткий обзор недавно опубликованных работ пользователя с возможностью перехода к полному

списку. Каждая работа отображается в виде информационного блока, содержащего:

- заголовок работы;
- метаданные: категорию, год создания, научного руководителя;
- статистические показатели: количество просмотров, лайков и

ключевые слова.

Страница добавления научной работы. После успешной авторизации в системе пользователи получают доступ к функционалу публикации академических работ через специализированную форму добавления контента. Данная страница представляет собой комплексный интерфейс для создания и публикации научных материалов с интегрированными возможностями искусственного интеллекта (рисунок 3.8).

Название работы *

Курсовая работа

Минимум 10 символов. Укажите точное и информативное название. 15/200

Описание работы *

lsdkvh;kjvidsjvjsdfbjsdjlbfjdfsljbfidsjfi;dsjilfbjfiljbdsi;lfjbi;dfsjb;ij

Минимум 50 символов. Подробное описание поможет создать более точное изображение ИИ. 0/2000

Категория работы *

Курсовая работа

Ключевые слова *

Нитс x модераторс x адмб x

Введите ключевое слово и нажмите Enter

Добавьте 3-10 ключевых слов, характеризующих вашу работу. Нажмите Enter после ввода каждого слова.

Год написания *

2025

Научный руководитель

ФИО научного руководителя (необязательно)

Укажите ФИО и должность вашего научного руководителя (если есть). 0/100

Файл работы

Выберите файл или перетащите сюда

Поддерживаются форматы: PDF, DOC, DOCX (макс. 10 МБ)

Курсовая_Багыманов_Руслан_Маратович_08_254.docx (0.01 МБ)

Загрузите файл вашей работы. Система проанализирует содержимое для создания изображения.

Предварительный просмотр изображения ИИ

Рисунок 3.8. Интерфейс страницы добавления научной работы с
AI-генерацией изображений.

Архитектурное решение страницы включает следующие функциональные компоненты:

Карточка формы использует технологию блюра для создания полупрозрачного эффекта размытия, обеспечивая читаемость контента при сохранении эстетической привлекательности фона.

Форма публикации работы структурирована в соответствии с академическими стандартами и включает следующие обязательные и опциональные поля:

Поле названия работы с валидацией минимальной длины (10 символов) и максимального ограничения (200 символов), обеспечивающее корректное именование академических материалов.

Расширенное поле описания работы с требованием минимального объема в 50 символов и ограничением до 2000 символов, позволяющее авторам предоставить исчерпывающую информацию о содержании, целях, методах и результатах исследования.

Селектор категории работы, включающий полный спектр академических типов: от научно-исследовательских работ и выпускных квалификационных работ до диссертаций, научных статей, конференционных докладов, учебных проектов, курсовых работ и рефератов.

Система управления ключевыми словами реализована через интерактивный интерфейс с динамическим добавлением тегов. Пользователи могут добавлять от 3 до 10 ключевых слов, каждое из которых отображается в виде интерактивного тега с возможностью удаления. Система автоматически валидирует уникальность добавляемых тегов и предоставляет визуальную обратную связь.

Поле года написания работы с валидацией диапазона от 2000 до 2025 года, обеспечивающее актуальность публикуемых материалов.

Опциональное поле научного руководителя с ограничением в 100 символов для указания ФИО и должности научного наставника.

Инновационная система загрузки файлов реализована с использованием современных веб-технологий:

Перетаски-и-брось интерфейс обеспечивает интуитивную загрузку документов через перетаскивание файлов в специально выделенную область с визуальными индикаторами активности.

Автоматическая валидация загружаемых файлов включает проверку формата (поддерживаются PDF, DOC, DOCX), размера файла (максимум 10 МБ) и типа содержания. Визуальный индикатор загруженного файла

отображает название документа, размер и предоставляет возможность отмены загрузки.

Система генерации изображений с использованием искусственного интеллекта представляет уникальную функциональность платформы:

Автоматическая генерация превью-изображения основана на анализе названия работы, описания, категории и ключевых слов через криптографический хеш-алгоритм SHA-256. Интерактивная кнопка "Создать превью" запускает процесс AI-генерации с визуальным индикатором загрузки и прогресс-баром. Блок предварительного просмотра отображает сгенерированное изображение в адаптивном контейнере с возможностью повторной генерации при необходимости.

