

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук  
Кафедра информационных систем

Система учёта успеваемости и посещаемости  
студентов факультета «Зачётка.ру»

Курсовая работа

*09.03.02 Информационные системы и технологии*  
*Информационные системы и сетевые технологии*  
*6 семестр 2022/2023 учебного года*

Зав. кафедрой	_____	к. т. н., доцент Д.Н. Борисов
Обучающийся	_____	ст. 3 курса оч. отд. А.О. Левин
Обучающийся	_____	ст. 3 курса оч. отд. Ф.Г. Парамонов
Обучающийся	_____	ст. 3 курса оч. отд. Д.Д. Боброва
Руководитель	_____	ст. преп. В.С. Тарасов

Воронеж 2023

## Содержание

Введение .....	4
1 Постановка задачи.....	5
2 Цель создания программы .....	5
3 Анализ предметной области .....	6
3.1 Сфера применения.....	6
3.2 Технический обзор .....	7
3.3 Анализ аналогов .....	9
3.4 Требования к функциональности .....	12
3.4.1 Неавторизованный пользователь.....	12
3.4.2 Авторизованный пользователь .....	12
3.4.3 Администратор .....	13
3.4.4 Преподаватель .....	13
3.4.5 Студент .....	15
3.5 Требования, не касающиеся функциональной части .....	16
3.6 Оценка актуальности создания веб-приложения.....	16
4 Реализация .....	19
4.1 Средства реализации .....	19
4.2 Архитектура приложения .....	22
4.2.1 База данных.....	22
4.2.2 Схема базы данных.....	23
4.2.3 Диаграмма классов .....	24
4.2.4 Диаграмма прецедентов .....	24
4.2.5 IDEF0 диаграмма .....	25
4.2.6 Диаграмма последовательности .....	26
4.2.7 Диаграмма развёртывания .....	26
4.2.8 Диаграмма потока данных .....	27
4.2.9 Диаграмма активностей .....	28
4.3 Серверная часть.....	28
4.3.1 Основные положения .....	28

4.3.2 Развёртывание .....	29
4.4 Клиентская часть .....	29
4.4.1 Графический пользовательский интерфейс .....	30
4.5 Тестирование .....	36
Заключение .....	41
Список использованных источников .....	42

## **Введение**

В настоящее время использование информационных технологий в образовательном процессе является неотъемлемой частью современного образования. Технологии помогают не только автоматизировать рутинные задачи, но и упростить и улучшить процессы обучения и оценки результатов учебной деятельности.

Одним из важных аспектов образовательного процесса является учет успеваемости студентов, который позволяет оценить их академические результаты и прогресс в учебе. В традиционных методах учета успеваемости используется ручное ведение журналов, что может быть неэффективным и неудобным для использования. Кроме того, ручное ведение журналов не позволяет проводить детальный анализ результатов учебной деятельности студентов, что может затруднять принятие обоснованных решений.

В связи с этим возникает необходимость в использовании специализированных программных продуктов, которые позволяют автоматизировать процесс учета успеваемости студентов. В настоящее время существует множество систем учета успеваемости студентов, в том числе веб-приложений, которые позволяют не только автоматизировать процесс учета, но и проводить анализ результатов учебной деятельности.

## **1 Постановка задачи**

В ходе данной работы нужно разработать веб-приложение для учёта успеваемости и посещаемости студентов. Для этого необходимо:

- разработать функциональные требования к веб-приложению, учитывающие потребности пользователей и особенности учета успеваемости и посещаемости студентов;
- спроектировать базу данных для хранения информации об успеваемости и посещаемости студентов;
- реализовать веб-приложение в соответствии с функциональными требованиями;
- разработать дизайн веб-приложения и сверстать его;
- протестировать веб-приложение на соответствие требованиям, выявить и устранить возможные ошибки.

Выполнение этих задач позволит создать веб-приложение, которое будет эффективно выполнять свои функции, обеспечивать удобство использования и защиту данных.

## **2 Цель создания программы**

Цель данной курсовой работы – создание веб-приложения для последующей интеграции его в учебные заведения. Веб-приложение должно обеспечить возможность учета преподавателями и просмотра студентами успеваемости и посещаемости занятий на факультете учебного заведения. Оно позволит автоматизировать процесс ведения журналов успеваемости и посещаемости, а также анализа результатов учебной деятельности студентов.

Данным приложением будет удобно пользоваться как студентам, так и преподавателям: для студентов вся необходимая информация собрана в одном месте, а для преподавателей предусмотрена функция автоматической отметки посещаемости студентов на занятии.

В рамках работы будет рассмотрено проектирование и реализация функциональных возможностей веб-приложения, в том числе разработка базы данных для хранения информации об успеваемости студентов и разработка интерфейса пользователя.

### **3 Анализ предметной области**

#### **3.1 Сфера применения**

Такие приложения могут быть использованы в различных учебных заведениях – от школ до вузов. Они предназначены для автоматизации учета успеваемости студентов и облегчения работы преподавателей и администрации образовательных учреждений.

Программное обеспечение для учета успеваемости может быть использовано для следующих задач:

- учет посещаемости студентов: с помощью приложения преподаватели могут отмечать, кто был на занятии, а кто - нет. Также это помогает администрации учебного заведения контролировать посещаемость студентов и отслеживать их активность;
- учет оценок: приложение позволяет преподавателям вводить оценки и оценивать работы студентов. Оценки могут быть записаны в журнал, а также использоваться для создания отчетов и анализа успеваемости;
- учёт посещаемости: приложение позволяет преподавателям отмечать, посещал ли студент занятия. Это поможет преподавателю принимать решения по поводу оценки работы студента в течение семестра при возникновении спорных ситуаций;
- анализ успеваемости студентов: приложение может собирать данные о успеваемости студентов и анализировать их. Это

позволяет выявлять проблемные моменты и проводить анализ эффективности обучения.

Таким образом, веб-приложения для учета успеваемости студентов могут быть использованы в различных образовательных учреждениях для автоматизации процессов учета и анализа успеваемости студентов, что в свою очередь способствует повышению качества обучения и улучшению работы преподавателей.

### **3.2 Технический обзор**

Рассмотрим технические аспекты, которые следует учитывать при создании веб-приложения для учета успеваемости студентов.

Язык программирования – необходимый инструмент для создания любого веб-приложения. При создании приложения для учета успеваемости студентов можно использовать различные языки программирования. Так, PHP является наиболее популярным языком программирования для создания веб-приложений. Он широко используется для разработки сайтов и веб-приложений, основанных на системе управления контентом (Content management system, CMS). Python также популярен среди разработчиков веб-приложений и обладает мощными инструментами для создания сложных приложений. Ruby, Java и другие языки программирования также могут использоваться для создания веб-приложений.

Также фреймворки являются важными инструментами для создания веб-приложений, поскольку они позволяют разработчикам быстро и эффективно создавать приложения, используя готовые компоненты. Существует множество фреймворков для создания веб-приложений, включая Laravel, Django, Ruby on Rails, Spring и другие. Laravel и Django являются популярными фреймворками, которые обладают мощными инструментами для создания веб-приложений и позволяют разработчикам быстро и эффективно создавать приложения.

Базы данных необходимы для хранения данных, связанных с учетом успеваемости студентов. Для создания веб-приложения можно использовать такие базы данных, как MySQL, PostgreSQL, MongoDB или другие. MySQL является наиболее распространенной базой данных, используемой в веб-разработке, и обладает хорошей производительностью и широким набором инструментов для управления данными. PostgreSQL также популярен среди разработчиков веб-приложений и обладает мощными функциями для управления данными. MongoDB является NoSQL базой данных, которая позволяет хранить данные в формате JSON.

Интерфейс – важная часть любого веб-приложения. Для создания интерфейса пользователя можно использовать различные технологии, включая HTML, CSS и JavaScript. HTML является языком разметки, который определяет структуру и содержимое веб-страницы. CSS позволяет определять внешний вид и стиль веб-страницы, включая цвета, шрифты и расположение элементов. JavaScript позволяет добавлять динамическое поведение к веб-странице, например, валидацию форм, анимацию и интерактивность.

Системы контроля версий позволяют разработчикам работать совместно над кодом, отслеживать изменения и вносить правки безопасно. Для управления кодом при создании веб-приложения для учета успеваемости студентов можно использовать системы контроля версий, такие как Git.

Для размещения веб-приложения можно использовать различные хостинг-провайдеры, включая Timeweb, Spaceweb, Reg.ru и другие. Эти провайдеры предоставляют облачные ресурсы для размещения и масштабирования веб-приложений.

Безопасность является важным аспектом при создании веб-приложений для учета успеваемости студентов. Веб-приложения могут быть подвержены атакам, таким как SQL-инъекции, кросс-сайтовые скрипты и другие. Для обеспечения безопасности приложения можно использовать различные методы, включая хэширование паролей, проверку ввода данных и использование SSL-сертификатов.



Таким образом, создание веб-приложения для учета успеваемости студентов подразумевает под собой использование различных технологий, включая языки программирования, фреймворки, базы данных, интерфейс пользователя, системы контроля версий, хостинг сервисы.

