**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Прикладной информатики**

Направление подготовки **09.03.03 Прикладная информатика**

Образовательная программа **Мобильные и сетевые технологии**

**К У Р С О В О Й   П Р О Е К Т**

Тема: «Веб-приложение для поиска учебных материалов “EduMat”»

Обучающийся: Мезенцев Богдан Геннадьевич К3139

Санкт-Петербург 2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ](#_heading=h.gjdgxs) 3

[ОСНОВНАЯ ЧАСТЬ](#_heading=h.1fob9te) 5

[1 Описание проекта](#_heading=h.3znysh7) 5

[2 Этапы работы над проектом](#_heading=h.2et92p0) 7

[3 Мой вклад в проект](#_heading=h.tyjcwt) 9

[4 Анализ своей работы](#_heading=h.3dy6vkm) 13

[5 Взаимодействие с командой](#_heading=h.1t3h5sf) 15

[6 Взаимодействие с руководителем проекта](#_heading=h.4d34og8) 16

[7 Оценка работы руководителя](#_heading=h.2s8eyo1) 17

[ЗАКЛЮЧЕНИЕ](#_heading=h.17dp8vu) 18

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#_heading=h.3rdcrjn) 21

[ПРИЛОЖЕНИЕ](#_heading=h.26in1rg) 22

# ВВЕДЕНИЕ

Идея данного проекта напрямую связана с удобной организацией хранения и поиска информации. Мне и моим однокурсникам уже удалось столкнуться с проблемой после поступления в университет, так как у нас появилось множество разных дисциплин, для каждой из которых есть чаты в телеграмме, созданные преподавателями. В данных чатах содержится множество ссылок, заметок и материалов связанных с конкретной дисциплиной. И уже через пару месяцев учебы становится очень неудобно искать информацию среди огромного потока информации в чатах, так как нет единого места, где будет сохранена вся информация, и многое теряется. Иногда нужную информацию сложно найти из-за отсутствия удобной системы поиска. Поэтому разработка нашего веб-приложения имеет большую актуальность среди студентов и преподавателей. Так как реализация удобной системы поиска и размещения информации на платформе будет упрощать образовательный процесс.

Основной целью данного курсового проекта является разработка веб-приложения, которое обеспечит студентам и преподавателям возможность удобно искать и размещать необходимые учебные материалы по различным дисциплинам.

Для достижения данной цели были выполнены следующие задачи:

1. Разработать пользовательский интерфейс для поиска и просмотра материалов.
2. Разработать модели базы данных для хранения информации о предметах и материалах.
3. Разработать REST API для работы с базой данных.
4. Тестирование и отладка системы

Для реализации серверной части данного проекта использовался язык программирования Python и такие фреймворки, как Django и Django Rest Framework.

# 

# ОСНОВНАЯ ЧАСТЬ

## **1 Описание проекта**

Разрабатываемое веб-приложение представляет собой платформу(сайт), предназначенный для упрощения поиска учебных материалов студентами. В современном образовательном процессе студенты часто сталкиваются с трудностями при поиске необходимых ресурсов, поскольку информация разбросана по различным источникам, а поиск занимает значительное количество времени. Наше приложение решает эту проблему, предоставляя централизованную платформу, где студенты могут быстро и эффективно находить нужные материалы.

Основными пользователями приложения являются студенты различных курсов и специальностей, нуждающиеся в учебных материалах для подготовки к занятиям, экзаменам и другим видам учебной деятельности. Также платформа предусматривает регистрацию преподавателей. Приложение предоставляет им следующие возможности:

* регистрация и авторизация
* добавление своих учебных материалов
* поиск учебных материалов
* добавление необходимых материалов в избранное
* добавление собственного тега (пометки) при публикации материала

Архитектура системы построена по принципу “клиент-сервер”. Frontend, разработанный с помощью фреймворка React, отвечает за пользовательский интерфейс и взаимодействие клиента с пользователем. В роли клиента выступает браузер, с помощью которого пользователь использует нашу платформу. Backend, разработанный на Python с использованием Django и Django REST Framework, отвечает за обработку данных, их хранение в базе данных и предоставление API для доступа к данным. Frontend взаимодействует с Backend через HTTP-запросы. Выбор технологий был обусловлен следующими факторами:

Python. Является популярным языком для веб-разработки, обладает большим количеством библиотек и фреймворков, которые упрощают разработку [1].