Система валидации формы реализована на двух уровнях:

Клиентская валидация в режиме реального времени обеспечивает мгновенную проверку корректности вводимых данных с динамическими счетчиками символов и визуальными индикаторами ошибок.

Серверная валидация гарантирует безопасность и целостность данных на уровне API-маршрутов.

Функционал автосохранения черновиков использует localStorage [15] браузера для периодического сохранения введенных данных каждые 30 секунд, предотвращая потерю информации при случайном закрытии страницы или технических сбоях.

Автоматическое скрывание уведомлений через 5 секунд для предотвращения загромождения интерфейса. Анимированные переходы и плавные появления сообщений обеспечивают профессиональный пользовательский опыт.

4. Тестирование веб-приложения учета и анализа проектов

4.1. Тестирование функциональной части приложения

Проведение ручного тестирования пользовательского интерфейса UniSite включало моделирование ключевых сценариев взаимодействия пользователей с академической витриной. Основные тестируемые функции:

1. Регистрация нового студента и вход в систему.
2. Подача заявки на публикацию научной работы.
3. Модерация и публикация работ.
4. Просмотр и взаимодействие с опубликованными работами.
5. Управление модераторами и статистикой.

В первую очередь будем проверять регистрацию новых пользователей, а вместе с тем и валидацию корректных и некорректных данных. Для этого мы с главной страницы (рисунок 3.2) заходим на вкладку “Регистрация”, после чего мы и попадаем на форму регистрации пользователя. В ней же мы заполняем все поля правильно, кроме одно - id студента, этот id уже существует, о чем нам и сообщит система (рисунок 4.1).

Регистрация

Присоединяйтесь к академической витрине КФУ

Студент уже зарегистрирован

Полное имя *

Багыманов Руслан Маратович

Введите ваше полное имя

Корпоративная почта КФУ *

MRBagymanov@stud.kpfu.ru

Только адреса с доменом @stud.kpfu.ru

Пароль *

Минимум 8 символов, включая буквы и цифры

Подтвердите пароль *

Пароли должны совпадать

ID студента *

12345678

8-значный номер студенческого билета

Факультет *

Институт вычислительной математики и информ. ▼

Уровень образования *

Бакалавр ▼


 Зарегистрироваться

Рисунок 4.1. Пользователь с таким id уже имеется в базе

Теперь проверим регистрацию пользователя с некорректным почтовым доменом (рисунок 4.2).

Полное имя *

Багыманов Руслан Маратович

Введите ваше полное имя

Корпоративная почта КФУ *

MRBagymanov@kpfu.ru

Разрешены только а

Введите данные в указанном формате.

Пароль *

.....

Минимум 8 символов, включая буквы и цифры

Подтвердите пароль *

.....

Пароли должны совпадать

ID студента *

12345732

8-значный номер студенческого билета

Факультет *

Институт вычислительной математики и информ...

Рисунок 4.2. Отказ в создании записи

И в конце проверим, создается ли пользователь, если все данные верны (рисунок 4.3), а также проверим записи в базе данных (рисунок 4.4).

