МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра информационных технологий управления

**Курсовая работа**

Веб-приложение для учета успеваемости студентов

Направление 09.03.02 «Информационные системы и технологии»

Профиль «Информационные технологии управления»

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Тарасов, ст. преподаватель

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_К.С. Семенихин, 3 курс, д/о

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Еремеев, 3 курс, д/о

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.Е. Парфенова, 3 курс, д/о

Воронеж 2023

Содержание

[Введение 3](#_Toc146660518)

[1 Постановка задачи 4](#_Toc146660519)

[2 Анализ предметной области 5](#_Toc146660520)

[2.1 Глоссарий 5](#_Toc146660521)

[2.2 Цели создания веб-приложения 6](#_Toc146660522)

[2.3 Сфера применения 7](#_Toc146660523)

[2.4 Требования к веб-приложению в целом 7](#_Toc146660524)

[2.5 Пользователи системы 7](#_Toc146660525)

[3 Анализ задачи 9](#_Toc146660526)

[3.1 Диаграмма прецедентов 9](#_Toc146660527)

[3.2 Диаграмма последовательностей 9](#_Toc146660528)

[3.3 Диаграмма состояний 10](#_Toc146660529)

[3.4 Диаграмма активностей 11](#_Toc146660530)

[4 Реализация 13](#_Toc146660531)

[4.1 Разработка frontend 13](#_Toc146660532)

[4.2 Разработка backend 14](#_Toc146660533)

[5 Навигация по приложению 16](#_Toc146660534)

[5.1 Экран авторизации 16](#_Toc146660535)

[5.2 Экран администратора 17](#_Toc146660536)

[5.3 Экраны добавления данных 18](#_Toc146660537)

[Заключение 19](#_Toc146660538)

[Список использованных источников 21](#_Toc146660539)

Введение

В современной образовательной среде учет успеваемости студентов является важным элементом работы факультета. Это позволяет контролировать процесс обучения и делать выводы о его эффективности. В данной курсовой работе будет рассмотрена задача разработки системы учета успеваемости студентов на факультете.

В данной курсовой работе рассмотрен процесс разработки веб-приложения для отслеживания личной успеваемости в университете, которое будет условно бесплатным и удобным в использовании. Для реализации данного приложения будут использованы современные технологии веб-разработки, такие как React, Python, PostgreSQL.

1. Постановка задачи

Задачей данной курсовой работы является разработка веб-приложения для учета успеваемости. Онлайн-сервис позволит отслеживать баллы по каждому предмету, которым обучается студент.

Следует реализовать следующие пункты:

* Добавление пользователя-студента со стороны администратора;
* Добавление пользователя-преподавателя со стороны администратора;
* Авторизация;
* Просмотр баллов по аттестациям;
* Просмотр баллов по экзаменам;
* Возможность подписывания студента на новый курс.

1. Анализ предметной области
   1. Глоссарий

* Авторизация – это процесс проверки прав пользователя на осуществление определенных действий на сайте;
* Студент – человек, который обучается на факультете;
* Успеваемость – оценка знаний студента по определенным предметам за определенный период времени;
* Средний балл – среднее значение оценок по всем предметам за определенный период времени;
* Факультет – структурное подразделение университета, где проводится обучение по определенным специальностям и курсам;
* База данных – это упорядоченный набор структурированной информации или данных, которые хранятся в электронном виде в компьютерной системе;
* Веб-приложение, веб-сервис, интернет-сервис, онлайн-сервис, проект – это программное обеспечение, которое размещено на удаленном сервере и доступно через браузеры в интернете;
* Личный кабинет, профиль – это раздел сервиса, в котором пользователь может получить доступ к персональным данным;
* Развертывание – это все действия, которые делают систему готовой к использованию;
* Сервер – это отдельный класс компьютерных устройств, предназначенных для обработки запросов от различных узлов сети;
* СУБД – это система управления базы данных;
* Фреймворк – это программная среда, облегчающая разработку и объединение разных компонентов большого программного проекта;
* Хостинг – это услуга, предоставляемая провайдером, которая позволяет размещать веб-сайты, приложения или другие ресурсы на удаленных серверах, обеспечивая доступ к ним через интернет;
* Frontend – это клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса;
* Backend – это программно-аппаратная часть сервиса;
* REST API – это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов;
* React – это JavaScript библиотека для создания пользовательских интерфейсов;
* SPA – это одностраничное веб-приложение, которое работает на одной HTML-странице, обновляя данные на ней;
* GitHub – это крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки;
* Header – это верхняя часть веб-страницы, которая содержит информацию о его содержании или навигационные элементы;
* HTTP – это протокол передачи данных, используемый для обмена информацией между клиентом и сервером в сети интернет;
* HTTPS – это защищенная версия протокола HTTP, которая обеспечивает шифрование данных для безопасной передачи информации между клиентом и сервером;
* HTML – это стандартизированный язык разметки документов для просмотра веб-страниц в браузере;
* SQL-запросы – это наборы команд для работы с реляционными (табличными) базами данных.
  1. Цели создания веб-приложения