Django. Предоставляет мощный инструментарий для создания веб-приложений, включая ORM (Object-Relational Mapping), который упрощает работу с базой данных. Включает в себя много необходимых уже реализованных функций.

Django REST Framework. Позволяет быстро и эффективно создавать RESTful API для взаимодействия frontend и backend. Позволяет быстро и удобно верстать html-страницы.

Postman. Удобство интерфейса – понятный графический интерфейс, в котором несложно разобраться, есть подсветка синтаксиса, возможность создавать и хранить автотесты. Возможность многократного запуска тестов – это помогает поддерживать стабильность и надежность API.

## **2 Этапы работы над проектом**

Работа над проектом была разделена на следующие этапы:

Обсуждение требований к продукту и планирование. Перед началом нашей технической работы мы с командой организовали несколько созвонов, на которых обсуждали, чего конкретно хотим добиться и какие функции должен включать наш проект. Также обсуждали и техническую составляющую: разбивали работу на части и распределяли ее между собой. После распределения всех обязанностей, каждый из нас получил своё задание.

В нашей команде участвовали 3 backend-разработчика, поэтому мы распределили весь объем работы между собой и приступили к выполнению задач. Сначала были спроектированы модели базы данных. Далее мы перешли к изучению предоставленных руководителем требований и документации фреймворков Django и Django Rest Framework. Так как наш репозиторий представлял собой нестандартную структуру проекта Django, а более сложный шаблон проекта со своими особенностями, потребовалось время на изучение документации к данному шаблону. Каждый из нас выполнил приложенную инструкцию, как итог, развернув локальный сервер. После этого, нужно было переходить к реализации распределенных задач - созданию моделей и созданию своих API, для CRUD операций. Результат - каждый разработчик создал свою панель администратора и разработал модель данных, API для CRUD-операций (создание, чтение, обновление, удаление) над моделями.

Разработка Frontend. Целью frontend разработчиков было разработать основные страницы пользовательского интерфейса, который позволяет пользователю взаимодействовать с backend через API. В ходе работы были разработаны страницы регистрации и авторизации пользователя, страница поиска материалов, карточки материала, панель настроек пользователя.

Самой сложной частью работы было связать Backend код и разработанные страницы. На эту часть ушло больше всего времени. Результаты: Разработан пользовательский интерфейс, который взаимодействует с API.

Работа над проектом велась планомерно, с соблюдением сроков каждого этапа. Если кто-нибудь из разработчиков сталкивался с проблемой, то руководитель проекта отвечал на все вопросы и предоставлял документацию, также обсуждение велось и между разработчиками. Для отслеживания прогресса и распределения задач использовались инструменты телеграмм канал проекта и GitLab, где каждый создал свою ветку, в которой был виден прогресс. Общий ход работы можно описать как итеративный процесс, где каждый этап строился на результатах предыдущего, что позволило нам своевременно выявлять и устранять проблемы.

## **3 Мой вклад в проект**

Моей личной задачей было спроектировать и разработать модель базы данных для регистрации и аутентификации пользователей, а также разработать API для взаимодействия с данной моделью. Также было необходимо протестировать разработанную модель для проверки корректной обработки HTTP-запросов я использовал программу Postman, предназначенную для тестирования API.

В ходе работы над данными задачами я изучил:

* официальные документации по работе с библиотеками Django и Django Rest Framework [2]
* основные принципы работы с моделью
* основные принципы работы с операциями CRUD
* способы корректной передачи данных и методы их хранения
* процесс тестирования HTTP-запросов

Я сделал визуализацию модели пользователя, которая представлена на рисунке 1.