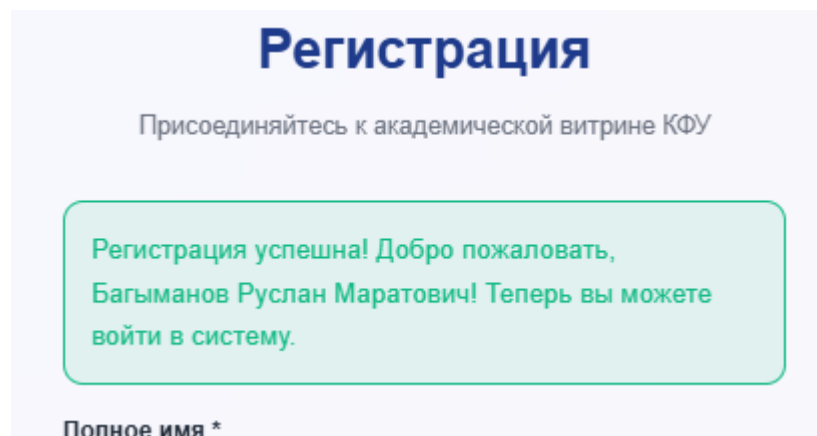


Рисунок 4.3. Регистрация прошла успешно

	id [PK] integer	name character varying (255)	email character varying (255)	password_hash character varying (255)	student_id character varying (50)	fac cha
1	1	Багыманов Руслан Маратович	RMBagymanov@stud.kpfu.ru	\$2b\$12\$ppj1/60IEWP4dCBHcZ4e40diiIHlNkQ8n0u.60TjgOiFhcXuYxtmq	12345678	Ин
2	17	Алексей Морозов	morozov@stud.kpfu.ru	\$2b\$12\$SLQv3c1yqBwfNu/1+MMBVMeYPLFQxqHry12Gtse4HFnCNjnFJBbXfG	2021001	Ин
3	18	Елена Волкова	volkova@stud.kpfu.ru	\$2b\$12\$SLQv3c1yqBwfNu/1+MMBVMeYPLFQxqHry12Gtse4HFnCNjnFJBbXfG	2020050	Фи
4	19	Дмитрий Козлов	kozlov@stud.kpfu.ru	\$2b\$12\$SLQv3c1yqBwfNu/1+MMBVMeYPLFQxqHry12Gtse4HFnCNjnFJBbXfG	2019025	Хи
5	20	Анна Петрова	petrova@stud.kpfu.ru	\$2b\$12\$SLQv3c1yqBwfNu/1+MMBVMeYPLFQxqHry12Gtse4HFnCNjnFJBbXfG	2022010	Ин
6	21	Сергей Николаев	nikolaev@stud.kpfu.ru	\$2b\$12\$SLQv3c1yqBwfNu/1+MMBVMeYPLFQxqHry12Gtse4HFnCNjnFJBbXfG	2020075	Гес
7	22	Багыманов Руслан Маратович	MRBagymanov@stud.kpfu.ru	\$2b\$12\$X./rH1k6gMPMX83VP4GdOG2TDNbjA3QTdEBbZpmeaNBGfzSz5Rj...	12345732	Ин

Рисунок 4.4. Пользователь был добавлен

Далее проверим функционал личного кабинета, а именно добавления новой научной работы. После перехода в личный кабинет мы спускаемся вниз по странице и заходим во вкладку “Добавить работу”. Как нас переадресовали на страницу добавления работы, нас встречает окно автозаполнения, которое помнит прошлую нашу попытку заполнения данной формы (рисунок 4.5).