Данное приложение создается для упрощения процесса обучения и преподавания путем автоматизации процесса проставления баллов и отслеживания статистики. Для достижения данной цели были выделены следующие задачи:

* Проектирование и развертывание базы данных;
* Разработка frontend части веб-приложения;
* Реализация бизнес-логики приложения на сервере;
* Реализация связи между клиентом и сервером с применением подхода REST API;
* Развертывание приложения на серверной части.
  1. Сфера применения

Приложения для учета успеваемости могут быть использованы в любом учебном заведении. Данные приложения имеют большое значение для многих средне-специальных и высших заведений, поскольку автоматизируют процесс выставления баллов, что ведет к уменьшению человеческого факта и временных затрат. Все это, в свою очередь, положительно сказывается на всем процессе обучения.

* 1. Требования к веб-приложению в целом

Приложение должно удовлетворять следующим требованиям:

* Обеспечить работу в Google Chrome, Firefox и Microsoft Edge;
* Предоставить возможность авторизации;
* Предоставить возможность студенту отслеживать свою успеваемость;
* Предоставить возможность преподавателю отслеживать успеваемость по своему предмету;
* Предоставить возможность администратору добавлять пользователей (как преподавателей, так и студентов), направления, дисциплины, баллы.
  1. Пользователи системы

В системе существуют такие группы пользователей как: преподаватель и студент.

Преподаватель – пользователь веб-сайта, ведущий некоторые дисциплины в учебном заведении. Для преподавателя должна быть реализована следующая функциональность:

* Авторизация в веб-приложении.
* Просмотр статистики студентов по своей дисциплине;
* Проставление баллов студентам по своей дисциплине.

Студент – пользователь, обучающийся в некотором учебном заведении. Для студента должна быть реализована следующая функциональность:

* Авторизация в веб-приложении;
* Просмотр своих баллов по всем предметам.

Администратор – пользователь, который имеет возможность вносить изменения в приложения и добавлять других пользователей. Для администратора должна быть реализована следующая функциональность:

* Возможность добавления новых пользователей;
* Возможность добавления новых направлений;
* Возможность добавления новых дисциплин;
* Возможность добавления новых групп студентов;
* Возможность редактирования баллов.

1. Анализ задачи
   1. Диаграмма прецедентов

Диаграмма прецедентов помогает описать, как пользователи будут взаимодействовать с системой.

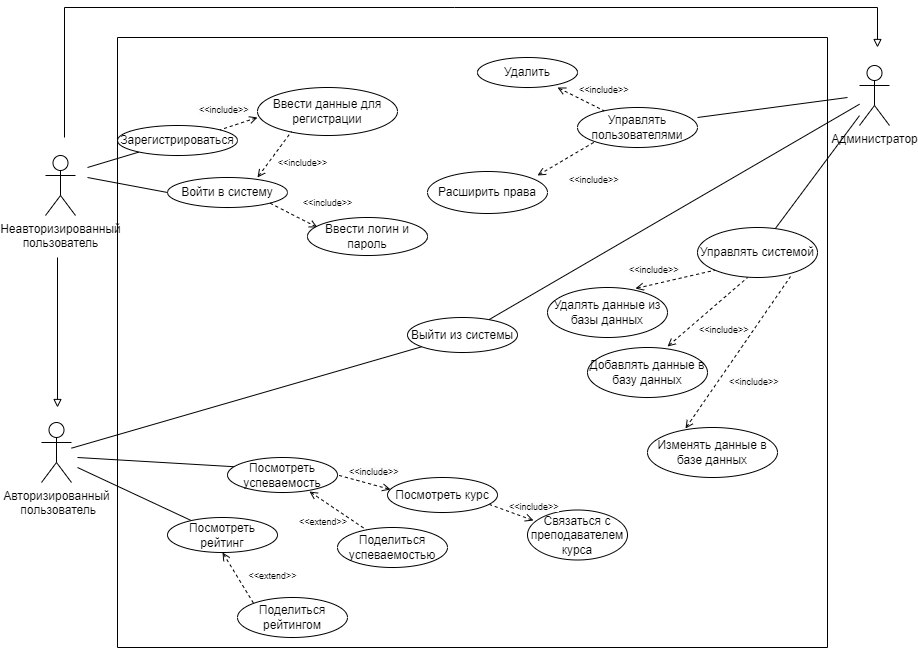


Рисунок 1 - Диаграмма прецедентов

* 1. Диаграмма последовательностей

Диаграмма последовательностей используется для уточнения диаграмм прецедентов, более детального описания логики сценариев использования.

