**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 учебных материалов»

Обучающийся: Федорова Мария Витальевна К3139

Санкт-Петербург 2025

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_heading=h.30j0zll)

[ОСНОВНАЯ ЧАСТЬ 6](#_heading=h.1fob9te)

[1 Содержание проекта 6](#_heading=h.3znysh7)

[2 Этапы работы над проектом 8](#_heading=h.tyjcwt)

[3 Мои задачи 10](#_heading=h.3dy6vkm)

[4 Работа над проектом 12](#_heading=h.1t3h5sf)

[5 Результаты работы и анализ 13](#_heading=h.4d34og8)

[6 Взаимодействие с командой 16](#_heading=h.2s8eyo1)

[7 Взаимодействие с руководителем проекта 17](#_heading=h.2s8eyo1)

[8 Оценка работы руководителя 18](#_heading=h.2s8eyo1)

[ЗАКЛЮЧЕНИЕ 19](#_heading=h.17dp8vu)

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

[ПРИЛОЖЕНИЕ 22](#_heading=h.4i7ojhp)

# [ВВЕДЕНИЕ](#_heading=h.2xcytpi)

Современные студенты часто сталкиваются с проблемой эффективной организации учебных материалов. Из-за большого объема информации, связанного с множеством предметов, преподавателей и коммуникаций в различных мессенджерах (например, Telegram), становится сложно эффективно систематизировать и найти нужные материалы. В групповых чатах часто скидываются ссылки на лекции, Google-диски, полезные документы и другие ресурсы, но с увеличением объема данных поиск необходимой информации становится трудоемким. Существующие решения, такие как заметки и приложения для задач, имеют свои ограничения — они не поддерживают эффективное совместное редактирование и не позволяют быстро организовать и искать материалы в рамках одного предмета. В результате студенты тратят много времени на поиск нужных данных, что снижает их продуктивность и увеличивает стресс.

Наша разработка направлена на решение этих проблем. Мы предлагаем единое веб-приложение, которое интегрирует все учебные материалы студентов в одном месте, позволяя эффективно управлять информацией, как студентам так и преподавателям, хранить материалы по предметам и обеспечивать быстрый доступ к ним. Это поможет студентам улучшить организацию учебного процесса и повысить продуктивность, а также сделает процесс обучения более удобным и комфортным.

Цель данного проекта заключается в создании единого веб-приложения, которое обеспечит удобное добавление, хранение и поиск учебных материалов по предметам, повышая эффективность учебного процесса и улучшая организацию работы студентов.

В рамках реализации веб-приложения для организации учебных материалов студентов, необходимо решить несколько ключевых задач, которые обеспечат удобство и функциональность работы с системой.

Первая задача заключается в разработке и внедрении страниц, обеспечивающих авторизацию и регистрацию пользователей. Это позволит ограничить доступ к личным данным и материалам, а также создать персонализированный опыт для каждого студента. Для этого необходимо реализовать интерфейс ввода данных для регистрации нового пользователя, а также механизм авторизации для возвращающихся пользователей.

Вторая задача — создание страницы с перечнем всех предметов, с возможностью фильтрации и поиска по названию предмета или преподавателю. Это значительно упростит навигацию, особенно в случае большого объема учебных материалов. Страница должна быть интуитивно понятной, с возможностью сортировки и быстрого поиска нужного предмета.

Третьей задачей является разработка страницы материалов по каждому предмету, где студент сможет просматривать, добавлять и редактировать учебные ресурсы. Эта страница должна поддерживать отображение различных типов контента — ссылок на лекции, видеоматериалов, документов и других файлов, связанных с конкретным курсом. Ключевой функцией этой страницы будет удобный интерфейс для добавления материалов через модальные окна, а также возможность их быстрого поиска и сортировки.

Четвертой задачей является создание личного кабинета пользователя, где будет отображаться ключевая информация о студенте.. Личный кабинет должен включать в себя настройки профиля, возможность редактирования информации через модальное окно и выход из аккаунта.

Пятая задача — разработка функционала для добавления новых предметов и материалов через модальные окна. Пользователи должны иметь возможность добавлять новые предметы и загружать к ним различные типы материалов (ссылки, файлы, заметки), что обеспечит гибкость в использовании приложения.

Решение этих задач обеспечит создание полноценного веб-приложения, которое будет эффективно организовывать учебные материалы, облегчать их поиск и управление, а также создавать условия для совместной работы студентов и преподавателей.

# ОСНОВНАЯ ЧАСТЬ

## 1 Содержание проекта

Проект "Разработка веб-приложения для поиска учебных материалов" фокусируется на усовершенствовании процесса хранения, организации и поиска учебных материалов для студентов. Приложение предоставляет удобный интерфейс и функционал для внесения и обмена материалами по изучаемым предметам.