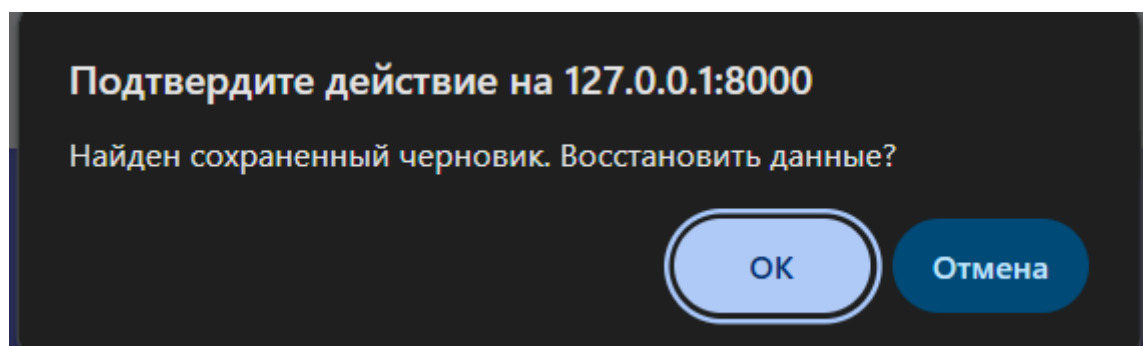
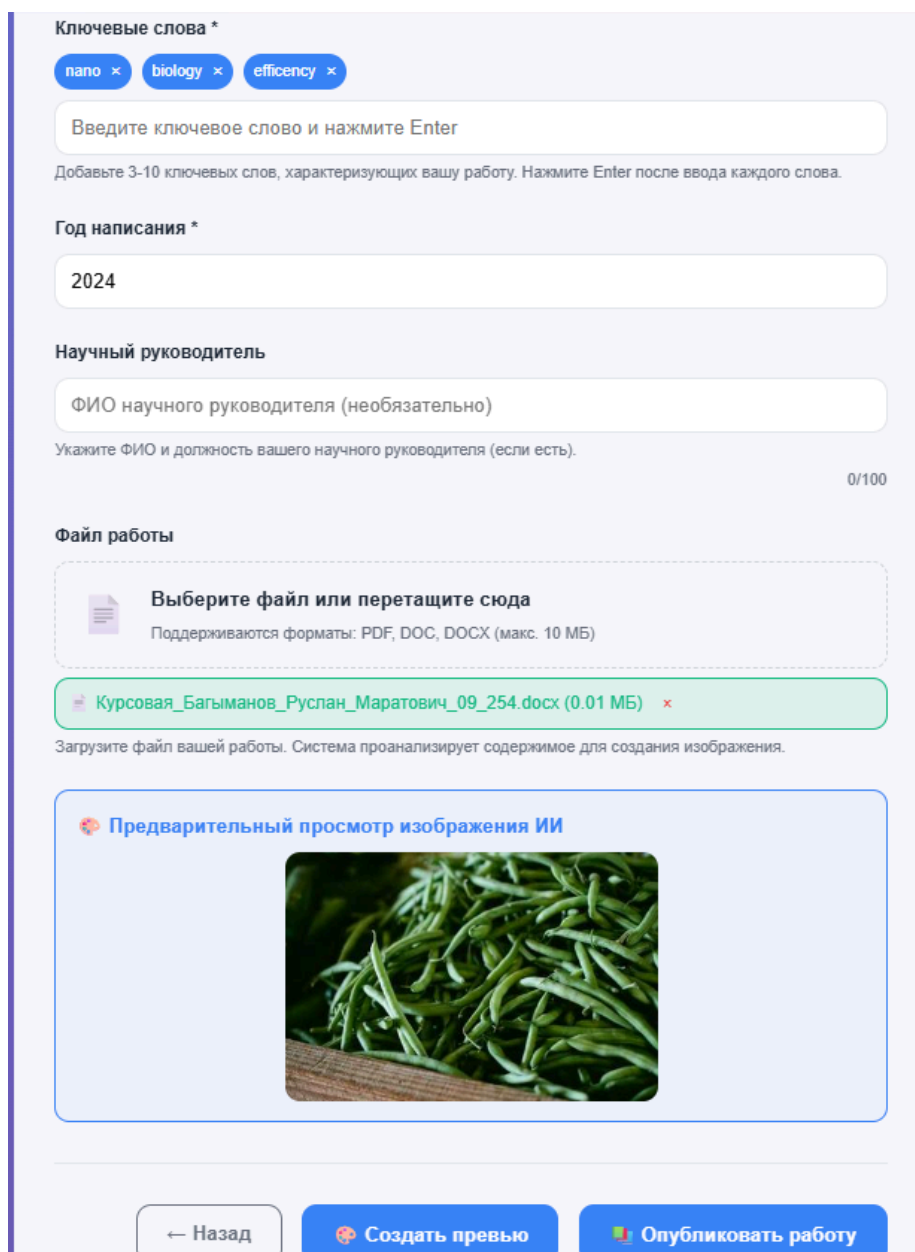


Рисунок 4.5. Черновик прошлой работы

После заполнения формы нам сразу генерируется искусственным интеллектом превью, которое описывает ключевые слова (рисунок 4.6).



Ключевые слова *

напо × biology × efficiency ×

Введите ключевое слово и нажмите Enter

Добавьте 3-10 ключевых слов, характеризующих вашу работу. Нажмите Enter после ввода каждого слова.

Год написания *

2024

Научный руководитель

ФИО научного руководителя (необязательно)

Укажите ФИО и должность вашего научного руководителя (если есть).

0/100

Файл работы


Выберите файл или перетащите сюда

Поддерживаются форматы: PDF, DOC, DOCX (макс. 10 МБ)

Курсовая_Багыманов_Руслан_Маратович_09_254.docx (0.01 МБ) ×

Загрузите файл вашей работы. Система проанализирует содержимое для создания изображения.