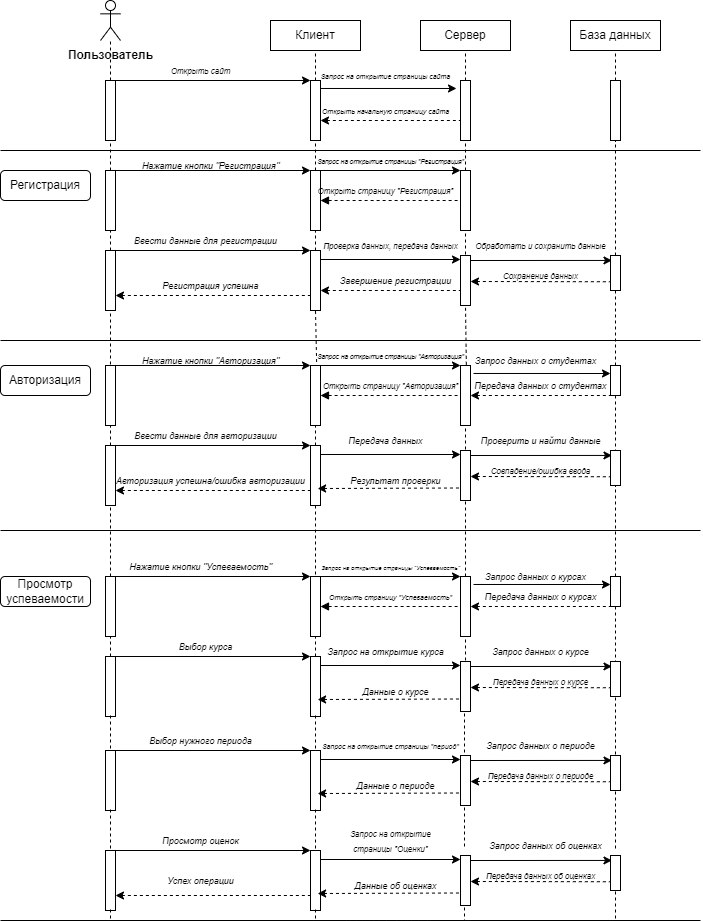


Рисунок 2 - Диаграмма последовательностей

* 1. Диаграмма состояний

Диаграмма состояний показывает все возможные изменения в состоянии конкретного объекта на протяжении всего его жизненного цикла.



Рисунок 3 - Диаграмма состояний

* 1. Диаграмма активностей

Диаграмма активностей отражает динамические аспекты поведения системы и наглядно показывает, как поток управления переходит от одной деятельности к другой.

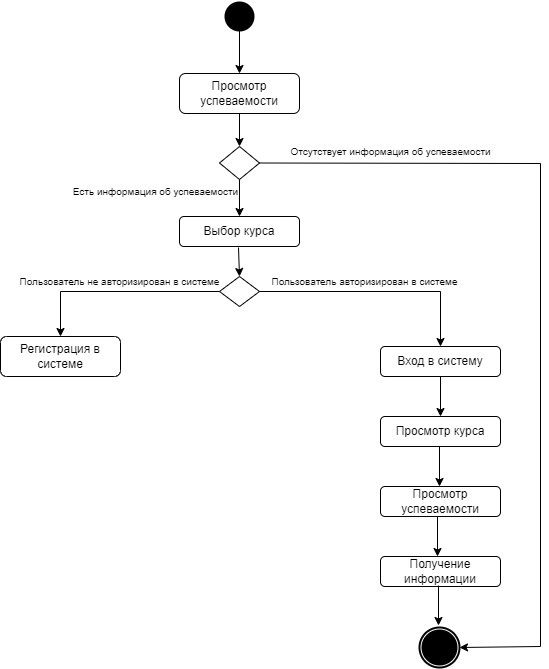


Рисунок 4 - Диаграмма активностей

1. Реализация
   1. Разработка frontend

При разработке клиентской части приложения использованы следующие технологии:

* JavaScript (JS) – язык программирования, который выполняется внутри браузера и позволяет внедрять в сайт различные функции на стороне клиента [1];
* React – библиотека с открытым исходным кодом, позволяющая создавать одностраничные приложения. Работает как с JS, так и с TS. Библиотека основана на компонентном подходе, что позволяет разбить интерфейс на небольшие, переиспользуемые компоненты, облегчая разработку и поддержку кода;

На рисунке 21 изображена структура веб-приложения на клиентской стороне.

