**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 поиска мобильных материалов»

Обучающийся: Левахин Лев Александрович К3140

Санкт-Петербург 2024

СОДЕРЖАНИЕ

Введение3

1 Описание проекта4

2 Этапы работы над проектом6

3 Мой вклад в проект8

4 Анализ проделанной работы10

5 Взаимодействие с командой12

6 Взаимодействие с руководителем команды13

7 Оценка руководителя команды14

Заключение15

Список использованных источников17

Приложение. Техническое задание к проекту18

# 

# **ВВЕДЕНИЕ**

В современном образовательном процессе поиск актуальных и качественных учебных материалов является ключевым фактором успешного обучения. За первые несколько месяцев обучения в ИТМО мы столкнулись с проблемой, что у нас есть огромное количество дисциплин, информация о которых публикуется в разных местах. У каждого студента появилось несколько десятков новых телеграмм каналов, сохраненных ссылок, заметок. В таком большом объёме информации очень просто потерять что-то важное, пропустить дедлайн, не найти нужный ресурс или учебный материал. Наше приложение - отличное решение данной проблемы. В нём вся нужная информация была бы собрана воедино, пользователь смог бы быстро искать и сохранять нужную для себя информацию, которая бы не потерялась. По этой причине разработка веб-приложения, которое бы упростило процесс поиска учебных материалов является крайне актуальной задачей.

Основной целью данного курсового проекта является разработка веб-приложения, которое обеспечит студентам удобный и быстрый поиск необходимых учебных материалов по различным предметам. Приложение должно предоставлять возможность искать и добавлять учебные материалы.

Для достижения поставленной цели были решены следующие задачи: разработка модели данных для хранения информации о предметах и материалах; создание REST API для доступа к данным; реализация интерфейса пользователя для поиска и просмотра материалов; тестирование и отладка системы.

Для разработки проекта были выбраны язык программирования Python, фреймворк Django и Django REST Framework.

**1 Описание проекта**

Веб-приложение представляет собой платформу, предназначенную 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. Приложение предоставляет им следующие возможности:

* поиск учебных материалов,
* добавление своих учебных материалов,
* регистрация и авторизация.

Архитектура системы построена по принципу “клиент-сервер”. Frontend, разработанный на React, отвечает за пользовательский интерфейс и взаимодействие с пользователем. 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 с backend. Позволяет быстро и удобно верстать страницы.

**2 Этапы работы над проектом**

Работа над проектом была разделена на следующие этапы:

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

Разработка Backend. В команде было 3 backend разработчика, поэтому мы распределили весь объём работы между собой и приступили к заданию. Первым делом мы разработали визуализацию нашей базы данных, обсудили и доработали её вместе с руководителем проекта. Далее мы перешли к изучению предоставленных руководителем требований и документации к реализации своих моделей Django. Так как наш репозиторий представлял собой не стандартную структуру Django, а более сложный макет со своими особенностями, потребовалось время на его изучение и понимание. Каждый из нас выполнил приложенную инструкцию, как итог, развернув локальный сервер. После этого, нужно было переходить к реализации распределённых задач - созданию моделей и созданию своих API, для CRUD операций. Результат - каждый разработчик создал свой админ аккаунт 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**

Лично передо мной была поставлена задача разработки модели данных для хранения информации об учебных материалах и разработка API для взаимодействия с этой моделью. В ходе работы я изучил новые принципы работы с базой данных и с библиотекой Django + Rest Framework, принцип работы с моделью и CRUD операциями, способами корректной передачи данных и методами их хранения.

На момент начала работы, я понял, что моих знаний может не хватить для нашего проекта, а поскольку проблема действительно актуальная и важная лично для меня - хотелось постараться и выложиться максимально. Я просмотрел курс по работе с Django, в ходе которого освоил основы работы с этой библиотекой и скорректировал своё представление о работе с базой данных [2]. Также я разработал визуализацию модели предмета, которая представлена на рисунке 1.



Рисунок 1 – Модель для предметов и избранного

Далее мне нужно было изучить предоставленный репозиторий. Структура папок и файлов в нём немного отличалась от стандартной, но, так как была приложена инструкция со всеми основными шагами, я сделал всё поэтапно и всё удалось поднять свой локальный сервер и создать свой admin аккаунт. Далее я ознакомился с документацией, предоставленной в инструкции, и принялся создавать свою модель [3].

Основной проблемой было не просто спроектировать модель, которая бы хранила данные, а создать такую структуру, которая была бы гибкой, масштабируемой и позволяла бы легко реализовывать различные функции поиска и фильтрации. Также нужно было написать сериализаторы и десериализаторы, что было для меня в новинку [4]. Я переживал, точно ли сделал всё правильно, потому что в ходе работы можно было пользоваться разными способами реализации предоставления CRUD операций [5]. Я выбрал тот, который показался мне наиболее понятным, и реализовал его. Уточнив у руководителя проекта корректность проделанной работы, получил ответ, что всё правильно и выглядит хорошо, поэтому отправил свою работу в GitLab.