Предварительный просмотр изображения ИИ



← Назад Создать превью Опубликовать работу

Рисунок 4.6. Заполненная форма учебной работы

После этого мы можем зайти на страницу модерирования контента и увидеть, что проект там висит и ждет проверки и дальнейших событий (рисунок 4.7).

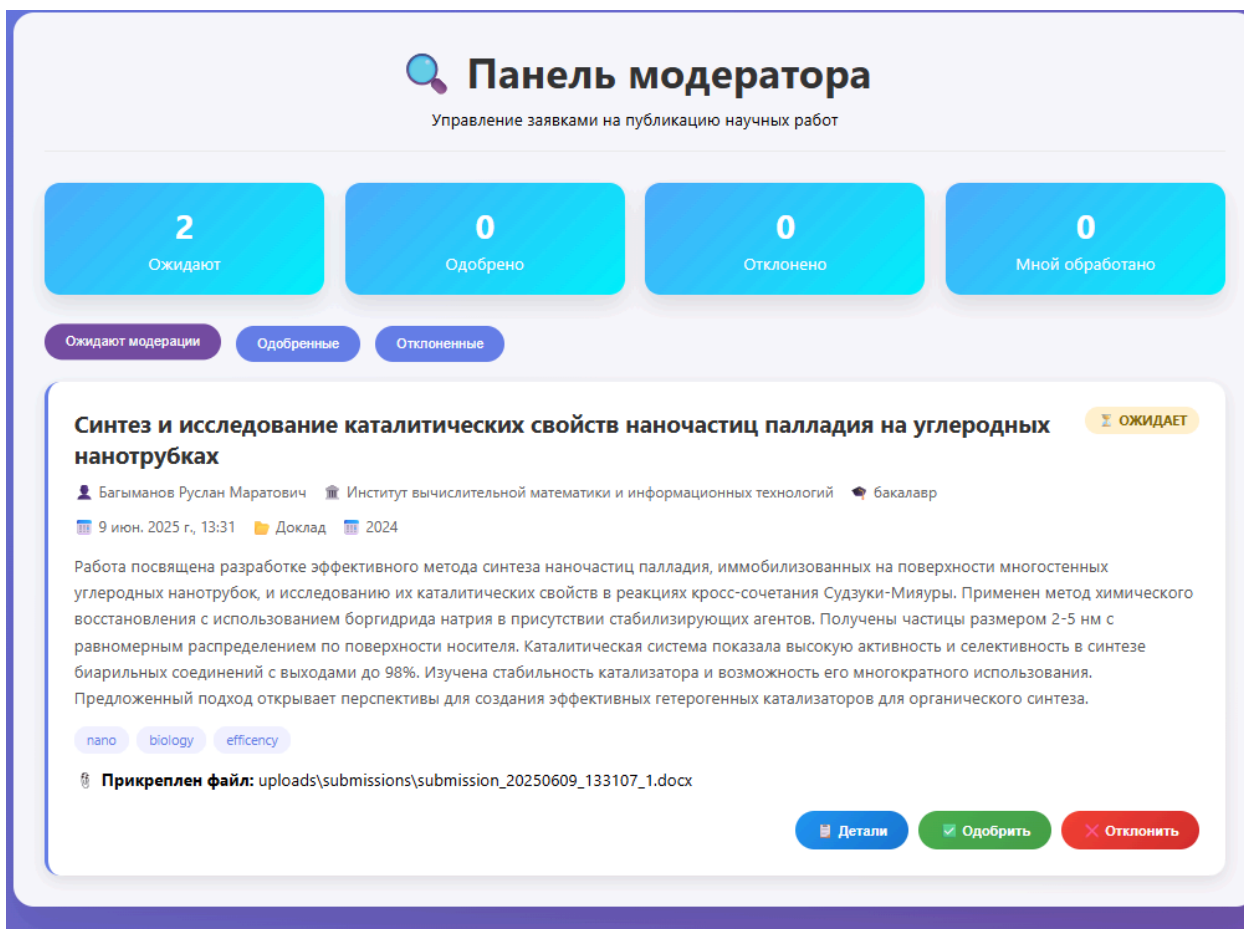


Рисунок 4.7. Работа в отделе модерирования

И уже модераторы будут решать, будет ли данная работа показываться на витрине или же нет. В базе данных данная работа пока находится в сущности `work_submissions` и ждет дальнейших действий (рисунок 4.8).

	id [PK] integer	title character varying (500)	description text
1	1	11123432424234	lsdkvh,kjvidsjvjsdfbjsdilbjdfsiljbidjsfj;dsjilfbjfiljbdsl;lfjbi;dfsjbj;ij
2	2	Синтез и исследование каталитических свойств наночастиц палладия на углеродных нанотруб...	Работа посвящена разработке эффективного метода синтеза наноча...