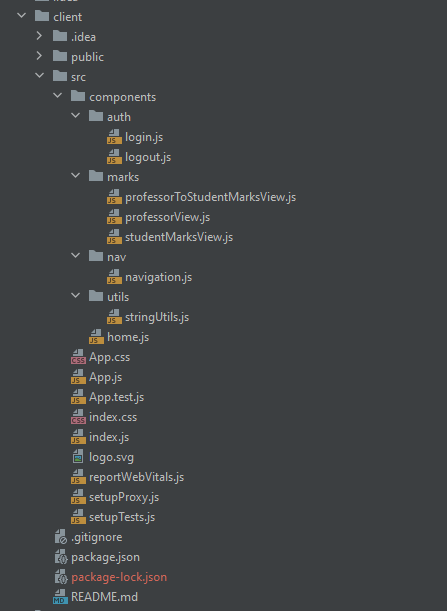


Рисунок 5 - Структура React-приложения

* 1. Разработка backend

При разработке серверной части приложения использованы следующие технологии:

* Основной язык – Python в связке с фреймворком Django. Django — бесплатный и свободный фреймворк для веб-приложений, написанный на Python. Приложение будет оперировать реляционной БД, в качестве СУБД будет использоваться СУБД с открытым исходным кодом PostgreSQL [3];
* В качестве инструмента развертки приложения будет использоваться Docker, который позволяет автоматизировать процесс развертывания и управления приложениями [4];
* Для документации разрабатываемого REST API будет использоваться Swagger, предоставляющий набор инструментов, который позволяет автоматически описывать API на основе его кода [5].

Структура приложения приведена на рисунке 6:

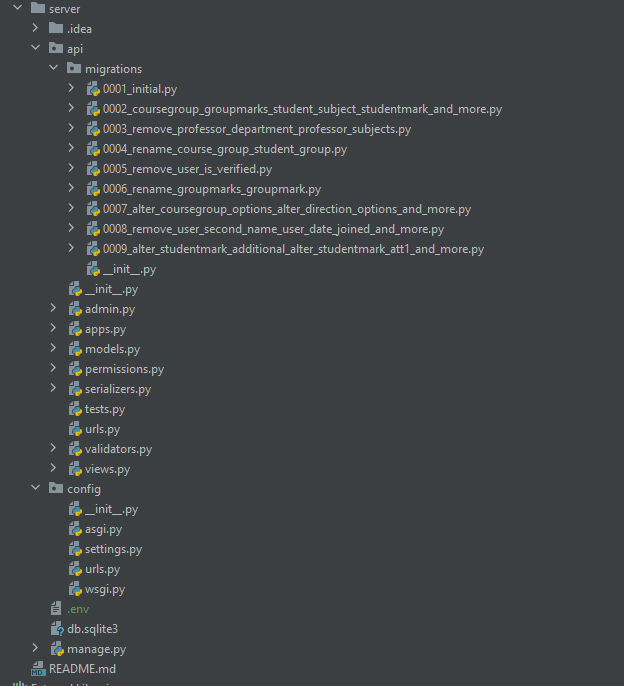


Рисунок 6 - Структура серверной части приложения

Сервер был развернут (загружен) на хостинге. База данных находится на сервере.

1. Навигация по приложению
   1. Экран авторизации

При первом запуске приложения пользователь попадает на экран авторизации, после чего может нажать на кнопку «Войти» и перейти на главный экран.