Задачи, которые предстоит реализовать в работе над проектом:

1) разработка интерфейса на основе UX-UI дизайна: разработка веб-интерфейса, который будет включать в себя страницы авторизации и регистрации пользователя, предметы, материалы, модальные окна для добавления информации и её редактировании, а также личный аккаунт пользователя,

2) разработка React компонентов с использованием библиотеки ShadcnUI: позволяет значительно ускорить процесс создания интерфейса, обеспечивая готовые и стилизованные компоненты, которые легко адаптируются под различные нужды проекта. Библиотека интегрируется с Tailwind CSS, что предоставляет дополнитель,

3) настройка клиент-серверного взаимодействия: один из ключевых этапов, который представляет из себя организацию обмена данными между фронтендом и сервером. Это позволяет пользователю отправлять запросы на сервер и получать ответы, например, при авторизации или получении информации, обеспечивая безопасность и стабильную работу этого процесса, чтобы данные передавались корректно,

4) проектирование моделей баз данных: обеспечивает структурированное и эффективное хранение данных. Модель пользователей будет включать информацию о каждом пользователе, такую как ФИО, ИСУ, email, курс, пароль и имя пользователя, что необходимо для авторизации и персонализации. Модель предметов будет хранить данные о предметах, такие как название, описание, имя преподавателя, название программы и номер курса. Модель тегов позволит удобно организовать и классифицировать материалы, связывая их с конкретными предметами и обеспечивая легкий доступ через поиск и фильтрацию,

5) регистрация моделей в панели администратора Django: будут созданы специальные классы администратора, которые связывают модели с интерфейсом администрирования. Это позволит настроить отображение данных, фильтрацию, поиск и сортировку, а также ограничить доступ к определенным функциям. Регистрация моделей в админке значительно упрощает процесс работы с данными, предоставляя администраторам интуитивно понятный интерфейс для их управления и редактирования,

6) развертывание и поддержка: позволит реализовать взаимодействие между клиентом и сервером, предоставляя данные в формате JSON. Создание API с помощью Django REST Framework гарантирует простоту и гибкость в управлении данными.

В итоге реализация этого проекта значительно улучшит и облегчит образовательный процесс в университете, обеспечив более эффективным процессом обмена материалами или какой-либо другой информацией по изучаемым предметам. Пользователи смогут легко находить, создавать и делиться материалами. Создание этого приложения увеличит шанс того, что у студентов будет меньше потерянной важной информации, нежели в групповых чатах Telegram. После успешного тестирования и развертывания на серверах, поддержка и обновления будут обеспечивать стабильную работу и добавление новых функций в будущем, делая веб-приложение популярнее среди студентов ITMO.

## 2 Этапы работы над проектом

Работа над проектом была разделена на несколько ключевых этапов.

Для начала мы с командой на созвоне обсудили требования к проекту, определили его функциональные особенности и распределили задачи, поставленные перед frontend и backend разработчиками. После созвона каждый из нас получил свое задание, что позволило приступить к реализации проекта.

Frontend-разработка. Над frontend частью работали два человека. Для начала мы ознакомились с документацией, которую прислал нам руководитель проекта, изучили предоставленные руководителем требования и ознакомились с UX-UI дизайном. Затем мы инициализировали проект на React с TypeScript, установили, настроили все необходимые библиотеки, в том числе библиотеку ShadcnUI, заранее изучив ее, и настроили структуру самого проекта. Далее мы добавили изменения на GitLab, где находился репозиторий, каждый создал свою ветку для дальнейшей разработки и приступил к созданию интерфейса и реализации функциональности своей части проекта. После выполнения своих задач в создании интерфейса мы получили: страницу авторизации и регистрации пользователя, предметов, материалов и модальные окна для добавления и обновления информации. Затем соединили сделанную работу воедино и приступили к настройке функциональности приложения и клиент-серверного взаимодействия, так как другие члены команды, которые занимались backend-разработкой на тот момент доделали свою работу. После добавили маршрутизацию, расположили последние изменения в репозитории и получили итоговый результат.

Backend-разработка. Перед backend разработчиками стояли следующие задачи: во-первых это инициализация проекта на Django; во-вторых проектирование моделей баз данных, регистрация моделей в панели администратора, а также создание API для моделей с помощью Django REST Framework.

Работа над проектом была структурирована по этапам с соблюдением сроков. Весь процесс был итеративным: каждый этап основывался на результатах предыдущего, что способствовало своевременному выявлению и устранению проблем. Для отслеживания прогресса и управления задачами использовались, созданный нами Telegram-чат и GitLab, где каждый разработчик работал в своей ветке, что позволило легко контролировать выполнение работы. В случае возникновения вопросов или проблем у разработчиков, руководитель быстро предоставлял ответы и необходимую документацию, а также проводил обсуждения в команде.

## 3 Мои задачи

В процессе формирования проекта и распределения ролей в команде возникла необходимость четко определить обязанности каждого участника. Учитывая мой опыт работы с frontend-разработкой на React, я решила взять на себя роль фронтенд-разработчика, отвечающего за разработку клиентской части приложения. После обсуждения требований и распределения задач в команде, мне были назначены следующие ключевые обязанности.