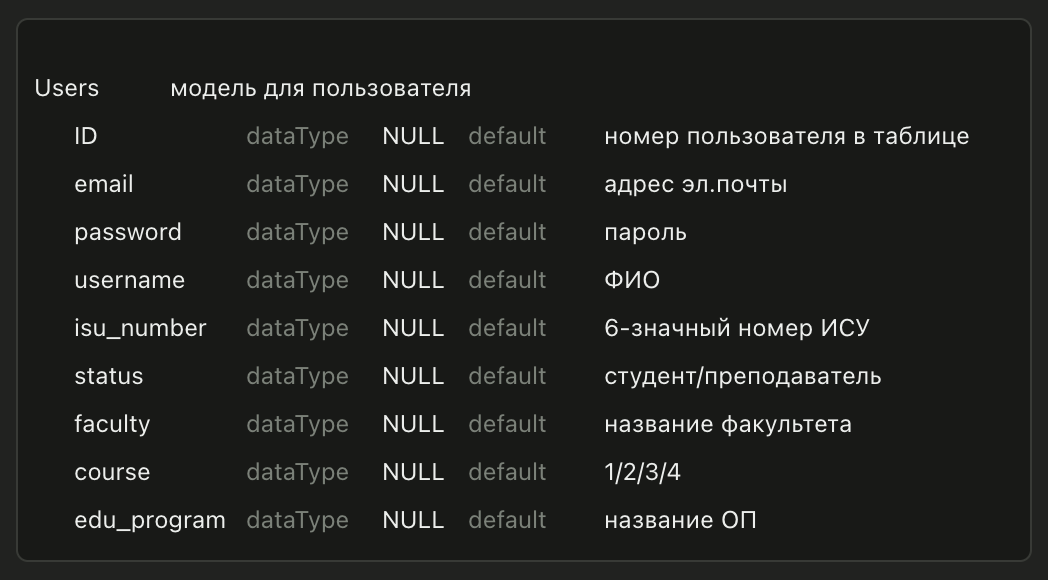


Рисунок 1 - Макет модели пользователя

В Django уже по умолчанию есть модель для регистрации и аутентификации пользователей, которая имеет только три поля: ID, имя пользователя, пароль.

Так как данная модель совершенно не подходит для реализации наших требований, я расширил модель пользователя, добавив необходимые поля, которые хранят данные о студенте или преподавателе. Наша платформа предоставляет возможность регистрации как студентов, так и преподавателей, поэтому при регистрации преподавателя некоторые поля являются необязательными, это было предусмотрено при разработке модели.

Структура нашего проекта немного отличалась от стандартной, но, так как была приложена инструкция со всеми основными шагами, я сделал все поэтапно и мне удалось запустить локальный сервер. Далее я ознакомился с документацией, предоставленной в инструкции, и принялся создавать свою модель [3].

После описания всех полей моей модели я также настроил панель администратора для нее. Для админ панели я указал поля, которые должны в ней отображаться, и их названия. Панель администратора представлена на рисунке 2.