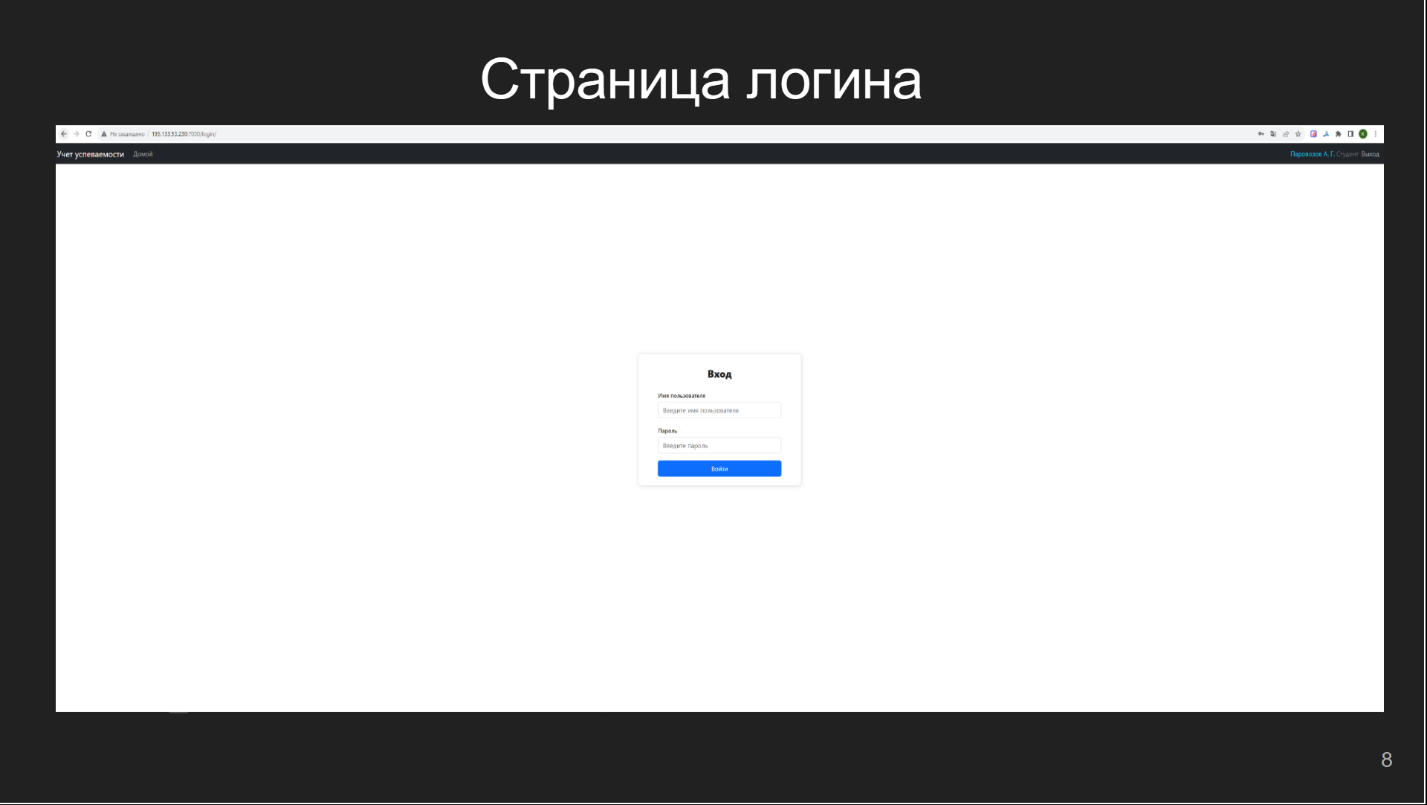


Рисунок 7 - Экран авторизации

После успешной авторизации пользователи-студенты попадают на экран успеваемости, а пользователи-преподаватели попадают на экран со своими дисциплинами.