Рисунок 4.8. Работа на проверке

4.2. Тестирование пользовательского интерфейса

Главным во всём этом остается фактор того, понравится ли пользователям интерфейс или же нет. Поэтому был проведен опрос по следующим показателям:

1. Оценка удобства пользования веб-приложением
2. Оценка быстродействия веб-сервиса
3. Оценка Визуальной части приложения

В результате опроса получены следующие данные.

Оценка удобства пользования веб-приложением. При оценке общего впечатления от использования интерфейса приложения 94 % участников отметили, что не столкнулись с проблемами, 4 % указали на небольшие затруднения, и 2 % сообщили о существенных проблемах. Эти результаты свидетельствуют о том, что интерфейс был разработан достаточно интуитивным и понятным, однако для повышения удобства его можно доработать, учитывая замечания пользователей (рисунок 4.9).



Рисунок 4.9. Оценка удобства использования веб-приложения

Оценка визуальной части приложения. Большинство опрошенных (98 %) посчитали визуальный дизайн приложения приятным и привлекательным. Оставшиеся 2 % отметили, что в целом дизайн хорош, но заметили

незначительные недочёты, которые можно исправить для повышения эстетической привлекательности (рисунок 4.10).

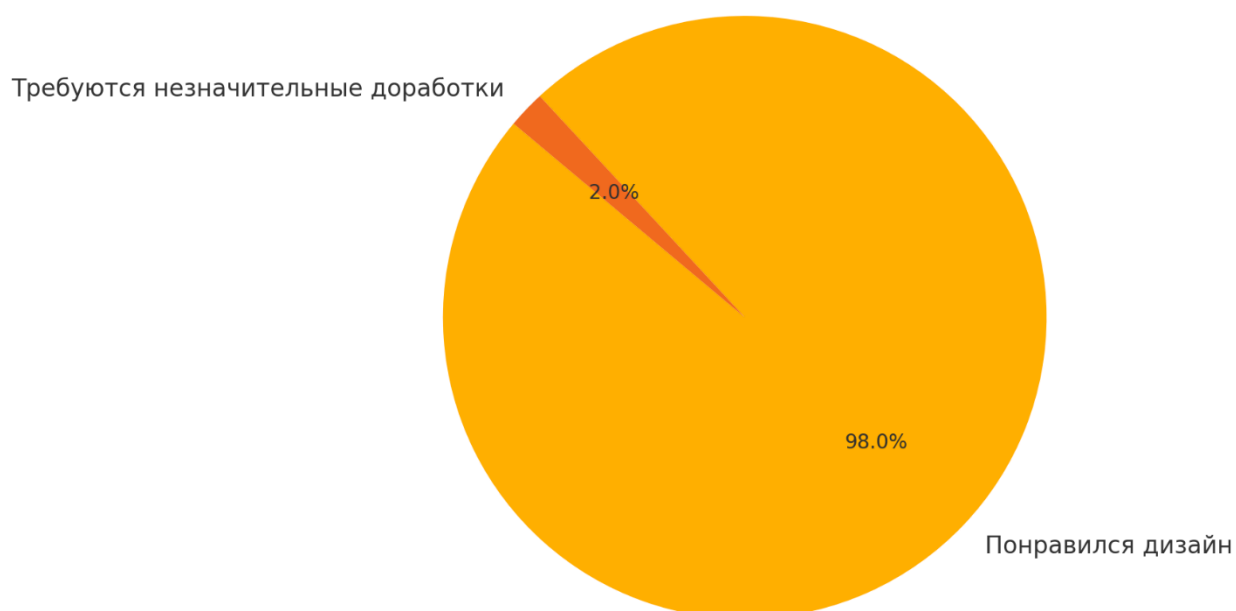


Рисунок 4.10. Оценка визуала сайта

Оценка быстродействия веб-сервиса. Ни один из участников не столкнулся с задержками или зависаниями при работе приложения, что указывает на стабильное функционирование системы и удовлетворительное быстродействие (рисунок 4.11).