### **3.3 Анализ аналогов**

На рынке существует множество различных веб-приложений для учета успеваемости студентов. Рассмотрим некоторые из наиболее популярных аналогов, а также их преимущества и недостатки:

Moodle — это популярная система управления обучением с открытым исходным кодом. Она предоставляет функциональность для учета посещаемости, выставления оценок и анализа успеваемости. Moodle также имеет множество дополнительных модулей, которые расширяют ее функциональность.

Преимущества:

- бесплатный и имеет открытый исходный код;
- множество дополнительных модулей и плагинов для расширения функциональности;
- поддержка многих языков.

Недостатки:

- требуется некоторое время для изучения и освоения системы;
- управление и настройка может потребовать некоторых технических знаний;
- некоторые пользователи могут считать интерфейс системы неудобным или устаревшим.

Blackboard — это еще одна популярная система управления обучением, которая используется во многих учебных заведениях по всему миру. Она предоставляет множество функций, включая учет посещаемости, выставление оценок, обмен сообщениями и т.д.

Преимущества:

- удобный интерфейс и интуитивно понятный дизайн;
- множество интегрированных инструментов для управления обучением;
- поддержка многих языков.

Недостатки:

- высокая стоимость, особенно для небольших учебных заведений;
- некоторые функции могут быть ограничены или требовать дополнительной оплаты;
- некоторые пользователи могут считать систему слишком сложной для использования.

Gradescope — это веб-приложение, которое специализируется на проверке заданий и выставлении оценок. Оно позволяет преподавателям быстро проверять большие объемы работ и выставять оценки.

Преимущества:

- быстрое выставление оценок и проверка больших объемов работ;
- возможность загрузки и проверки различных форматов файлов;
- интеграция с другими системами управления обучением.

Недостатки:

- основная функциональность системы ограничена проверкой заданий и выставлением оценок;
- нет возможности учета посещаемости или анализа успеваемости студентов;
- стоимость использования может быть высокой для небольших учебных заведений.

Easyclass — это веб-приложение, которое предоставляет функциональность для учета посещаемости, выставления оценок и обмена сообщениями. Easyclass также позволяет преподавателям создавать онлайн-уроки и делиться материалами с учениками.

Преимущества:

- удобный и простой в использовании интерфейс;

- множество функций для учета посещаемости, выставления оценок и обмена сообщениями;
- поддержка многих языков.

Недостатки:

- некоторые функции могут быть ограничены или требовать дополнительной оплаты;
- не всегда достаточно гибок для учета специфических требований учебных заведений;
- сервис не ориентирован на высшие учебные заведения;
- могут возникать проблемы с безопасностью данных.

Dnevnik.ru — это популярный сервис управления образовательным процессом, предоставляющий широкий функционал для учета посещаемости, успеваемости и взаимодействия между участниками образовательного процесса.

Преимущества:

- широкий функционал: Dnevnik.ru предлагает разнообразные возможности для учета посещаемости, выставления оценок, анализа успеваемости и взаимодействия между учителями, студентами и родителями;
- интеграция с другими инструментами: сервис позволяет интегрироваться с различными системами и приложениями, что обеспечивает более полное управление образовательным процессом;
- удобный интерфейс: Dnevnik.ru предоставляет простой и понятный интерфейс, что упрощает использование сервиса для всех участников образовательного процесса.

Недостатки:

- некоторые функции могут быть ограничены или требовать дополнительной оплаты;
- сервис не ориентирован на высшие учебные заведения;

— могут возникать проблемы с безопасностью данных.

Это лишь некоторые из популярных аналогов веб-приложений для учета успеваемости студентов. Несмотря на то, что каждый из них имеет свои преимущества и недостатки, веб-приложение для учета успеваемости студентов, созданное специально для конкретной учебной организации, может быть более гибким и удобным для использования, так как учитывает специфические потребности этой организации и не имеет ограничений, связанных с использованием сторонних систем.

### **3.4 Требования к функциональности**

Для реализации требуемых функций в веб-приложении предусмотрены три типа авторизованных пользователей: администратор, преподаватель, студент. Каждый тип будет иметь свой набор функциональных возможностей, а неавторизованные пользователи не смогут получить доступ ко всем функциональным возможностям веб-приложения. Часть возможностей, которые присутствует у всех пользователей, для удобства расписана отдельно.

#### **3.4.1 Неавторизованный пользователь**

Неавторизованный пользователь имеет доступ только к странице авторизации. Она же «вход в систему».

На странице «вход в систему» представлены поля для ввода логина, пароля и кнопка «вход».

#### **3.4.2 Авторизованный пользователь**

У любого авторизованного пользователя вверху каждой страницы представлена шапка, на которой есть кнопки «сменить пароль» и «выход».

При нажатии на кнопку «сменить пароль» пользователь попадает на страницу «смена пароля», на которой представлены три поля для заполнения:

- текущий пароль;
- новый пароль;

— повторный ввод нового пароля.

После заполнения всех полей пользователь может нажать кнопку «сменить пароль» для смены пароля.

При нажатии на кнопку «выход», пользователь может выйти из своей учётной записи, после чего он попадёт на страницу авторизации.

### **3.4.3 Администратор**

После авторизации в системе администратор попадает на страницу «панель администратора», содержащую перечень доступных ему функций. На данной странице расположены кнопки «добавить администратора», «добавить студента», «добавить преподавателя», «добавить группу», «добавить пару», «добавить дисциплину», «удалить пару», «удалить дисциплину», «изменить дисциплину». Также на данной странице представлены кнопки «перевести на следующий семестр», «перевести на предыдущий семестр», позволяющие переключить текущий семестр для всех студентов.

### **3.4.4 Преподаватель**

После авторизации в системе преподаватель попадает на страницу «выбор дисциплины», отображающую все преподаваемые им дисциплины. По нажатию на наименование дисциплины происходит переход на страницу «выбор группы», содержащую список групп студентов. Также на этой странице есть кнопка «проверка посещаемости» при нажатии на которую откроется всплывающее окно, в котором преподаватель выбирает дату и время занятия, а также длительность процесса проверки посещаемости. Возможная длительность процесса – от 10 секунд до 15 минут. После выбора параметров, преподаватель нажимает на кнопку «запустить проверку посещаемости», тем самым начиная процесс проверки посещаемости. После начала процесса, преподавателю необходимо сообщить находящимся на занятии студентам выданный ему системой код посещаемости – сгенерированную последовательность длиной в 7 символов, состоящую из цифр и букв

английского алфавита. Также преподаватель может предоставить студентам для сканирования специальный QR-код, содержащий в себе ссылку, перейдя по которой авторизованный в системе студент, отмечает своё присутствие на занятии. После завершения процесса проверки посещаемости, будет отображен список групп студентов, которые отметились на занятии с указанием для каждой группы числа студентов, отметивших своё присутствие. Также необходимо отобразить общее число присутствующих на занятии студентов. Сразу после проведения проверки посещаемости, если преподаватель сомневается в том, что студенты отметились добросовестно, он может нажать на кнопку «сбросить учтённую сейчас посещаемость», тем самым аннулировав отмеченную студентами посещаемость по этому занятию.

Со страницы «выбор группы» по нажатию на номер группы можно перейти на страницу «учёт посещаемости по дисциплине», содержащую таблицу посещаемости студентов этой группы. В эту таблицу преподавателем вносятся данные о посещаемости студентов. Данными о посещаемости является информация о пропуске студентом конкретного занятия, общее количество пропусков, а также количество пропущенных студентом занятий, разделённое на следующие категории: «Н» - пропуск без уважительной причины, «П» - пропуск по уважительной причине, «Б» - пропуск по болезни. Выше таблицы есть пункт, отображающий общее количество занятий.

Со страницы «учёт посещаемости по дисциплине» возможен переход на страницу «учёт успеваемости по дисциплине», содержащую таблицу оценок студентов этой группы. В эту таблицу преподавателем вносятся данные об успеваемости студентов. Данными об успеваемости являются оценки за три контрольные точки (аттестации), средний балл за три контрольные точки (аттестации), оценка за экзамен, итоговая оценка. Оценка за каждую из аттестаций выставляется в виде балла от 0 до 50. Средний балл за аттестации рассчитывается системой автоматически в виде балла от 0 до 50. Оценка за экзамен является баллом от 0 до 100. Для дисциплин с типами отчётности «зачёт» и «зачёт с оценкой», оценка за экзамен не выставляется. Градация

итоговых оценок для дисциплин с типами отчётности «экзамен» и «зачёт с оценкой» имеет следующий вид: 0–49 – «неудовлетворительно», 50–69 – «удовлетворительно», 70–89 – «хорошо», 90–100 – «отлично». Градация оценок для зачёта имеет следующий вид: 0–49 – «не зачтено», 50–100 – «зачтено». Все заносимые в систему баллы являются целыми, неотрицательными числами.

После перехода на страницу дисциплины на всех последующих страницах отображается навигационная цепочка, содержащая название выбранного ранее предмета, группы и страницы с её посещаемостью или оценками.