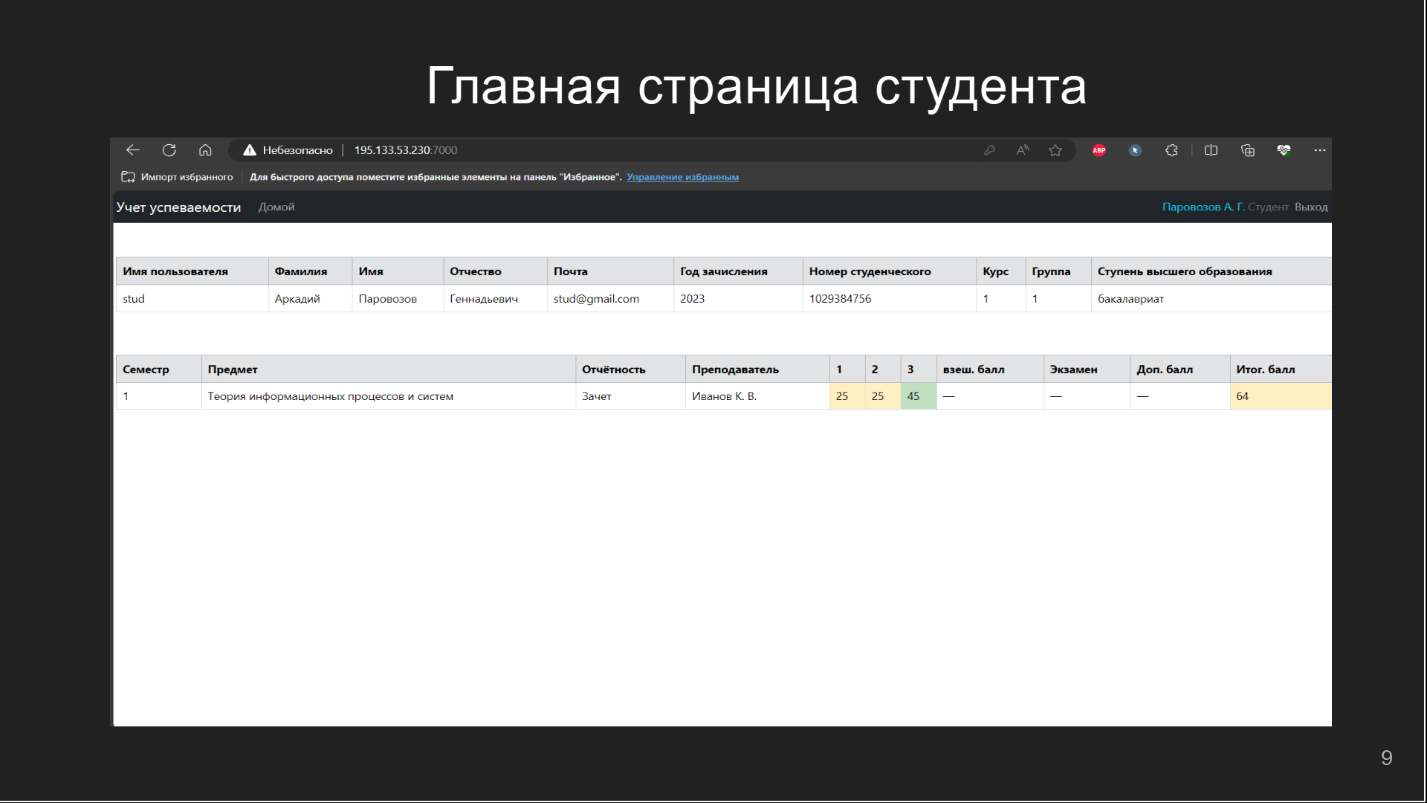


Рисунок 8 - Главный экран у пользователя-студента

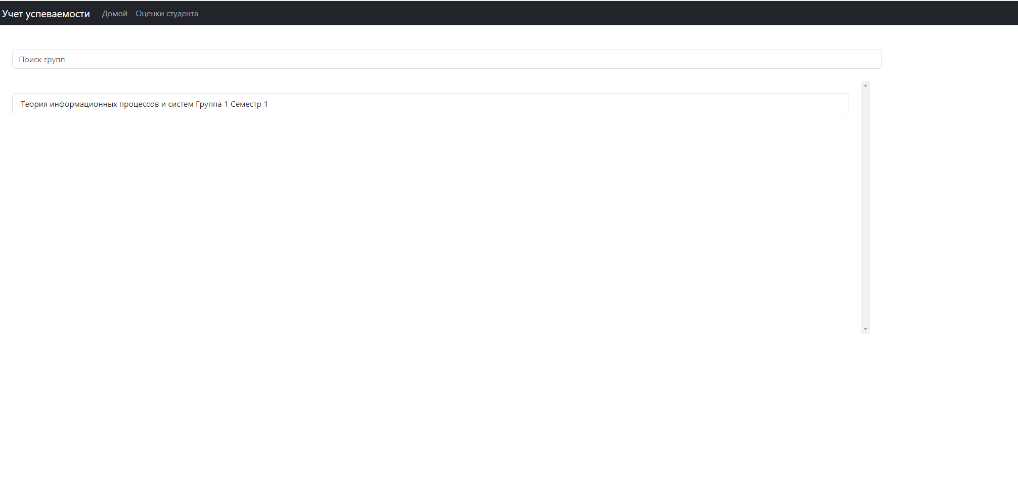


Рисунок 9 - Главный экран у пользователя-преподавателя

* 1. Экран администратора

После авторизации под логином администратора открывается панель администратора.