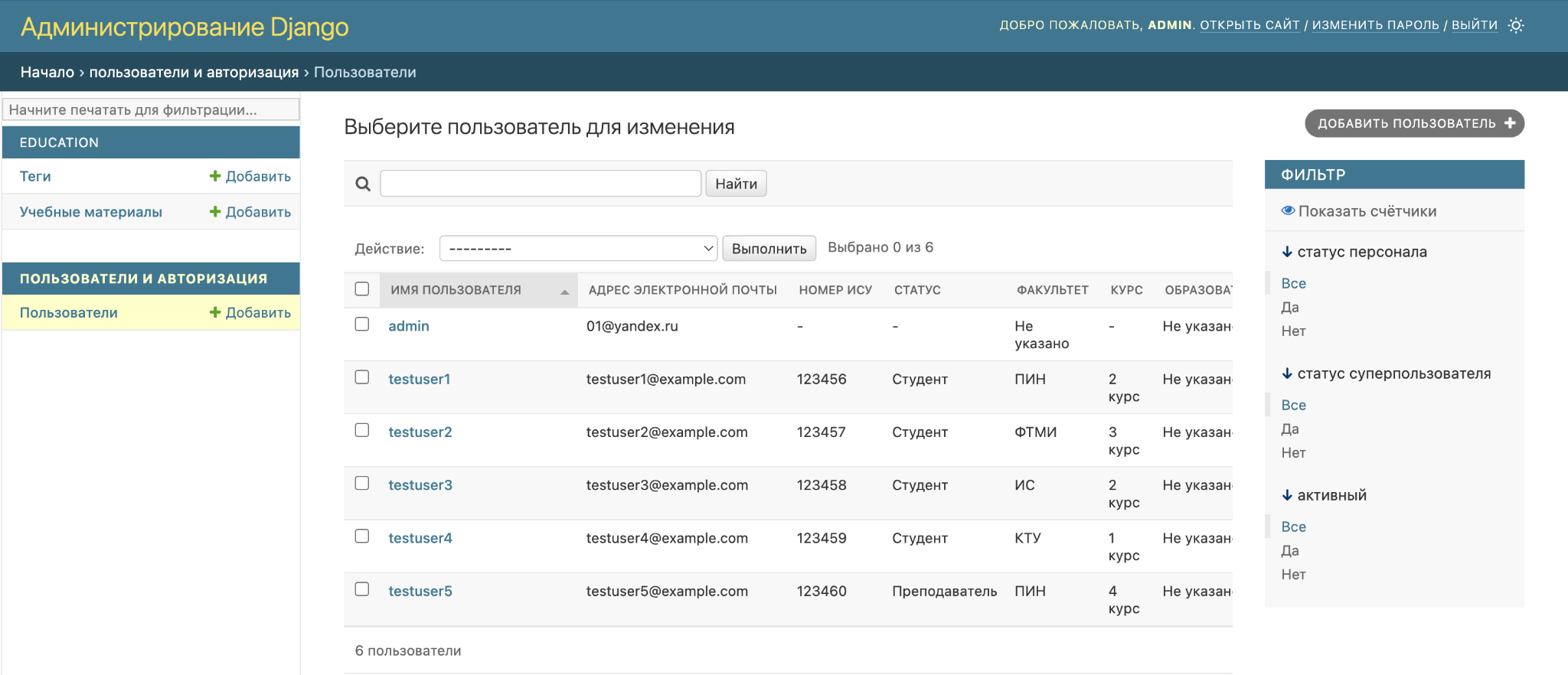


Рисунок 2 - Панель администратора для модели пользователей

Данная панель позволяет выполнять любые действия с аккаунтами уже зарегистрированных пользователей или создавать новые.

После регистрации админ-панели я преступил к разработке API. Необходимо было написать сериализатор для получения данных в нужном формате [4]. Для удобной работы с информацией о пользователе после сериализации мы должны получать данные в формате JSON (JavaScript Object Notation) — текстовый формат обмена данными, который используется для хранения данных и их передачи между различными системами и приложениями. Он основан на синтаксисе объекта в JavaScript, но не привязан к нему. Работу с ним поддерживают многие современные языки программирования: Python, Java и другие. JSON представляет собой удобный для чтения человеком формат, где данные организованы в виде пар «ключ — значение». Ключи — строковые переменные, а значения могут быть строками, числами, булевыми значениями, объектами, массивами или null [5].

После написания API было необходимо протестировать обработку HTTP-запросов [6]. Я провел тест основных видов запросов – GET, POST, PUT. Для этого я использовал программу Postman, предназначенную для тестирования API. Результат успешного тестирования PUT-запроса представлен на рисунке 3.