### **3.4.5 Студент**

После авторизации в системе студент попадает на страницу «еженедельное расписание». На этой странице отображается расписание студента на текущую неделю. Оно представлено в формате «разворота дневника»: слева – понедельник, вторник, среда; справа – четверг, пятница, суббота. Если размер экрана не позволяет отображать «разворот дневника», то дни недели располагаются друг под другом. В расписании указаны дата и время занятий, название дисциплины, номер аудитории (или «ДО», если занятие проходит в дистанционной форме), ФИО преподавателя, пункт для отображения посещаемости. При нажатии на ФИО преподавателя, открывается всплывающее окно с полным ФИО и контактными данными этого преподавателя. В пункте для отображения посещаемости указан статус посещения студентом этого занятия. Эти данные вносятся преподавателем на странице «учёт посещаемости по дисциплине». Студент также может переключать недели назад и вперёд. Также на этой странице есть кнопка «код посещаемости», при нажатии которой открывается всплывающее окно, содержащее в себе поле для ввода кода и кнопка «отметить».

С этой страницы возможен переход на страницу «моя успеваемость», на которой отображена таблица с текущей успеваемостью студента. В таблице

есть название дисциплины, ФИО преподавателя, оценки за три контрольные точки (аттестации), средний балл за три контрольные точки (аттестации), оценка за экзамен, итоговая оценка. Аналогично предыдущей странице, на этой также находится кнопка «код посещаемости».

### **3.5 Требования, не касающиеся функциональной части**

Разрабатываемое веб-приложение должно иметь архитектуру «клиент–сервер». Также необходимо реализовать взаимодействие front-end и back-end частей веб-приложения согласно архитектурному стилю REST.

Интерфейс веб-приложения необходимо оформить на русском языке. Иные языковые версии, помимо русскоязычной, не предусмотрены.

Все страницы веб-приложения должны быть выполнены в едином стиле. Вёрстка должна корректно отображаться в следующих браузерах:

- Google Chrome (версия desktop 95 или новее);
- Microsoft Edge (версия desktop 95 или новее);
- Opera (версия desktop 90 или новее).

Также корректное отображение веб-приложения на мобильных устройствах гарантируется в браузере Google Chrome (версия mobile 95 или новее).

### **3.6 Оценка актуальности создания веб-приложения**

С целью выяснения актуальности создания веб-приложения для учёта посещаемости и успеваемости, был проведён социологический опрос среди студентов. В опросе приняли участие 42 студента, которым были заданы вопросы об удобстве просмотра расписания, успеваемости и посещаемости, а также о предпочтении объединения всех этих функций в одно веб-приложение. Студенты оценивали функции по шкале от 1 до 5, где 1 - "Совсем неудобно", 5 - "Очень удобно".



Большинство опрошенных (31%) оценили удобство просмотра своего расписания как "3", в то время как 28.6% выбрали вариант ответа "2", а 16.7% выбрали "4". Однако только 9.5% студентов полностью довольны удобством просмотра расписания, и 14.3% напротив считают его совсем неудобным. Эту статистику можно увидеть на рисунке 1.

Насколько Вам удобно просматривать своё расписание?

42 responses

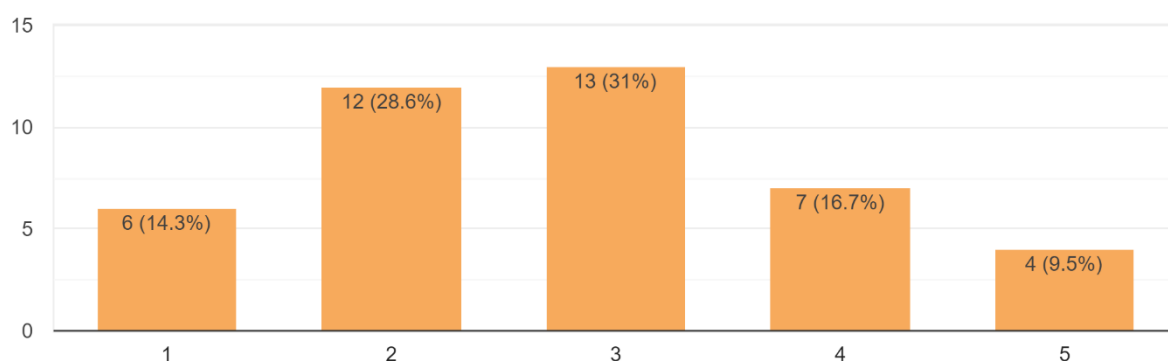


Рисунок 1 – Оценка удобства просмотра расписания

Удобством просмотра своей успеваемости полностью довольны 42.9% опрошенных студентов, тогда как 35.7% выбрали вариант ответа "4", 14.3% - "3", 4.8% - "2" и 2.4% - "1". Это означает, что большинство студентов считают удобным просмотр своей успеваемости. Это отображается на рисунке 2.

Насколько Вам удобно просматривать свою успеваемость?

42 responses

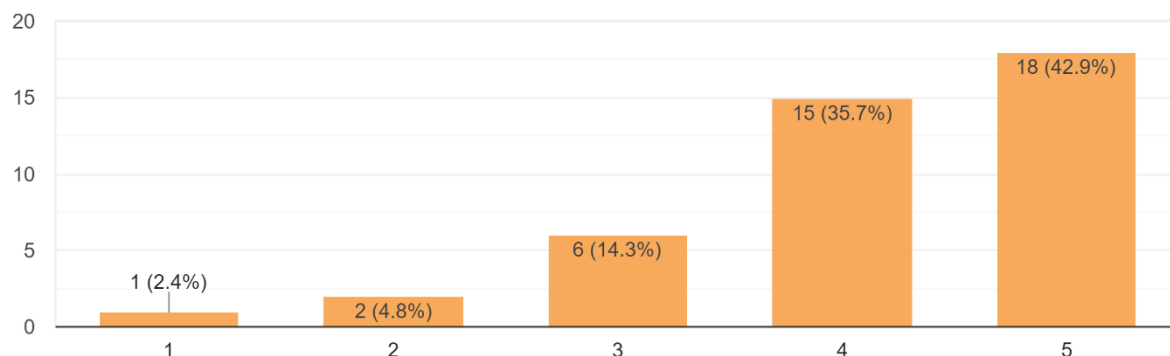


Рисунок 2 – Оценка удобства просмотра успеваемости

Касательно удобства просмотра своей посещаемости, только 4.8% опрошенных полностью довольны этим аспектом. 9.5% выбрали вариант ответа "3", в то время как 33% выбрали "2" и 47.6% - "1". Таким образом, большинство студентов считают просмотр своей посещаемости неудобным. На рисунке 3 это изображено визуально.

Насколько удобно Вам проверять свою посещаемость, чтобы узнать, какие конкретно занятия Вы пропустили?

42 responses

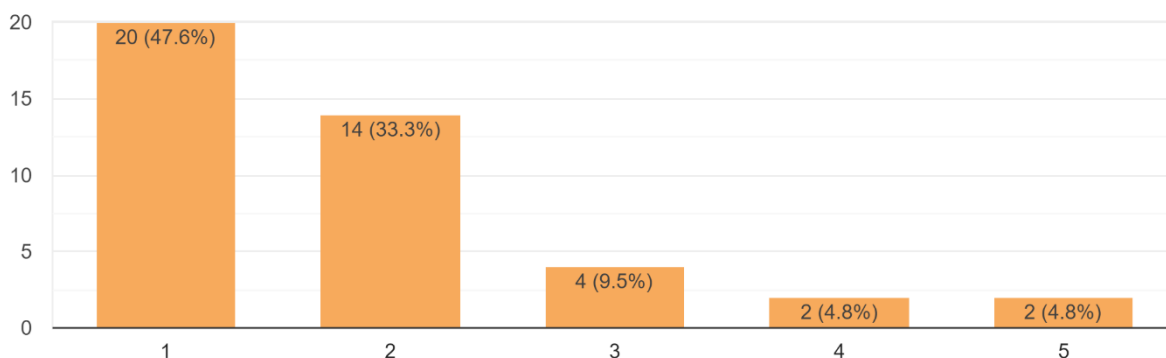


Рисунок 3 – Оценка удобства просмотра посещаемости

В ответ на вопрос о предпочтении объединения всех этих систем в одно веб-приложение, 90,5% опрошенных студентов ответили, что это было бы удобно, а 9,5% считают, что такое объединение не нужно, так как уже есть отдельные сервисы. Это продемонстрировано на рисунке 4.

Как Вы считаете, было бы удобным для Вас объединение всех этих систем в одно веб-приложение, в котором можно было бы пр...сещаемость и расписание на каждую неделю?  
42 responses

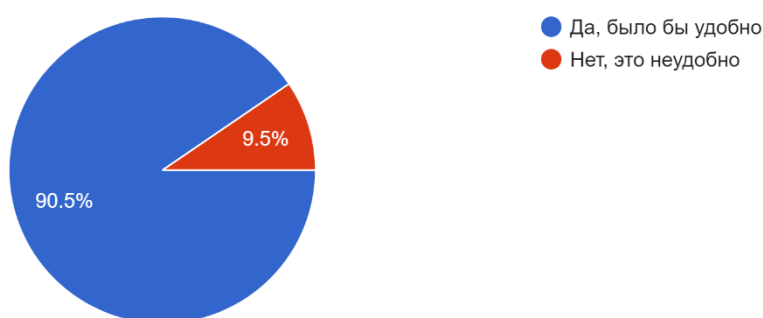


Рисунок 4 – Предпочтение объединения систем

Из результатов данного социологического опроса следует, что большинство студентов считают хорошей идеей создание веб-приложения, которое объединит функции учета посещаемости и успеваемости, а также предоставит возможность просмотра своего расписания, что позволит им удобно отслеживать свой прогресс в учебе.