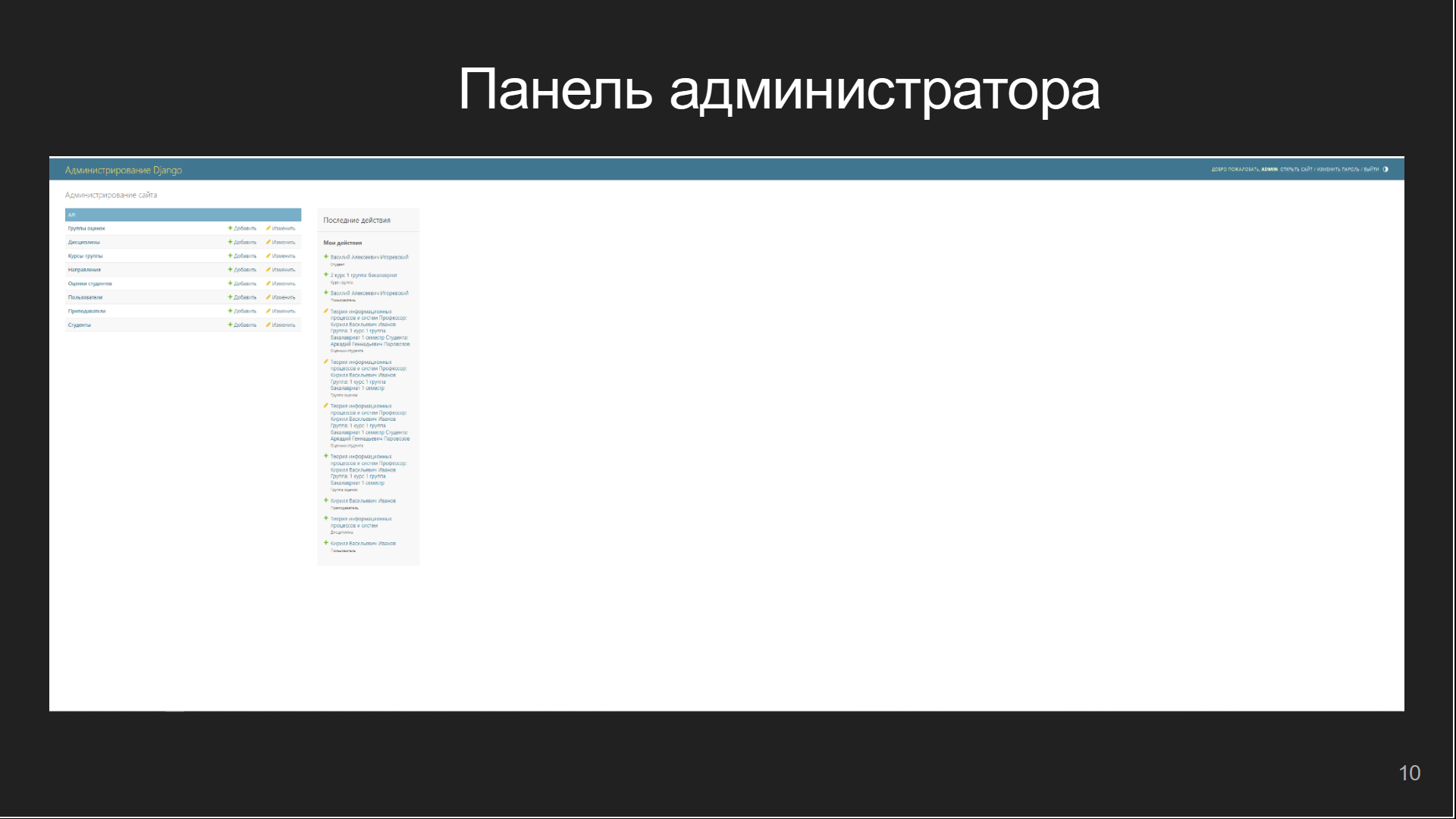


Рисунок 10 - Экран администратора

* 1. Экраны добавления данных

На экранах добавления данных администратор может добавлять, редактировать и удалять различные данные. В качестве примера приведен экран добавления пользователя.