Создание функционала для авторизации и регистрации пользователей. Важной частью любого веб-приложения является система аутентификации и авторизации пользователей. Моя задача заключалась в реализации страниц для регистрации нового пользователя и авторизации для уже существующих. Регистрация пользователя: создание формы, в которой пользователи могут ввести свои данные (ИСУ, номер курса, имя и фамилию, пароль, email ), с последующей валидацией этих данных. Авторизация: создание страницы для входа, где пользователь вводит свои данные для доступа к своему аккаунту. Для этого необходимо было реализовать механизм проверки введенных данных и их соответствие данным, хранящимся на сервере. Валидация данных: использование библиотеки для валидации, такой как Zod, чтобы убедиться, что все введенные данные соответствуют необходимым требованиям (например, правильный формат email, достаточная длина пароля и т.д.).

Создание страницы предметов. На главной странице приложения должна быть представлена информация о всех доступных предметах. Задача заключалась в отображении списка предметов с возможностью фильтрации и сортировки, что позволяет пользователю легко ориентироваться в приложении. А также нужно было сделать модальное окно с добавлением нового предмета, в которое пользователь вписывает ключевую информацию о предмете, тем самым создавая новый предмет.

Создание личного аккаунта. Пользователи должны иметь возможность управлять своим аккаунтом, изменять свои данные или выходить из системы. Для этого была задача разработать функциональность личного кабинета с возможностью редактирования профиля, а также кнопка выхода из аккаунта, чтобы давать пользователю полный контроль над своими данными.

Настройка маршрутизации всего проекта. Важной частью проекта является правильная навигация между различными страницами. Задача состояла в настройке маршрутизации, которая обеспечивала бы переход между страницами, а также защищенные маршруты для авторизованных пользователей, обеспечивая безопасность доступа к личным данным.

Интеграция и настройка клиент-серверного взаимодействия. Для корректной работы приложения было необходимо наладить эффективное взаимодействие между клиентом и сервером, что включало в себя обработку запросов и передачу данных. Этот процесс обеспечивал синхронизацию данных на разных уровнях и корректную работу всех функций приложения.

Таким образом, эти задачи составляли основную часть моей работы, которые представляли собой разработку клиентской части приложения, включая создание интерфейса для пользователей, реализацию функционала взаимодействия с сервером и обеспечение удобной навигации по проекту.

## 4 Работа над проектом

Перед началом выполнения своих задач, несмотря на опыт во frontend-разработке на React с использованием языка программирования JavaScript, мне нужно было изучить TypeScript [1], а также познакомиться с библиотекой ShadcnUI [2]. После прочтения большого количества статей и просмотра обучающих роликов в интернете я приступила к выполнению своих задач.

В ходе работы над проектом я сделала большую часть задач без всяких проблем, создавая React-компоненты [3] и используя компоненты для создания интерфейса, используя библиотеку ShadcnUI [2]. Для стилизации компонентов на странице я использовала файлы css*,* брав данные из макета с дизайном в Figma. Также были моменты, когда я помогала второму frontend-разработчику, так как у него не было опыта в работе над frontend. Но, доделывая свои последние задачи, я и сама столкнулась с проблемой реализации взаимодействия серверной и клиентской части приложения. Поэтому после создания интерфейса я вновь погрузилась в изучение материала для устранения проблемы, читая статьи на Habr [6]. После полученной информации, смогла реализовать оставшиеся задачи, включая проверку на валидацию вводимых данных c использованием библиотеки Zod [5], маршрутизацию, используя библиотеку React Router [4] и то, с чем я не смогла справиться раньше - настройку клиент-серверного взаимодействия. Затем я созвонилась с руководителем, чтобы он посмотрел логику реализации взаимодействия frontend и backend. Руководитель сказал, что сделанно всё правильно, поэтому я отправила итоговые изменения на GitLab, тем самым завершая работу команды над проектом.

## 5 **Результаты работы и анализ**

В ходе разработки клиентской части был сделан отдельный репозиторий на GitHub, представленный на рисунке 1. Также мной были выполнены следующие работы:

В первую очередь создание интерфейса основных страниц приложения таких, как «Вход» (рис.2), «Регистрация» (рис.3), «Предметы» (рис.4), «О себе», модальные окна добавления предмета (рис.5) и редактировании информации в разделе «О себе» .

Далее, важным этапом стало не только создание интерфейса, но и настройка функциональности приложения, например, добавление предмета, валидация введенных пользователем данных. Также было реализовано создание маршрутизации в проекте и настройка связи frontend и backend.

В ходе работы над проектом я не только получила новый опыт во frontend-разработке, но и изучила язык TypeScript, познакомилась с библиотекой ShadcnUI, применив это всё на практике. Также я впервые делала настройку клиент-серверного взаимодействия и реализовывала логику проверки входных данных при авторизации. Стоит отметить, что также я, благодаря руководителю команды, научилась видеть, в каких местах можно реализовать код проще и менее объемнее, что пригодиться мне в будущем. Я справилась со всеми своими задачами, сделав всё правильно и в срок.