## 4 Реализация

### 4.1 Средства реализации

В качестве средств реализации для разработки веб-приложения были выбраны:

- TypeScript (сервер и клиент);
- NestJS framework (сервер);
- Next.js framework (клиент);
- React;

Решение о выборе TypeScript [1] в качестве основного языка программирования для разработки веб-приложения обусловлено его строгой типизацией, которая является одним из главных преимуществ по сравнению с JavaScript. Строгая типизация в TypeScript позволяет выявлять и предотвращать ошибки на ранних стадиях разработки, повышая надежность и стабильность приложения.

Кроме того, TypeScript предоставляет разработчику широкий набор инструментов и возможностей для разработки. Он поддерживает объектно-ориентированный подход, модульность и разделение кода на множество файлов, что упрощает поддержку и расширение проекта.

Также стоит отметить, что TypeScript имеет преимущества по сравнению с другими языками программирования, такими как Java и Python. В отличие от Java, TypeScript обладает более лаконичным и выразительным синтаксисом, что способствует повышению производительности разработчика. По сравнению с Python, TypeScript обеспечивает компиляцию в нативный JavaScript, что позволяет эффективно выполнять код на стороне клиента и обеспечивает высокую скорость работы приложения.

Использование TypeScript в разработке веб-приложения является стратегическим выбором, который позволяет повысить надежность, масштабируемость и производительность приложения, обеспечивая комфортную и эффективную разработку.

При выборе фреймворка для разработки веб-приложения, для серверной части было принято решение использовать NestJS [2]. Этот выбор обоснован несколькими факторами.

Во-первых, NestJS обладает модульностью, что позволяет разделить приложение на независимые части и упростить его разработку и поддержку. Во-вторых, NestJS полностью поддерживает TypeScript, что обеспечивает строгую типизацию и улучшает читаемость и понятность кода. Кроме того, фреймворк основан на принципах SOLID, что способствует созданию гибкого

и легко поддерживаемого кода. NestJS также предлагает расширяемость и поддержку, благодаря активному сообществу разработчиков и обширной документации. И наконец, фреймворк хорошо интегрируется с другими инструментами, что обеспечивает гибкость и возможности расширения приложения.

Все эти факторы делают NestJS предпочтительным выбором для разработки веб-приложений, обеспечивая гибкость, надежность и эффективность в процессе разработки.

Для клиентской части был выбран фреймворк Next.js [3]. Такой выбор также обусловлен несколькими факторами.

Next.js предоставляет возможность создания мощных и масштабируемых веб-приложений с использованием TypeScript. Он обеспечивает быструю загрузку страниц, отличную оптимизацию для SEO, а также удобную разработку благодаря функциям автоматической предварительной загрузки и отображения на стороне сервера.

Выбор Next.js и NestJS обусловлен их способностью работать вместе и обеспечивать полноценную разработку приложения как на фронтенде, так и на бэкенде. Использование одного языка программирования (TypeScript) для обеих частей приложения упрощает командную работу, обеспечивает типизацию и улучшает общую надежность проекта. Кроме того, экосистема и сообщество Next.js и NestJS предоставляют обширную документацию, готовые решения и поддержку, что облегчает разработку и поддержку приложения на протяжении всего жизненного цикла проекта.

Выбор React [4] и Material-UI [5] был обоснован рядом факторов, которые делают эти технологии предпочтительными для разработки веб-приложения.

React - одна из самых популярных библиотек JavaScript для создания пользовательских интерфейсов. Она обладает компонентной архитектурой, что позволяет разбивать интерфейс на небольшие и независимые компоненты.

Строгая типизация TypeScript дополняет React, обеспечивая большую надежность и удобство в разработке.

Для удобной стилизации и создания современного интерфейса была выбрана Material-UI - библиотека компонентов пользовательского интерфейса, основанная на принципах Material Design. Она предоставляет готовые стилизованные компоненты, а также возможность настраивать их в соответствии с нашими потребностями.

Основываясь на преимуществах React, TypeScript и Material-UI, создаётся мощное и гибкое веб-приложение, которое обеспечивает высокую производительность, удобство разработки и привлекательный пользовательский интерфейс. Комбинация этих технологий с использованием NestJS в качестве фреймворка обеспечивает надежность и эффективность работы приложения.

## **4.2 Архитектура приложения**

### **4.2.1 База данных**

В качестве базы данных используется объектно-реляционная СУБД PostgreSQL. Такой выбор был сделан по нескольким причинам.

Эта система управления базами данных известна своей надежностью, стабильностью и широкими функциональными возможностями.

PostgreSQL обладает мощными инструментами, позволяющими эффективно работать с большими объемами данных и достигать высокой производительности. Он также предлагает широкий выбор расширений и возможностей настройки, позволяющих адаптировать базу данных под конкретные требования приложения.

Совместимость с SQL-стандартами обеспечивает совместимость с другими системами и упрощает процесс миграции данных.

В результате выбор PostgreSQL в качестве базы данных для этого веб-приложения гарантирует надежность, высокую производительность и гибкость работы с данными.

#### 4.2.2 Схема базы данных

На рисунке 5 представлена ER-диаграмма базы данных веб-приложения.

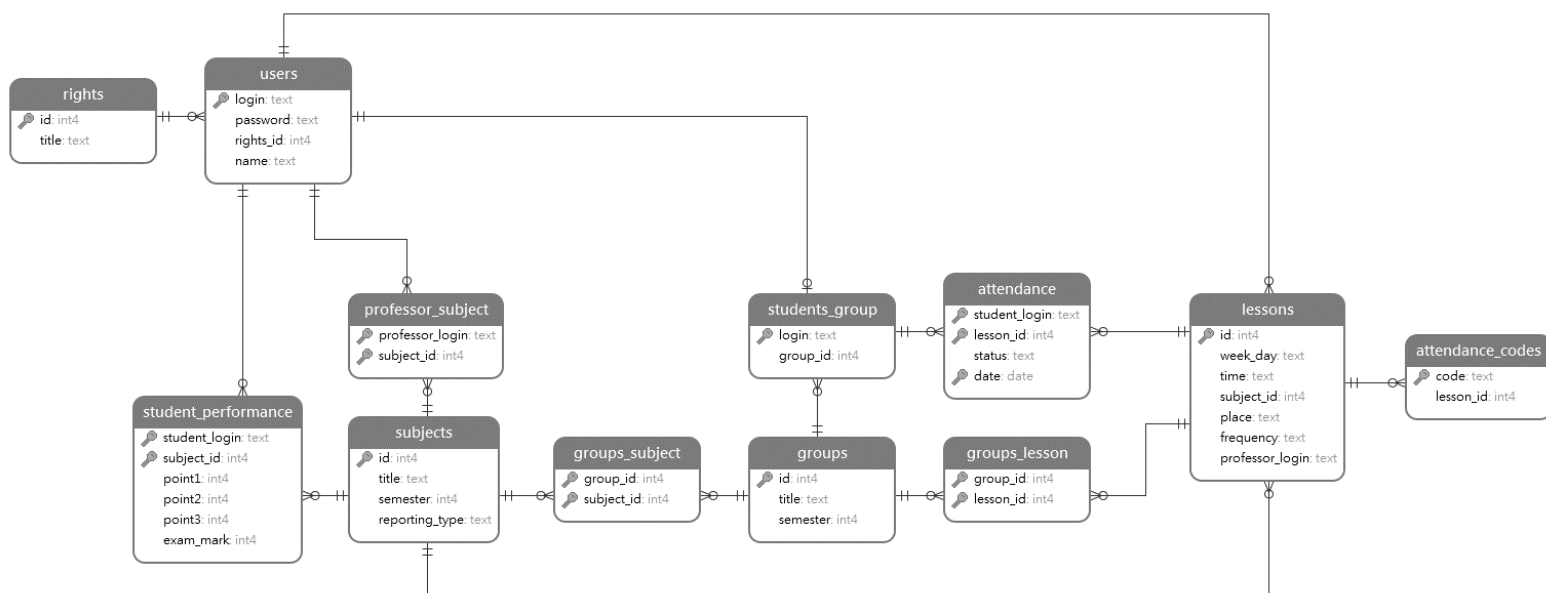


Рисунок 5 – ER-диаграмма базы данных

### 4.2.3 Диаграмма классов

Рассмотрим диаграмму классов, созданную для визуализации структуры веб-приложения.