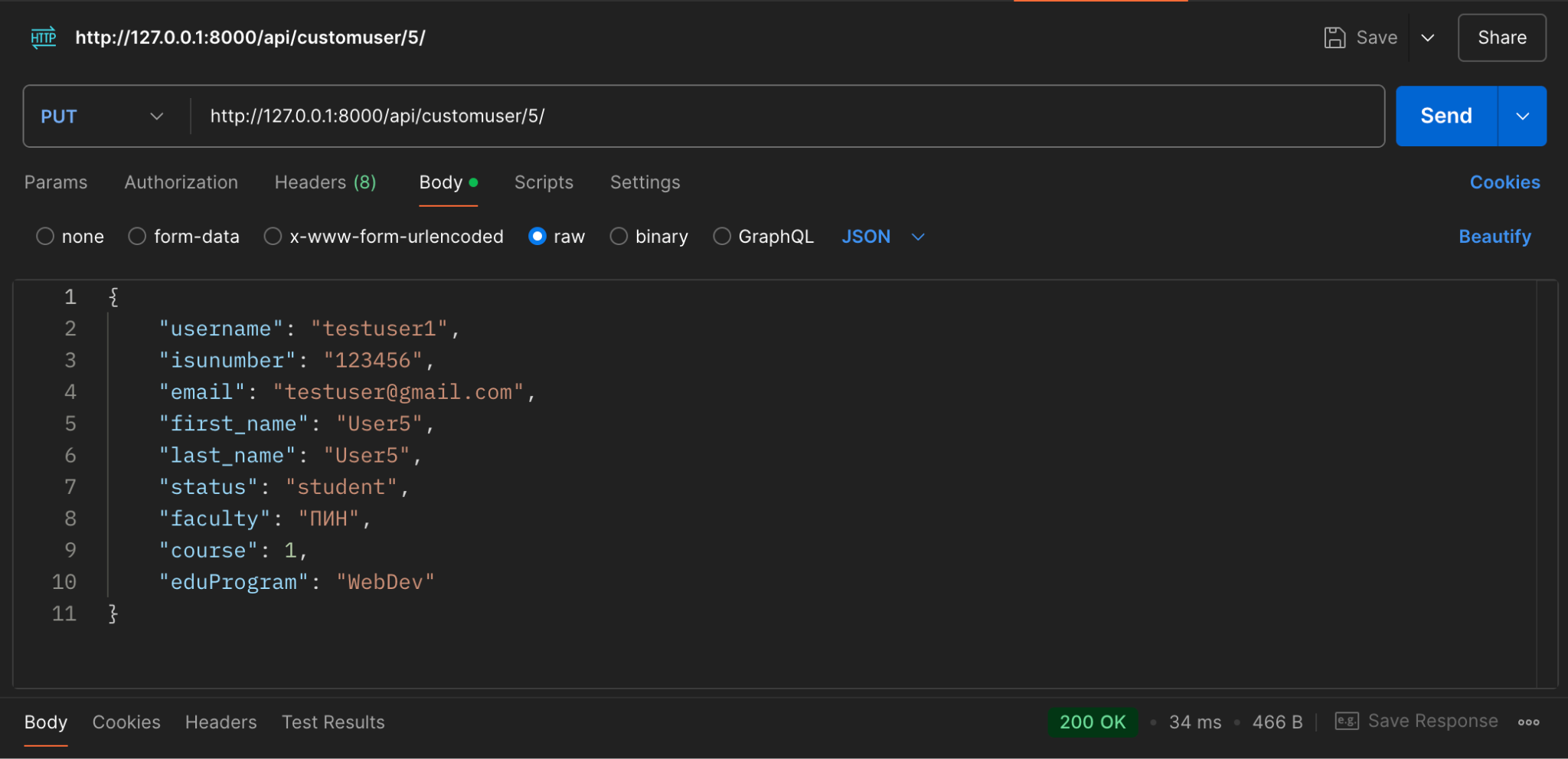


Рисунок 3 - Тестирование HTTP-запроса.

Разработав модель и API к ней, я уточнил у руководителя проекта корректность проделанной работы, получил ответ, что всё правильно и соответствует нужному функционалу, поэтому отправил свою работу в GitLab.

## **4 Анализ своей работы**

Я считаю, что успешно выполнил поставленную задачу. В процессе работы я столкнулся с некоторыми трудностями, которые приходилось решать, узнавая информацию у сокомандников и руководителя, также обращаясь к соответствующим документациям и видео руководствам. Разработанная мной модель и методы работы с ней обеспечивают необходимую функциональность для работы нашего веб-приложения.

Что получилось:

* разработана модель, содержащая в себе все необходимые поля для удобного хранения данных о пользователях
* созданы сериализаторы и десериализаторы для передачи данных
* произведено тестирование работы API
* обеспечена надежная и удобная работа API с высокой производительностью

Основные трудности:

* нехватка опыта работы с шаблонами проектов
* изучение документаций к библиотекам

Я считаю, что результат моей работы полностью соответствует полученному техническому заданию. Решение моих задач шло достаточно планомерно и соответствовало дедлайнам, которые обозначил руководитель. Я использовал GitLab для контроля версий кода нашего проекта и для отслеживания хода работы остальной команды. Иногда после обсуждений проделанной работы с участниками команды и руководителем приходилось исправлять уже написанный код.