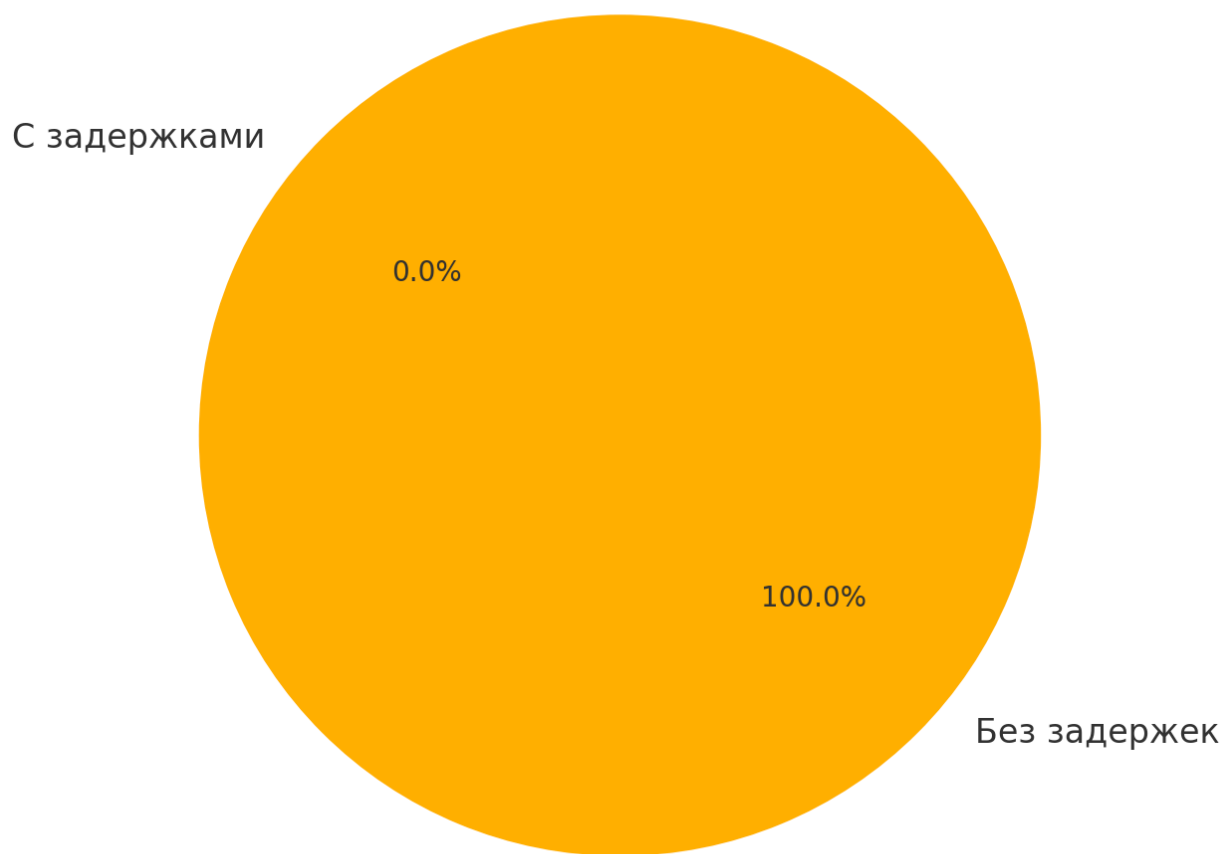


Рисунок 4.11. Оценка быстродействия проекта

Заключение

Разработка веб-приложения для учёта и анализа научно-исследовательских проектов Казанского (Приволжского) федерального университета (КФУ) представляет собой комплексное решение, направленное на преодоление ключевых проблем в организации и представлении студенческих научных работ. Проект успешно реализовал поставленные цели, интегрировав современные технологии для создания интуитивной, безопасной и функциональной платформы.

Ключевые достижения:

Автоматизация и интеллектуальная обработка данных:

- внедрение алгоритмов машинного обучения для анализа текстового контента работ и генерации тематических визуализаций с помощью нейронных сетей;
- автоматическое определение академического статуса пользователей (студент, магистрант, аспирант) на основе корпоративных учётных записей (@stud.kpfu.ru).