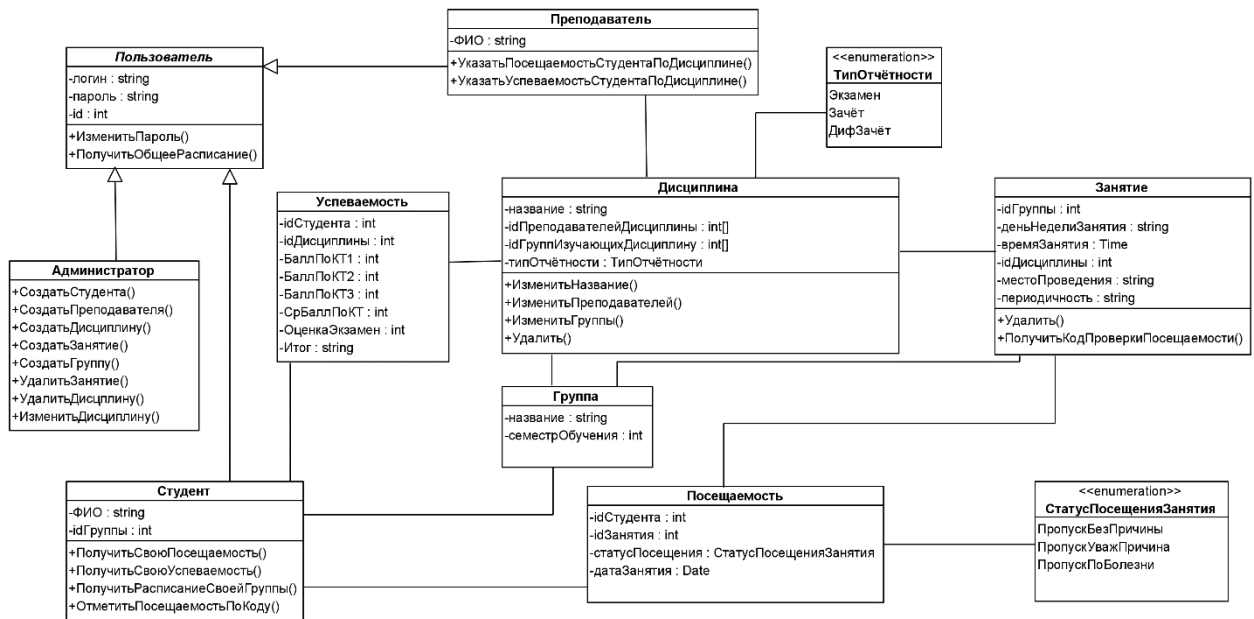


Рисунок 6 – Диаграмма классов

### 4.2.4 Диаграмма прецедентов

Рассмотрим диаграмму использования для неавторизованного пользователя.

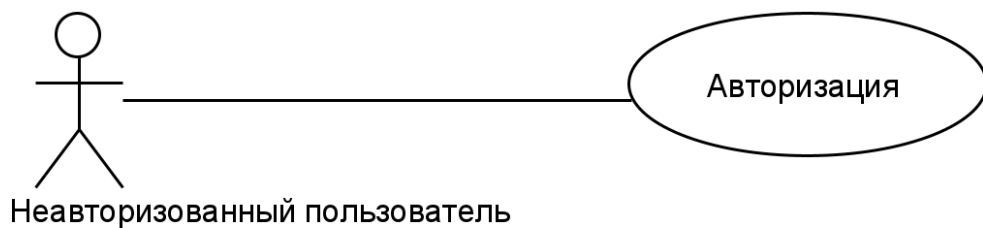


Рисунок 7 – Use-case диаграмма для неавторизованного пользователя



Также рассмотрим диаграмму использования для авторизованного пользователя.

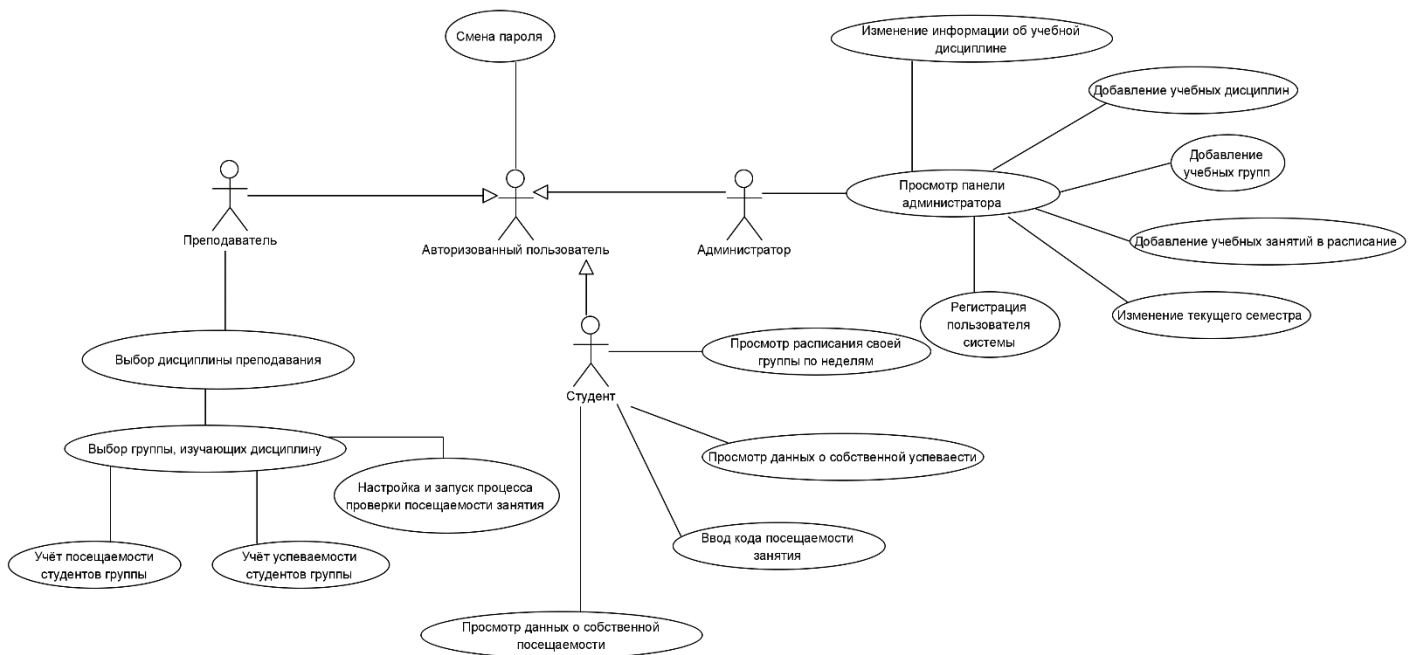


Рисунок 8 – Use-case диаграмма для авторизованного пользователя

#### 4.2.5 IDEF0 диаграмма

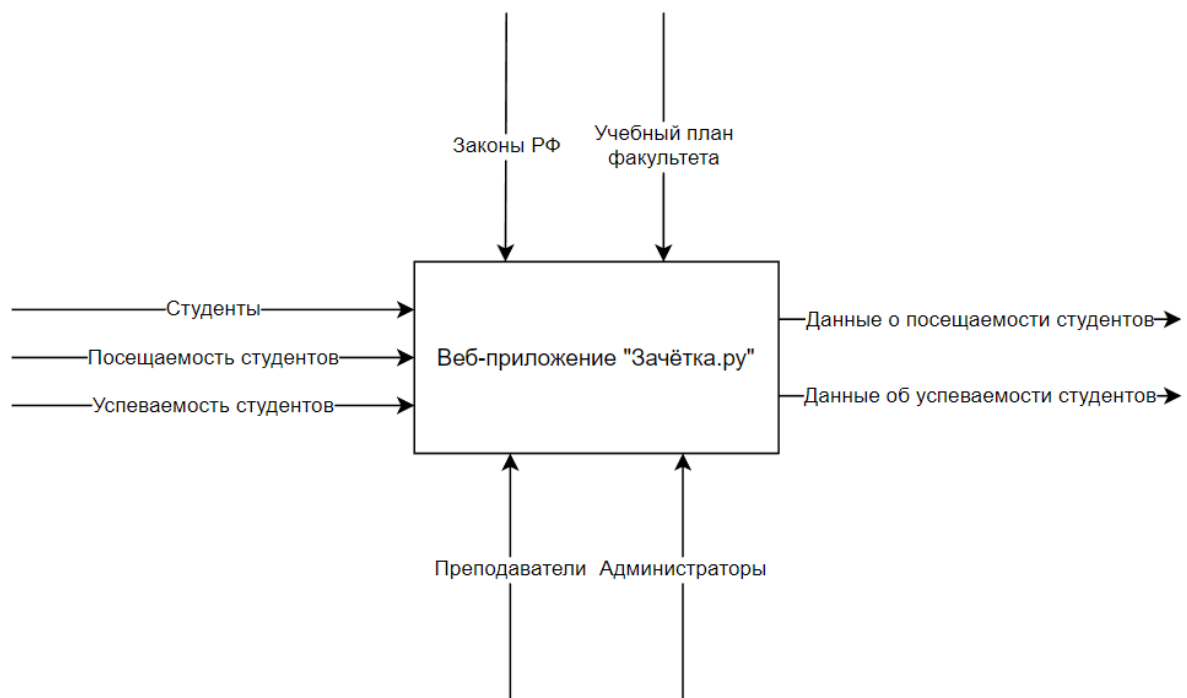


Рисунок 9 – IDEF0 диаграмма веб-приложения

## 4.2.6 Диаграмма последовательности

Диаграмма последовательности используется для визуализации взаимодействия объектов и последовательности их действий в рамках определенной функциональности или сценария. Она помогает понять порядок выполнения операций и коммуникацию между объектами, что полезно при проектировании и анализе систем.