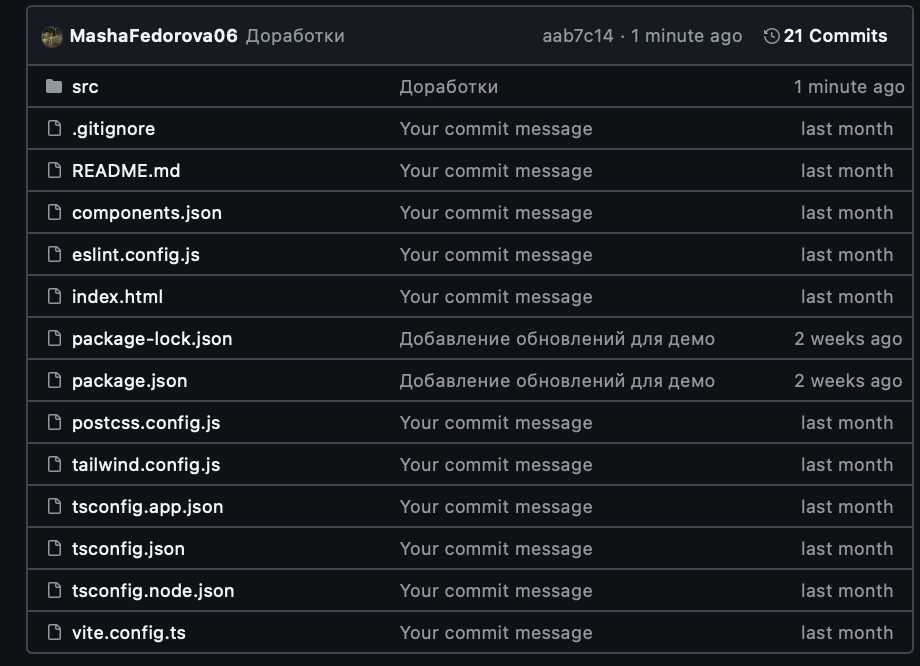


Рисунок 1 – GitHub

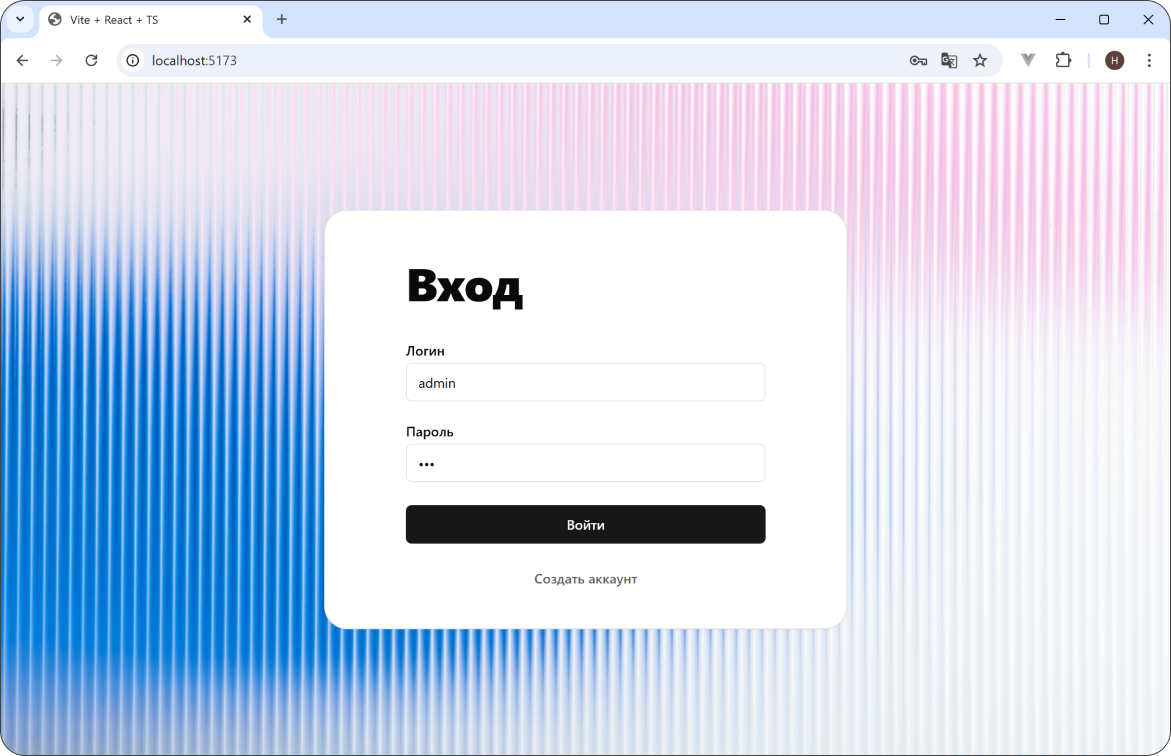


Рисунок 2 – Страница Вход

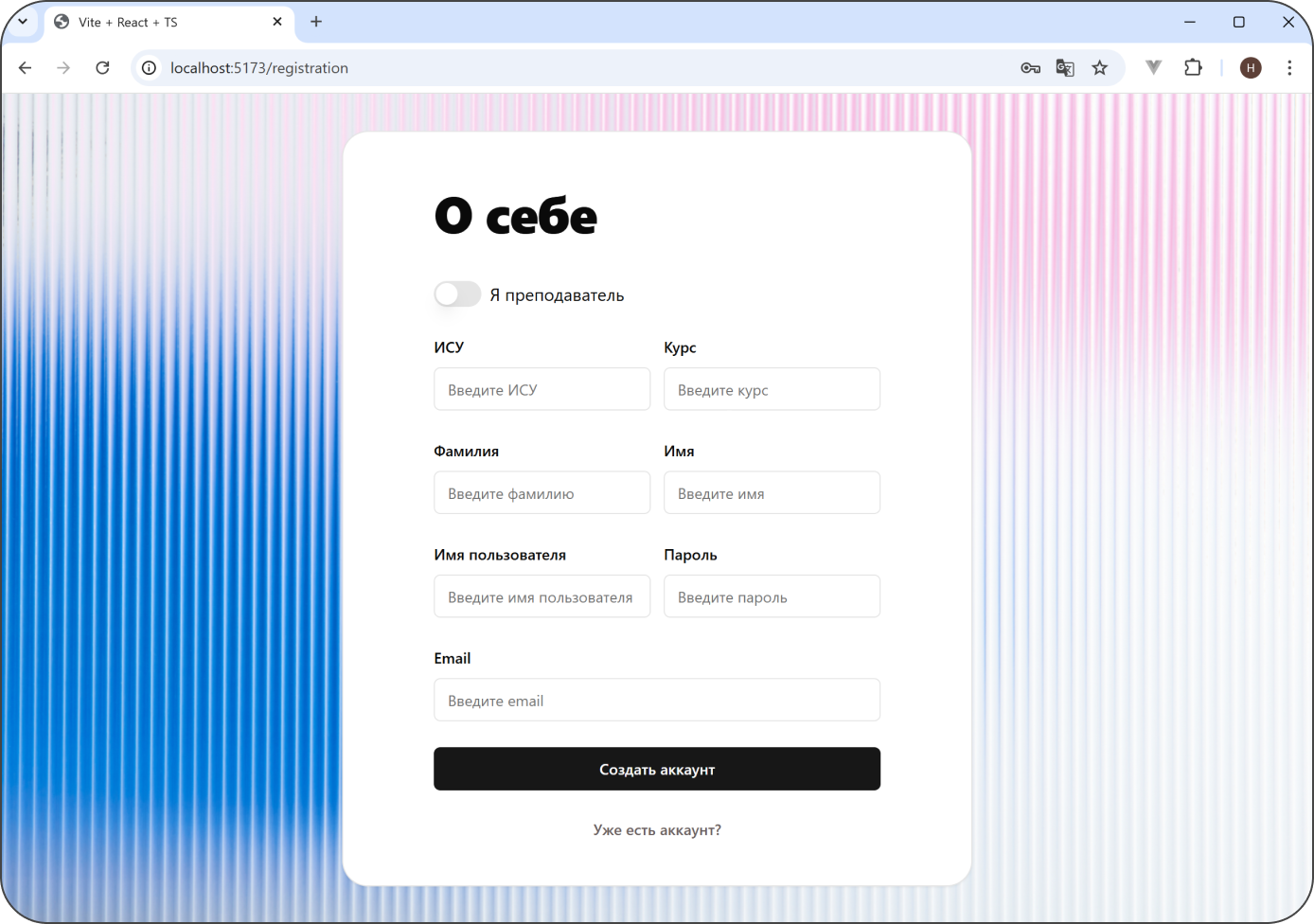


Рисунок 3 – Страница Регистрации

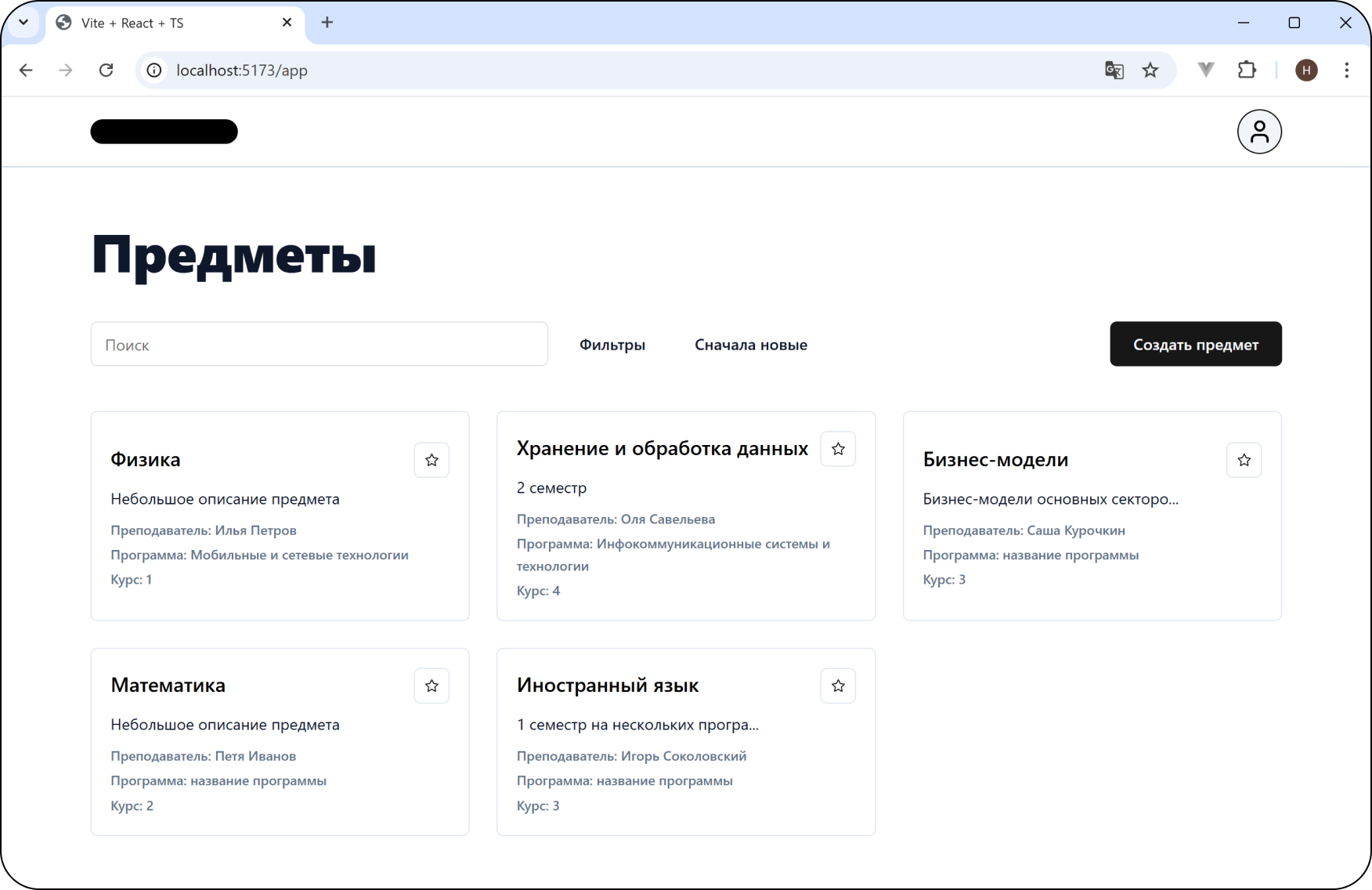


Рисунок 4 – Страница Предметы

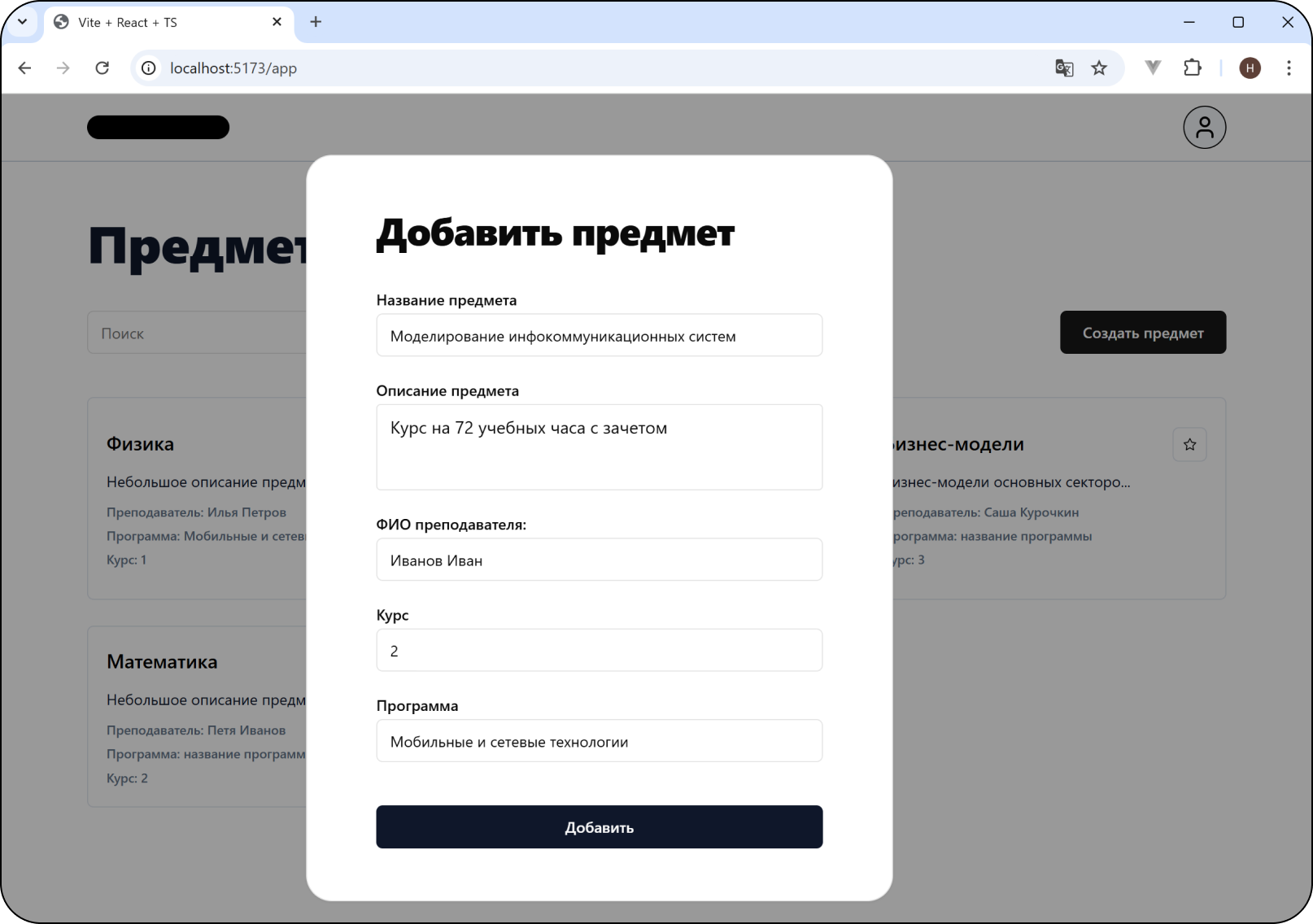


Рисунок 5 – Модальное окно добавление предмета

## 6 Взаимодействие с командой

Командное взаимодействие происходило через Telegram-чат, который служил основным каналом для постановки задач и обсуждения всех аспектов проекта. В чате были созданы отдельные группы для различных тем, таких как общие вопросы, задачи по проекту, а также для обсуждения технических деталей. Руководитель проекта организовал удобную таблицу для отслеживания всех задач, где были указаны сроки, приоритеты и ответственные. Это помогало всем членам команды четко понимать свои обязанности и следить за состоянием задач.

Для более детального обсуждения текущих задач и обмена обратной связью мы регулярно проводили встречи в Google Meet. На этих встречах обсуждались текущий прогресс, завершенные задачи, а также возникающие сложности и возможные пути их решения. Это позволило поддерживать высокий уровень координации и оперативности в команде, а также своевременно корректировать подходы в случае возникновения проблем.