**4 Анализ своей работы**

Я считаю, что успешно справился с поставленной задачей. В процессе работы я столкнулся с большим количеством трудностей, которые приходилось решать, узнавая информацию у сокомандников и руководителя, также обращаясь к соответствующей документации и видео руководствам. Разработанная мной модель и методы работы с ней обеспечивают необходимую функциональность для работы нашего веб-приложения.

Что получилось:

* разработана гибкая и масштабируемая модель данных для хранения информации о предмете,
* созданы сериализаторы и десериализаторы для передачи данных,
* обеспечена надежная и удобная работа API с высокой производительностью.

Основные трудности:

* сложность репозитория проекта,
* нехватка опыта в работе с инструментарием,
* неуверенность в работе кода.

В целом, моя работа над проектом была достаточно планомерной. У нас в телеграмм канале были прописаны задачи и сроки, также и в сообщениях с руководителем. Я использовал GitLab для контроля версий и для отслеживания хода работы остальной команды. Однако, периодически возникали ситуации, когда необходимо было внести изменения в кода или доработать тот или иной метод, что вызывало задержки и пересмотр планов, выделения дополнительного времени.

Чему я научился:

* принципам проектирования моделей данных в Django,
* навыкам создания RESTful API с использованием Django REST Framework,
* 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**

Взаимодействие с командой производилось через телеграмм канал. 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 ветка.

**6 Взаимодействие с руководителем проекта**

Взаимодействие с руководителем проекта происходило в основном через телеграмм канал, а также в личных сообщениях с ним, где он выписывал задачи, правки, а также отвечал на мои вопросы по коду. 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 испытывал затруднения с выбором правильного подхода к реализации backend-части приложения.

**7 Оценка работы руководителя**

Я оцениваю работу руководителя проекта как отличную. 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.

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.

В целом, работа с таким руководителем была очень позитивным и ценным опытом. Возможно, на будущее стоило бы еще более детально прорабатывать сроки и цели каждого этапа проекта для более точного понимания дедлайнов и распределения ресурсов, а также более активно участвовать в соединении работы каждого участника в единое целое.

**ЗАКЛЮЧЕНИЕ**

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

Оценка выполнения всего проекта. 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. Приложение предоставляет основные функции поиска и добавления материалов, просмотра детальной информации о материале и возможность удостовериться в качественности материала, посредствам появления у материала галочки, если он от преподавателя.

Однако, хочу отметить, что некоторые второстепенные функции, такие как сохранение избранных материалов или расширенная система авторизации пользователей, не были реализованы из-за ограниченности времени и ресурсов. Тем не менее, основные цели проекта были достигнуты, и приложение готово к использованию.

Мой личный вклад в достижение цели проекта заключается в разработке backend-части приложения. Я отвечал за:

* проектирование и реализацию моделей данных для хранения информации о предметах, учебных материалах и их авторах,
* разработку REST API для доступа к данным, включая endpoints для CRUD-операций (создание, чтение, обновление, удаление),
* обеспечение корректной работы API с высокой производительностью,
* активное взаимодействие с frontend-разработчиками для обеспечения слаженной работы приложения.

Моя работа обеспечила надежную основу для всего проекта, предоставив backend, который является ключевым для правильной работы и хранения информации. В процессе работы над проектом я получил ценный опыт в разработке веб-приложений с использованием Python, Django и Django REST Framework. Я научился:

* глубже понимать принципы проектирования баз данных и моделирования предметных областей,
* разрабатывать RESTful API с использованием Django REST Framework,
* эффективно использовать GitLab для управления проектом,
* работать в команде, взаимодействовать с другими разработчиками,
* находить и решать сложные технические проблемы.

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 поможет им в учебном процессе. Для дальнейшего развития проекта можно было бы добавить более продвинутые функции фильтрации, пользовательские профили, загрузку файлов самими пользователями и, возможно, интеграцию 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 25.11.2024).
2. Документация Django [Электронный ресурс] – URL: <https://docs.djangoproject.com/en/5.1> (дата обращения 25.11.2024).
3. Принципы работы с Django проектом [Электронный ресурс] – URL: <https://docs.djangoproject.com/en/5.1/intro/tutorial01/> (дата обращения 25.11.2024).
4. Документация Django Rest-framework [Электронный ресурс] – URL: <https://www.django-rest-framework.org/topics/documenting-your-api/> (дата обращения 3.12.2024).
5. Документация по CRUD операциям [Электронный ресурс] – URL: <https://www.geeksforgeeks.org/django-rest-api-crud-with-drf/> (дата обращения 3.12.2024).

**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 - Левахин Лев, Богдан Мезенцев, Никита Мурашёв, frontend разработчики - Мария Фёдорова, Михаил Бай.

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

1.6 Термины и сокращения: API - интерфейс программирования приложений, используется для связи между серверной частью и интерфейсом; SQLite - легковесная база данных, используемая для хранения данных и доступа к ним;

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 МVP системы, отвечающей описанным в ТЗ функциональным требованиям, развернутой на сервере и доступной для пользователей.