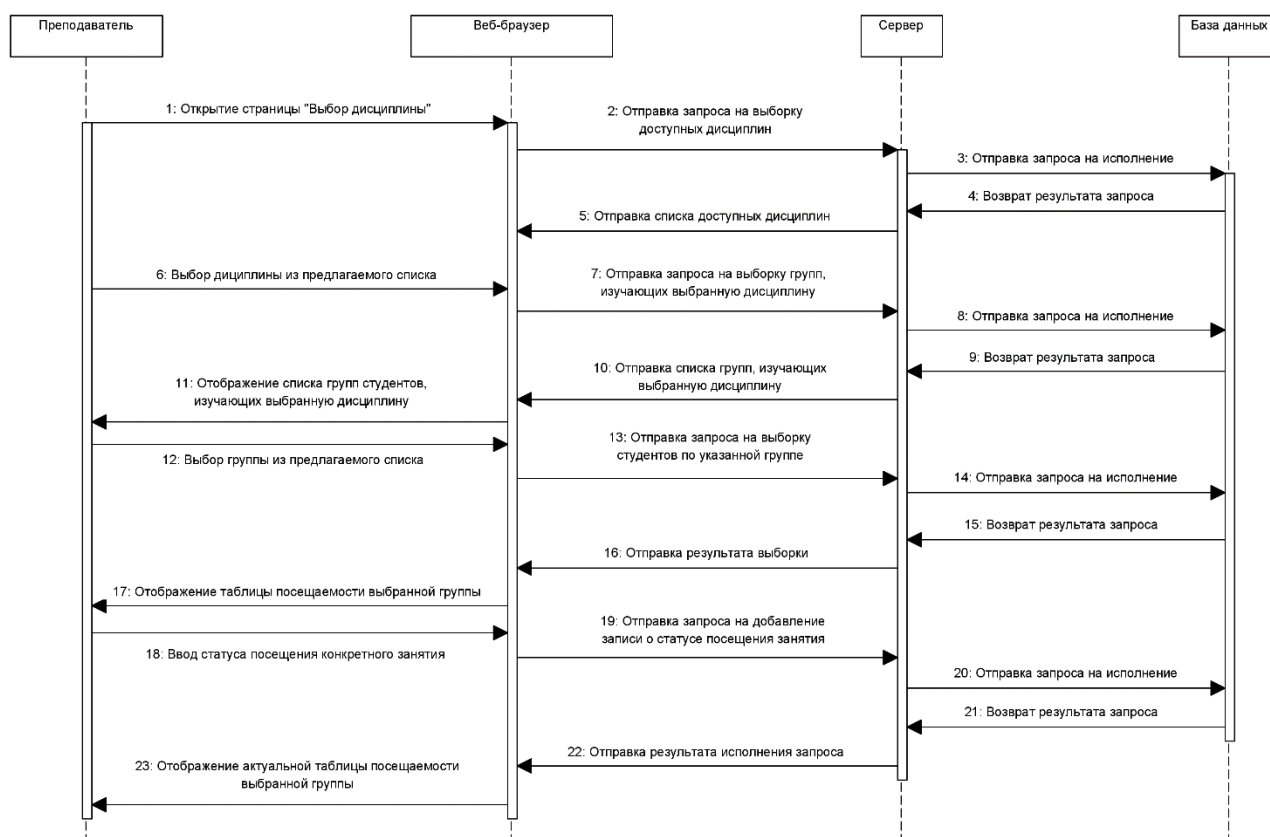


Рисунок 10 – Диаграмма последовательности

## 4.2.7 Диаграмма развёртывания

Диаграмма развёртывания используется для визуализации аппаратного и программного обеспечения, а также их взаимосвязи в системе. Она помогает понять, как компоненты системы развёртываются на аппаратных устройствах

и как они взаимодействуют между собой для обеспечения функциональности приложения или системы.

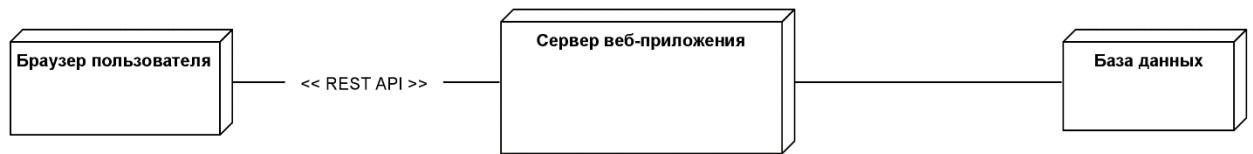


Рисунок 11 – Диаграмма развёртывания веб-приложения

#### 4.2.8 Диаграмма потока данных

Диаграмма потока данных используется для моделирования потоков информации в системе и показа их источников, приемников и преобразований. Она помогает визуализировать, как данные перемещаются через систему, отображая процессы обработки, хранения и передачи информации. Это полезный инструмент для анализа и оптимизации потоков данных в системе.

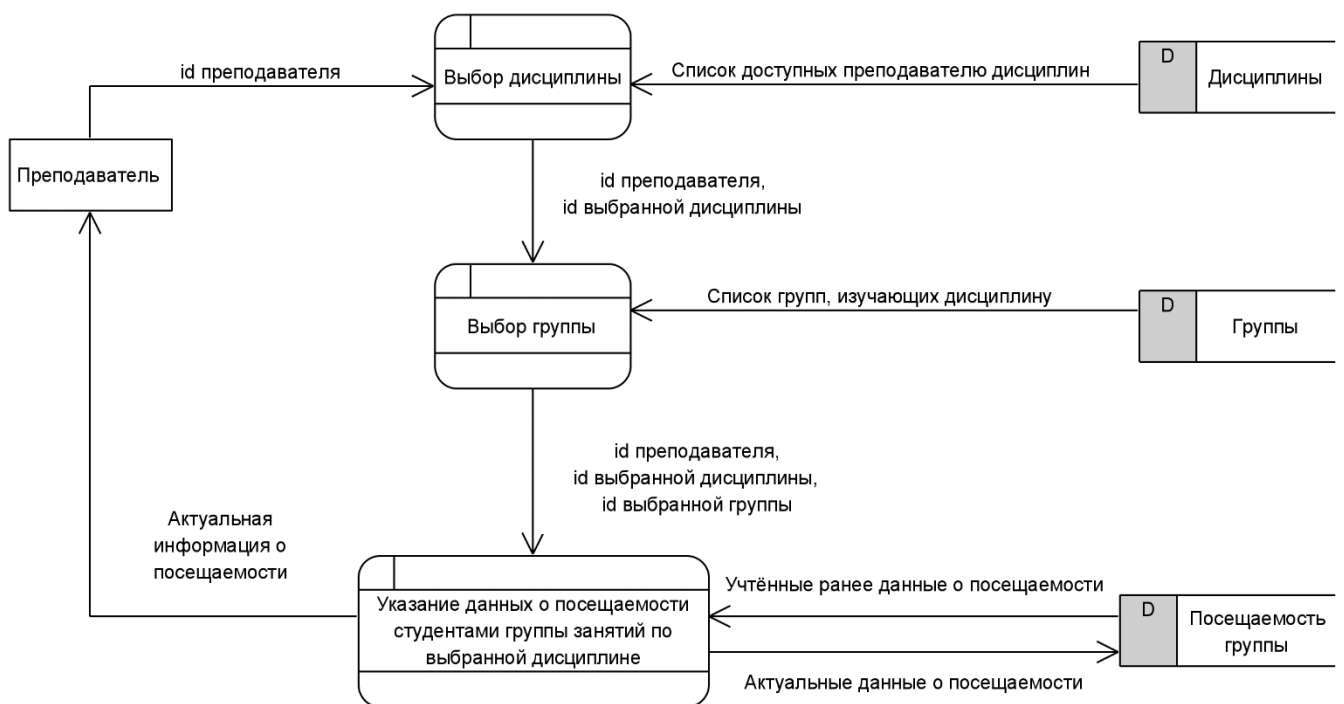


Рисунок 12 – Диаграмма потока данных (DFD)