В процессе работы в чате мы активно обменивались вопросами, предложениями и поддерживали друг друга. В случае возникновения трудностей каждый мог обратиться за помощью как к коллегам, так и к руководителю, которые всегда готовы были предложить решение. Это создавало атмосферу взаимопомощи и поддержки в команде, что значительно ускоряло процесс работы и снижало стрессовые ситуации.

Кроме того, весь процесс разработки отслеживался в общем репозитории GitLab. Каждый разработчик работал в своей ветке, что позволяло легко интегрировать изменения и поддерживать чистоту кода. В репозитории также велся подробный журнал коммитов. Такая структура работы обеспечивала хорошую организацию и синхронизацию работы всех участников проекта.

## 7 Взаимодействие с руководителем проекта

Как только команда была сформирована, руководитель проекта, Николай, оперативно вышел на связь и предоставил нам техническое задание, в котором были изложены основные требования, цели и задачи проекта. Затем мы провели встречу, на которой детально проработали этапы реализации и распределили задачи между участниками. Николай объяснил, как должны быть организованы работы на каждом этапе, какие приоритеты стоит учитывать и какие сроки нужно соблюдать. Это позволило всем членам команды четко понять свою роль и выработать общий план действий.

На протяжении всей работы Николай активно направлял нас, контролировал выполнение задач и оценивал промежуточные результаты. Он регулярно проводил ревью, помогал решать возникающие вопросы и давал рекомендации по улучшению качества работы. Если возникали трудности, Николай всегда был готов оперативно предоставить нужные указания и поддержку, что позволяло быстро находить решения и продолжать двигаться вперед.

Николай также поддерживал атмосферу открытости и прозрачности в коммуникациях, всегда оставался доступным для вопросов и запросов, своевременно предоставляя обратную связь. Это создавало доверие и взаимопонимание в команде, помогая нам оставаться на одной волне и уверенно двигаться к выполнению целей проекта.

## 8 Оценка работы руководителя

Мне было приятно работать с нашим руководителем. Он проявил высокий уровень вовлеченности и организованности на всех этапах работы. В начале проекта Николай предоставил четкое техническое задание и организовал встречу, на которой подробно объяснил структуру работы, приоритеты и сроки. Его подход позволил каждому из нас ясно понять свои обязанности и сформировать общий план действий.

Кроме того, Николай проявил себя как активный куратор проекта, помогая команде преодолевать трудности и контролируя качество выполнения задач. Его способность оперативно предоставлять нужные указания и поддерживать команду создала благоприятную атмосферу для работы. В результате этого подхода мы смогли эффективно организовать процесс разработки и поддерживать высокий уровень качества на всех этапах проекта.

Особенно стоит отметить, что помимо непосредственной работы над проектом, Николай также оказал помощь в подготовке к защите. Он активно участвовал в создании презентации, помогал составить речь и давал советы по ее представлению. Благодаря его поддержке мы чувствовали уверенность и не переживали перед защитой.

С учетом всей работы, вовлеченности и поддержки, которую Николай оказывал на протяжении всего проекта, я бы оценила работу руководителя на максимальный балл.

# ЗАКЛЮЧЕНИЕ

В процессе работы над курсовым проектом было разработано веб-приложение, предназначенное для поиска учебных материалов. Оно предоставляет студентам удобный интерфейс для быстрого доступа к необходимым учебным ресурсам, таким как ссылки на лекции, Google-диски и тд. Приложение позволяет эффективно искать, фильтровать и сортировать материалы по различным критериям, что значительно упрощает процесс поиска информации.