Чему я научился:

* принципам проектирования моделей данных в Django
* навыкам создания RESTful API с использованием Django REST Framework
* тестировать HTTP-запросы
* навыкам работы в команде и взаимодействия с другими разработчиками
* навыкам работы с инструментами GitLab
* умению находить и решать сложные технические проблемы
* умению работать с документацией

## **5 Взаимодействие с командой**

Для взаимодействия с командой нашим руководителем был создан общий чат в телеграмме. Для постановки задач и обсуждения проекта были использованы отдельные группы канала. Также руководителем была организована удобная таблица со всеми задачами проекта. На протяжении нашей работы, мы несколько раз созванивались в Google Meet, чтобы обсудить ход работы над проектом, реализованные задачи. В нашем чате мы обменивались вопросами и помогали друг другу. Я был уверен, что в случае чего могу обратиться к товарищам по команде или руководителю проекта, и они обязательно мне помогут. Также мы отслеживали процесс работы с проектом в общем репозиторие в GitLab, где у каждого была своя отдельная ветка, для разработки своей части проекта.

## **6 Взаимодействие с руководителем проекта**

Взаимодействие с руководителем проекта происходило в основном через общем чате в телеграмме, а также в личных сообщениях с ним, где он выписывал задачи, правки, а также отвечал на мои вопросы. Также происходили встречи, на которых обсуждались текущие задачи, прогресс и возникающие проблемы. Руководитель давал ценные советы и рекомендации по проектированию базы данных и API, а также помогал в решении сложных технических вопросов. Также он помогал организовывать процесс работы и предоставил репозиторий проекта, с организованной структурой, и инструкцию, в которой был подробно описан каждый этап разработки.

Руководитель также помогал мне находить новые источники информации, рекомендовал нужные статьи и документацию, которые значительно облегчали процесс работы. Его советы были очень полезны, особенно на начальных этапах проекта, когда я сталкивался с трудностями и выбором правильного подхода к реализации серверной части приложения и способам ее тестирования.

## **7 Оценка работы руководителя**

Я оцениваю работу руководителя проекта как отличную. Он уделил достаточное количество внимания при общении с ним. Он проявил высокий профессионализм, организованность и заинтересованность в успешном выполнении проекта. Руководитель всегда был готов помочь и ответить на мои вопросы, предоставлял необходимую поддержку и ресурсы для решения проблем. Особенно хочу отметить его способность давать своевременные и полезные советы, которые помогали мне находить оптимальные решения. Руководитель был достаточно гибким, чтобы адаптировать план проекта под возникающие обстоятельства, и в то же время был достаточно требовательным, чтобы мотивировать нас на достижение поставленных целей. Также руководитель активно участвовал в подготовке к защите проекта, давал советы по выступлению и наполненности презентации. Он очень верил в нас, и знал, что у нас всё получится, что придавало нам сил во время близкого дедлайна. Работа с таким руководителем была для меня очень полезным и ценным опытом.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта было разработано веб-приложение для поиска учебных материалов “EduMat”, которое позволяет студентам быстро и эффективно находить необходимые материалы для обучения.

Цель проекта, которая заключалась в создании удобной платформы для поиска учебных материалов, считаю достигнутой. Поставленные задачи по разработке backend и frontend части приложения были выполнены, хотя и с некоторыми трудностями, которые мы успешно преодолели, а также некоторой частью нереализованных функций, которые, хоть и не были в ТЗ, но пришли к нам по ходу выполнения работы. Платформа предоставляет основные функции поиска и добавления материалов, просмотра детальной информации о материале и возможность удостовериться в достоверности материала, посредствам появления у материала галочки, если он опубликован преподавателем. Все основные цели проекта были достигнуты, и приложение готово к использованию.

Мой личный вклад в работу для достижения цели проекта заключается в разработке серверной части приложения.

Я отвечал за:

* проектирование и реализацию модели базы данных для пользователей
* разработку RESTful API для взаимодействия с данными о пользователях
* обеспечение корректной работы API с высокой производительностью
* активное взаимодействие с frontend-разработчиками для обеспечения слаженной работы приложения.