## 4.2.9 Диаграмма активностей

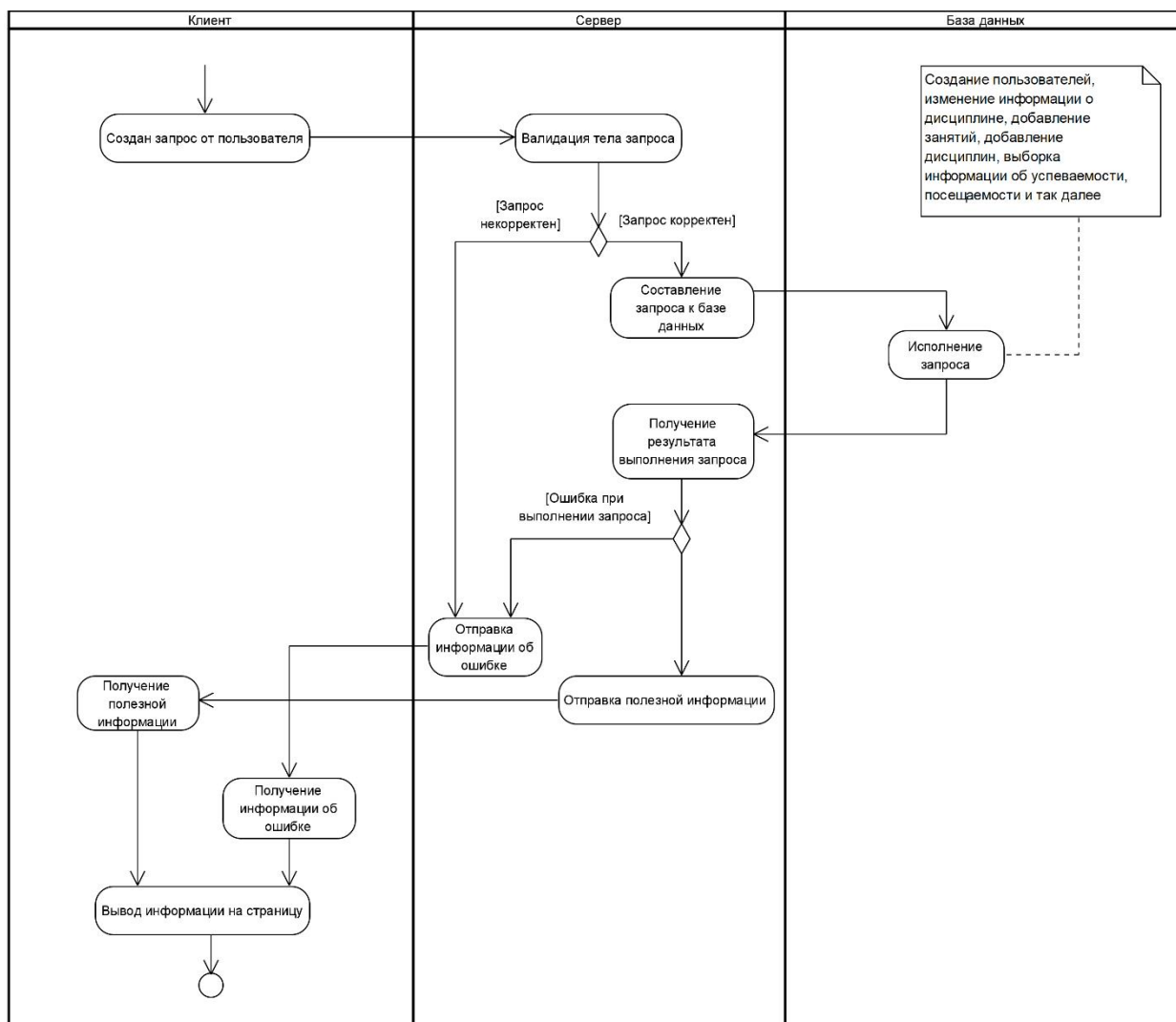


Рисунок 13 – Диаграмма активностей

## 4.3 Серверная часть

### 4.3.1 Основные положения

Основной фреймворк сервера – NestJS.

Запросы разных типов поступают на соответствующие ранее упомянутой базе данных контроллеры, которые передают обработку бизнес-логики сервисам. Сервисы же, в свою очередь, взаимодействуют с репозиториями и сущностями. По окончании работы метода сервиса результат возвращается в контроллер, а далее – в тело-ответ сервера.

Функциональность сервера была организована на отдельные модули, которые являются независимыми компонентами. Это дает возможность без

проблем удалять, заменять или добавлять новые модули, не влияя на остальную часть приложения. Каждый модуль представлен отдельной папкой в основной директории исходного кода сервера. Чтобы использовать определенный модуль, его необходимо импортировать в файл «app.module.ts» и добавить в массив «imports».

Таким образом, данная структура модулей обеспечивает гибкость и удобство в управлении функциональностью сервера.

#### **4.3.2 Развёртывание**

Для развёртывания серверной части приложения был использован хостинг Render [6]. Render предоставляет облачную платформу для размещения приложений и включает в себя необходимые ресурсы и сервисы, включая базу данных PostgreSQL.

Клиентская часть веб-приложения доступна по ссылке: <https://zachetka-ru.onrender.com>.

Серверная часть веб-приложения доступна по ссылке <https://zachetka-ru-server-ax9m.onrender.com>.

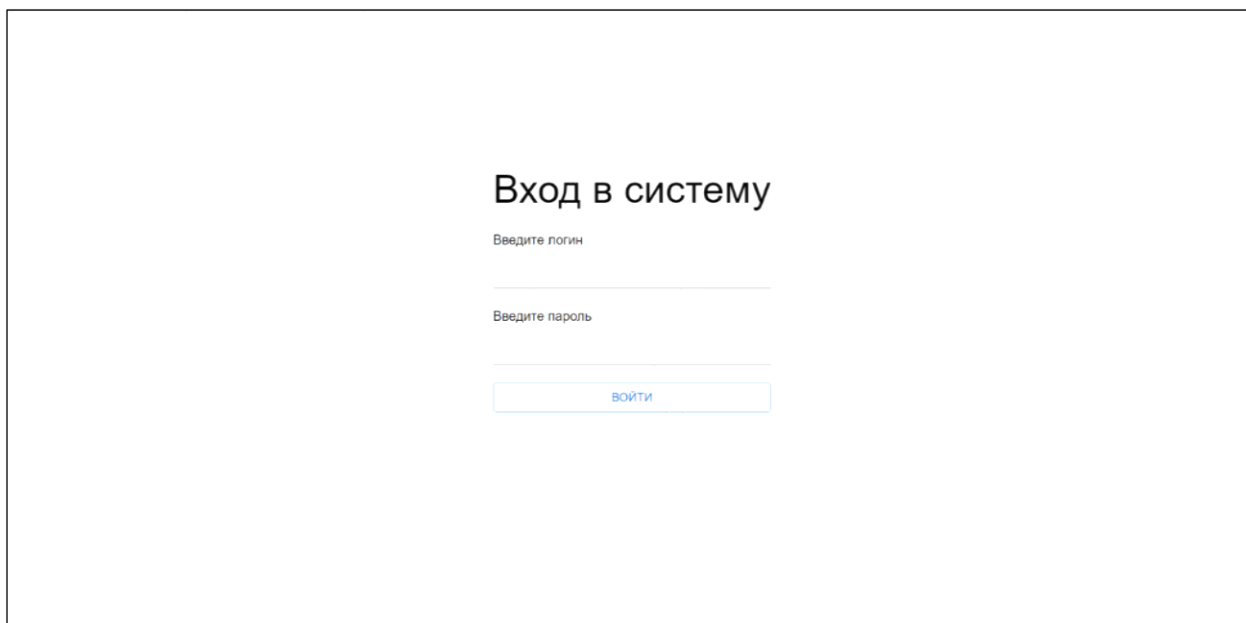
Также было настроено автоматическое развёртывание production версии веб-приложения на этом хостинге. При обнаружении обновлений в ветке main, актуальная версия веб-приложения автоматически разворачивается на хостинге Render.

#### **4.4 Клиентская часть**

Клиентская часть приложения обеспечивает пользовательский интерфейс и взаимодействие с серверной частью, позволяя отображать данные и выполнять операции по запросу пользователя.

#### 4.4.1 Графический пользовательский интерфейс

При входе в веб-приложение пользователя встречает страница авторизации (рисунок 14). На ней пользователь может войти в свой аккаунт по ранее полученным данным.



The image shows a login page with a white background and a thin black border. At the top center, the title 'Вход в систему' is displayed in a bold, black, sans-serif font. Below the title, there are two input fields. The first field is preceded by the text 'Введите логин' and the second by 'Введите пароль', both in a small, grey, sans-serif font. Each input field has a thin grey border. Below the second input field, there is a blue button with the word 'ВОЙТИ' in white, uppercase, sans-serif font.

Рисунок 14 – Страница авторизации

После процесса авторизации, пользователь попадает на определённую стартовую страницу. У всех предусмотренных типов пользователей она разная. Далее рассмотрим каждый тип по отдельности.

Администратор попадает на «панель администратора» (рисунок 15), откуда он может управлять системой.

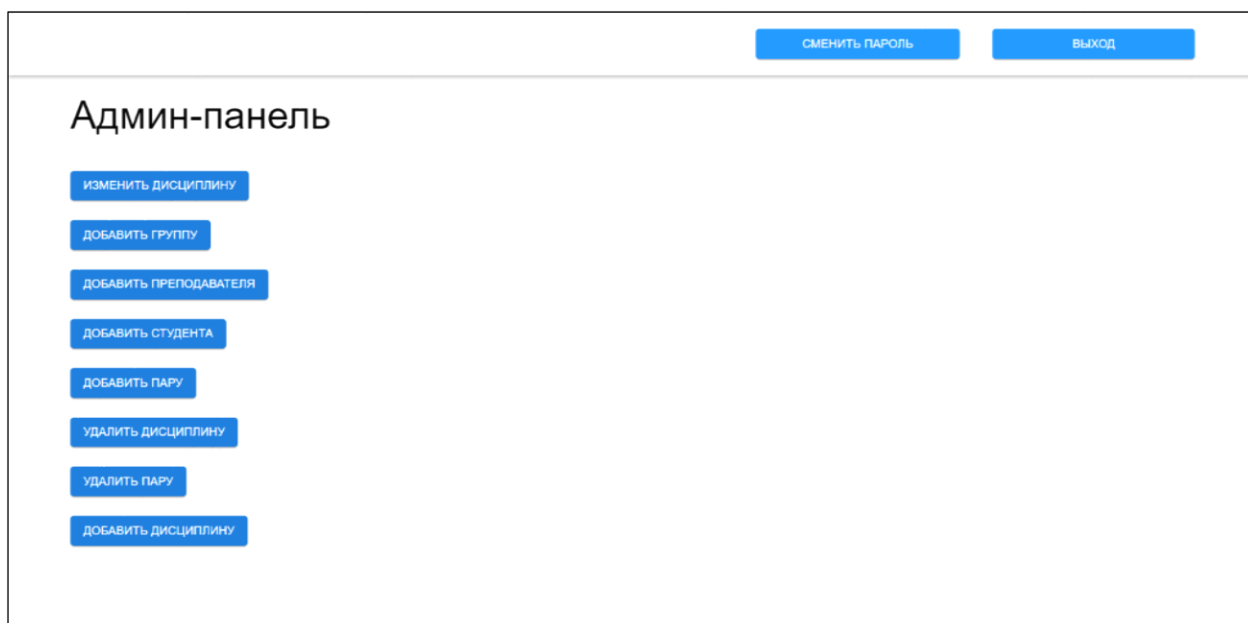


Рисунок 15 – Страница «панель администратора»

Преподаватель попадает на страницу «выбор дисциплины» (рисунок 16), на которой он может выбрать дисциплину и перейти на страницу «выбор группы» (рисунок 17).