Оценка выполнения проекта в целом показывает, что цель создания удобной платформы для поиска учебных материалов была достигнута. Задачи по разработке backend и frontend частей приложения были выполнены, хотя и с определенными трудностями, которые мы успешно преодолели. В ходе работы над проектом возникли идеи по созданию новых функций, они не были включены в техническое задание и не были реализованы, поэтому в ближайшем будущем мы планируем добавить их в наш продукт для его улучшения. На данный момент приложение предлагает основные возможности: поиск и добавление предметов, просмотр подробной информации о них, соответственно сами материалы по предмету с возможностью их фильтрации. Однако также стоит отметить, что некоторые второстепенные функции, такие как сохранение избранных материалов и расширенная система авторизации пользователей, не были реализованы из-за ограничений по времени и ресурсам. Но не смотря на это, в целом, основные цели проекта достигнуты, и приложение готово к использованию.

В рамках выполнения курсовой работы я активно участвовала в разработке клиентской части веб-приложения. Моей основной задачей было создание интерфейса основных страниц таких, как авторизация и регистрация пользователя, страница предметы и личный аккаунт студента, и модальных окон для добавления и редактирования предмета и личного аккаунт, а также настройка клиент-серверной части приложения, его функциональности и маршрутизации. В процессе работы над frontend были использованы современные технологии, такие как React c TypeScript, и библиотеки ShadncnUI, React Router, Zod, что обеспечило эффективность и удобство в разработке клиентской части приложения.

Важной частью проекта стала эффективное разработка клиент-серверного взаимодействия. Регулярное взаимодействие с другими участниками проекта и с руководителем, а также активное использование открытых информационных источников и документации помогло в эффективном решении возникающих задач. Итоговый продукт представляет собой надежное и производительное решение, интегрированное с backend-частью проекта.

В ходе выполнения проекта я активно участвовала в обсуждениях, что существенно способствовало успешной реализации определенных этапов работы. Мой вклад оказал значительное воздействие на прогресс в достижении поставленных задач, что в итоге сыграло важную роль в общем успехе команды. Без слаженной работы с коллегами и взаимодействия в команде, ничего бы не получилось, и это понимание содействовало достижению немалых результатов в рамках проекта.

В ходе работы над проектом были приобретены и улучшены навыки в области frontend-разработки, что отразится на качестве и эффективности моих будущих проектов. Я получила очень ценный опыт в командной работе, смогла понять, как происходит работа над реальным продуктом.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Язык программирования TypeScript [Электронный ресурс] – URL: <https://skillbox.ru/media/code/typescript_kak_s_nim_rabotat_i_chem_on_otlichaetsya_ot_javascript/> (дата обращения 17.11.2024).
2. Официальный сайт библиотеки ShadcnUI с документацией [Электронный ресурс] – URL:

<https://ui.shadcn.com/> (дата обращения 18.11.2024).

1. Официальный сайт библиотеки React c документацией [Электронный ресурс] – URL:

<https://ru.legacy.reactjs.org/> (дата обращения 14.11.24).

1. Документация библиотеки React Router [Электронный ресурс] – URL:

<https://reactrouter.com/> (дата обращения 28.11.2024).

1. Документация библиотеки Zod [Электронный ресурс] – URL:

<https://zod.dev/> (дата обращения 4.12.2024).

1. Статья с информацией о настройке клиент-сервного взаимодействия [Электронный ресурс] – URL: <https://habr.com/ru/articles/495698/> (дата обращения 10.12.2024).

# ПРИЛОЖЕНИЕ

Техническое задание на разработку веб-приложения

**1 Название проекта**

Веб-приложение поиска учебных материалов.

**2 Цель (назначение)**

Разработать веб-приложения, которое позволит студентам ITMO добавлять, искать и просматривать учебные материалы по различным предметам, улучшая доступность и удобство получения образовательной информации.

**3 Сроки выполнения**

Начало 1 ноября 2024 г. Окончание 20 декабря 2024 г.

**4 Руководитель проекта**

Прохоров Николай Игоревич, студент К4240, факультет ИКТ

**5 Термины и сокращения**

API — интерфейс программирования приложений, используется для связи между серверной частью (бэкенд) и интерфейсом (фронтенд).

SQLite — Легковесная база данных, используемая для хранения данных в приложении.

**6 Технические требования**

6.1 Регистрация и авторизация пользователя

6.2 Поиск и просмотр учебных материалов

6.3 Просмотр списка доступных предметов и добавление предметов в избранное

6.4 Фильтрация результатов поиска по категориям

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

6.6 Поддержка одновременной работы не менее 1000 пользователей

6.7 Интуитивно понятный и современный дизайн

6.8 Регулярное резервное копирование базы данных

6.9 Сервер с поддержкой Python и Django. База данных PostgreSQL

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

**7 Содержание работы (этапы по срокам, можно в таблицу)**

Таблица 1 - Этапы задач проекта

| Этапы задач |
| --- |
| TO DO |
| Development |
| Review |
| Done |

Таблица 2 - Задачи

**8 Основные результаты работы**

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