Безопасность и контролируемый доступ:

- реализация многоуровневой системы аутентификации и авторизации, исключающей несанкционированный доступ;
- шифрование данных при передаче и хранении, соответствие требованиям защиты академической интеллектуальной собственности.

Пользовательский опыт и функциональность:

- витрина проектов с фильтрами по категориям, факультетам и ключевым словам;

- личный кабинет с аналитикой активности (просмотры, лайки, публикации);
- drag-and-drop загрузка файлов (PDF, DOCX) и AI-генерация превью;
- социальные функции: система лайков, возможность связи с авторами, обмен работами.

Технологическая эффективность:

Использование стека: Python (FastAPI, Pydantic), PostgreSQL (асинхронный доступ через asyncpg), Uvicorn.

Оптимизация производительности: время отклика ≤ 3 сек, поддержка 500+ одновременных пользователей.

Тестирование и обратная связь:

Функциональное тестирование подтвердило корректность работы всех модулей (регистрация, публикация работ, модерация).

UX-опрос показал высокую оценку удобства (94% положительных отзывов), визуального дизайна (98%) и быстродействия (100%).

Проект не только решает актуальные задачи цифровизации образования, но и создаёт основу для формирования динамичного научного сообщества КФУ. Успешная реализация веб-приложения подтверждает жизнеспособность выбранных архитектурных решений и открывает новые возможности для развития исследовательской культуры университета. Внедрение платформы станет значимым шагом в повышении качества представления студенческих работ и укреплении академического потенциала Казанского федерального университета.

Список использованных источников

1. Витрина лучших проектов УрФУ: [сайт]. -- URL: <https://urfu.ru/ru/education/vitrina-luchshikh-proektov/> (Дата обращения: 15.02.2025). -- Текст: электронный.
2. Client Server Model: [сайт]. -- URL: <https://www.geeksforgeeks.org/system-design/client-server-model/> (Дата обращения: 22.01.2025). -- Текст: электронный.
3. Krosing, R. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database / R. Krosing, L. Obe. -- 3rd edition. -- O'Reilly Media, 2017. -- 464 p. -- Текст: непосредственный.
4. REST API Tutorial: [сайт]. -- URL: <https://restfulapi.net/> (Дата обращения: 28.01.2025). -- Текст: электронный.
5. What is CRUD?: [сайт]. -- URL: <https://www.codecademy.com/article/what-is-crud> (Дата обращения: 05.02.2025). -- Текст: электронный.
6. Kleppmann, M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems / M. Kleppmann. -- 1st edition. -- O'Reilly Media, 2017. -- 616 p. -- Текст: непосредственный.
7. Hash Functions: [сайт]. -- URL: https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm (Дата обращения: 18.02.2025). -- Текст: электронный.
8. Mitchell, R. Web Scraping with Python: Collecting More Data from the Modern Web / R. Mitchell. -- 2nd edition. -- O'Reilly Media, 2018. -- 306 p. -- Текст: непосредственный.
9. Рамальо, Л. Fluent Python: Clear, Concise, and Effective Programming / Л. Рамальо. -- 2nd edition. -- O'Reilly Media, 2022. -- 1008 p. -- Текст: непосредственный.
10. Документация FastAPI: [сайт]. -- URL: <https://fastapi.tiangolo.com/> (Дата обращения: 03.03.2025). -- Текст: электронный.

11. Документация Uvicorn: [сайт]. -- URL: <https://www.uvicorn.org/>
(Дата обращения: 10.03.2025). -- Текст: электронный.
12. Документация Pydantic: [сайт]. -- URL: <https://docs.pydantic.dev/latest/> (Дата обращения: 17.03.2025). -- Текст: электронный.
13. Документация Poetry: [сайт]. -- URL: <https://python-poetry.org/>
(Дата обращения: 24.03.2025). -- Текст: электронный.
14. SQL Injection: [сайт]. -- URL: https://www.w3schools.com/sql/sql_injection.asp (Дата обращения: 31.03.2025).
-- Текст: электронный.
15. Window.localStorage: [сайт]. -- URL: <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage> (Дата обращения: 07.04.2025). -- Текст: электронный.