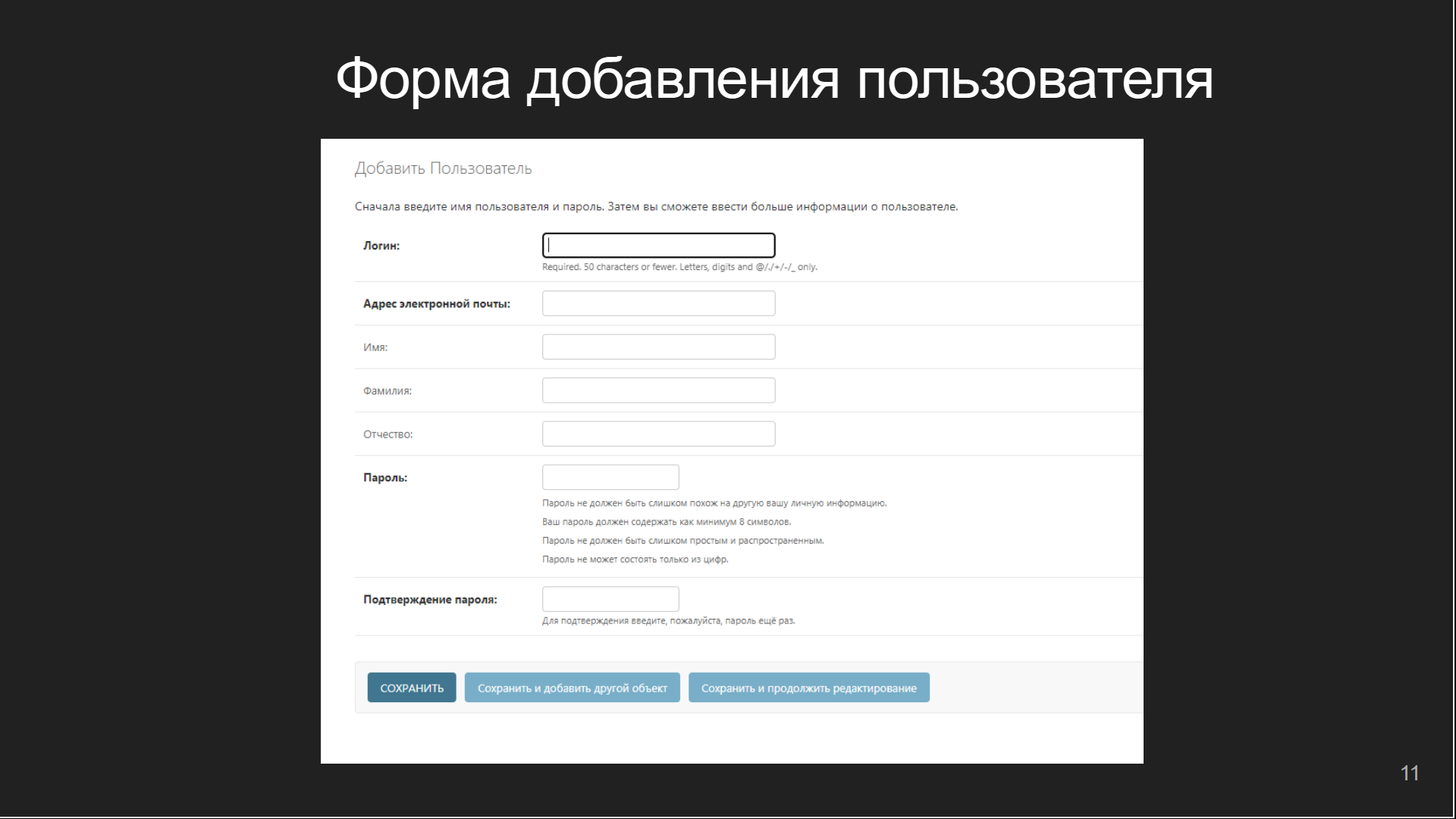


Рисунок 11 - Экран добавления пользователя

Заключение

В ходе данной курсовой работы был рассмотрен процесс разработки веб-приложения для учета успеваемости. Целью работы было создание приложения, которое будет полезно для широкого круга пользователей. В ходе выполнения проекта были выполнены следующие задачи:

* Проектирование и развертывание базы данных;
* Разработка frontend части веб-приложения;
* Реализация бизнес-логики на сервере;
* Реализация связи между клиентом и сервером с применением подхода REST API;
* Развертывание приложения на серверной части.

Веб-приложение предоставляет пользователю удобный способ учета успеваемости, а также возможность отслеживать статистику по каждой дисциплине. Основная функциональность приложения включает:

* Авторизация в веб-приложении.
* Просмотр статистики студентов по своей дисциплине;
* Проставление баллов студентам по своей дисциплине.
* Просмотр своих баллов по всем предметам.
* Возможность добавления новых пользователей;
* Возможность добавления новых направлений;
* Возможность добавления новых дисциплин;
* Возможность добавления новых групп студентов;
* Возможность редактирования баллов.

Для реализации данного приложения были использованы современные технологии веб-разработки, включая React, Python и PostgreSQL. Благодаря этим технологиям удалось создать функциональный интерфейс и обеспечить работу веб-приложения.

В результате данной работы было создано веб-приложение, которое соответствует поставленным целям и задачам. Оно предоставляет пользователям возможность отслеживать свою успеваемость. Приложение может быть полезным инструментом для многих пользователей и учебных заведений, помогая им уменьшить временные потери и достичь выполнения плана обучения.

Список использованных источников

1. MDN Web Docs — JavaScript [Электронный ресурс]. — Режим доступа: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide. — Заглавие с экрана. — (Дата обращения: 20.04.2023).
2. Что такое React и как его освоить? [Электронный ресурс]. — Режим доступа: https://academy.yandex.ru/journal/chto-takoe-react-i-kak-ego-osvoit. — Заглавие с экрана. — (Дата обращения: 29.04.2023).
3. Система управления объектно-реляционными базами данных PostgreSQL [Электронный ресурс]. — Режим доступа: https://web-creator.ru/technologies/webdev/postgresql. — Заглавие с экрана. — (Дата обращения: 2.05.2023).
4. Моуэт, Э. Использование Docker: Разработка и внедрение программного обеспечения при помощи технологии контейнеров. — Перевод с английского. — Москва: Издательство: «ДМК Пресс», 2019. - 156 с.
5. Тестирование API с помощью Swagger: особенности и преимущества [Электронный ресурс]. — Режим доступа: https://blog.ithillel.ua/ru/articles/api-testing-with-swagger. — Заглавие с экрана. — (Дата обращения: 14.05.2023).