В процессе работы над проектом я получил ценный опыт в разработке веб-приложений с использованием Python, Django и Django REST Framework. Я научился:

* проектировать модели данных с учетом, используя реляционные связи и оптимальные типы полей. Освоил работу с миграциями и нормализацией для улучшения структуры и производительности базы данных
* работать с ключевыми компонентами Django REST Framework и создавать API, соответствующие REST-принципам. Реализовывал аутентификацию пользователей
* использовать Postman для проверки работы API, тестирования эндпоинтов и обработки различных сценариев. Освоил настройку коллекций запросов и работу с тестовыми данными.
* получил опыт совместной разработки, участия в код-ревью и обсуждения технических решений
* эффективно использовать GitLab для управления проектом
* работать в команде, взаимодействовать с другими разработчиками
* освоил эффективный поиск информации в официальной документации и сторонних источниках. Научился разрабатывать внутреннюю документацию проекта, включая описание API и архитектуры.

Этот опыт не только расширил мои технические навыки, но и научил меня более эффективно планировать свою работу, правильно распределять время и находить оптимальные решения в сложных ситуациях.

В заключение хочу отметить, что наш проект является успешным примером разработки веб-приложения для решения конкретной проблемы в образовательной сфере. Уверен, что приложение принесет пользу студентам и преподавателям и поможет им в учебном процессе. Для дальнейшего развития проекта можно было бы добавить несколько функций такие как: модерация материалов на платформе, интеграция с другими образовательными платформами, внедрение нашего проекта в учебную систему университета ИТМО.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Python Documentation [Электронный ресурс] – URL: <https://docs.python.org/3/index.html> (дата обращения 24.11.2024).
2. Документация Django [Электронный ресурс] – URL: <https://docs.djangoproject.com/en/5.1> (дата обращения 24.11.2024).
3. Принципы работы с Django проектом [Электронный ресурс] – URL: <https://docs.djangoproject.com/en/5.1/intro/tutorial01/> (дата обращения 24.11.2024).
4. Документация Django Rest-framework [Электронный ресурс] – URL: <https://www.django-rest-framework.org/topics/documenting-your-api/> (дата обращения 2.12.2024).
5. Что такое JSON? [Электронный ресурс] – URL: <https://www.json.org/json-en.html> (дата обращения 2.12.2024).
6. Руководство по работе с Postman [Электронный ресурс] – URL: <https://testengineer.ru/gajd-po-testirovaniyu-v-postman/> (дата обращения 2.12.2024).

# ПРИЛОЖЕНИЕ

Техническое задание на разработку веб-приложения

1. Общие положения

1.1 Название проекта: веб-приложение поиска учебных материалов.

1.2 Цель (назначение): разработка веб-приложения, которое позволит пользователям добавлять, искать и просматривать учебные материалы по различным предметам, улучшая доступность и удобство получения образовательной информации.

1.3 Сроки выполнения: начало - 2024-11-01, конец - 2024-12-20.

1.4 Команда проекта: исполнитель проекта (руководитель проекта) - Прохоров Николай Игоревич, backend разработчики - Мезенцев Богдан, Левахин Лев, Мурашёв Никита, frontend разработчики - Фёдорова Мария, Бай Михаил.

1.5 Этапы задач: TO DO, Development, Review, Done.

1. Технические требования

2.1 Регистрация и авторизация пользователя

2.2 Поиск и просмотр учебных материалов

2.3 Просмотр списка доступных предметов и добавление предметов в избранное

2.4 Фильтрация результатов поиска по категориям

2.5 Модерация материалов через админ панель. Управление пользователями(блокировка, изменение прав доступа)

2.6 Поддержка одновременной работы не менее 1000 пользователей

2.7 Интуитивно понятный и современный дизайн

2.8 Регулярное резервное копирование базы данных

2.9 Сервер с поддержкой Python и Django. База данных PostgreSQL

2.10 Браузеры с поддержкой современных веб-технологий. Поддержка JavaScript и TypeScript.

1. Ожидаемый результат

3.1 MVP системы, отвечающей указанным в ТЗ функциональным требованиям, развернутой на сервере и доступной для пользователей.