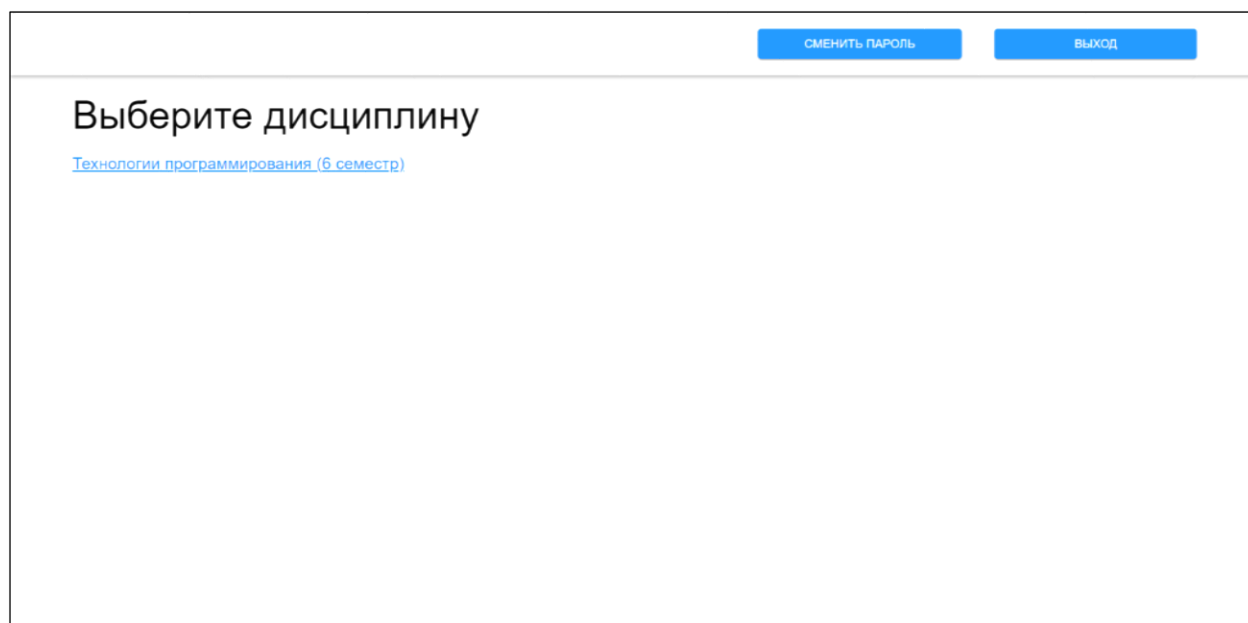


Рисунок 16 – Страница «выбор дисциплины»

Рисунок 17 – Страница «выбор группы»

На странице «выбор группы» преподаватель может выбрать любую группу, тем самым открыв станицу «учёт посещаемости по дисциплине» для этой группы (рисунок 18).

**Учёт посещаемости**

[Технологии программирования \(6 семестр\)](#) / [Группа 1](#) / [Учёт посещаемости](#) ОТКРЫТЬ ОЦЕНКИ

Общее количество занятий: 20

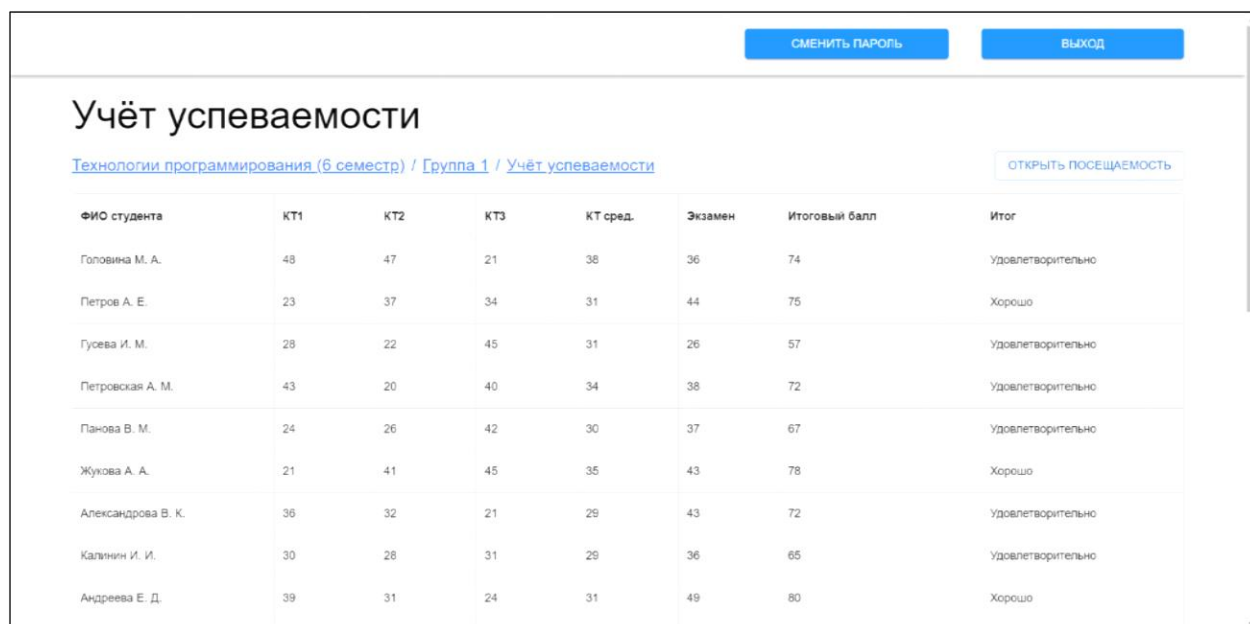
ФИО студента	1.12 15:00	2.12 8:00	6.12 8:00	7.12 8:00	7.12 8:00	10.12 15:00	11.12 8:00	20.12 8:00	2	Н	П	Б	Сумма
Головина М. А.				Н					3	1	0	4	
Петров А. Е.			Н			Н			2	0	0	2	
Гусева И. М.				Б					Н	1	0	3	4
Петровская А. М.									1	1	1	3	
Панова В. М.		Н	П				Н		2	2	0	4	
Жукова А. А.			П	Н				П	Б	2	3	2	7
Александрова В. К.		Н					Б		1	0	1	2	
Калинин И. И.				П	Н			Б	2	3	2	7	

[2 группа](#)  
[3 группа](#)  
[4 группа](#)

Рисунок 18 – Страница «учёт посещаемости по дисциплине»



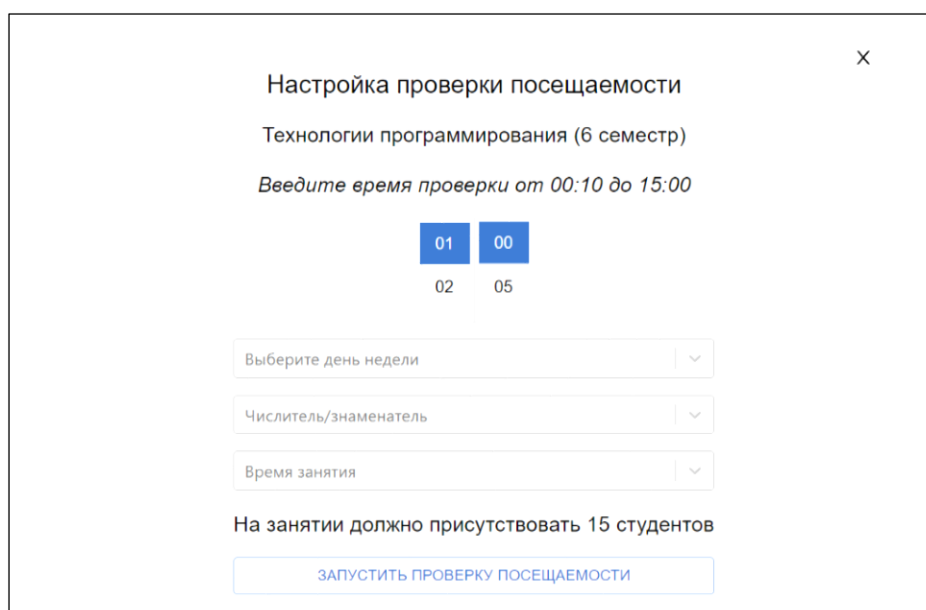
С этой страницы он может перейти на «учёт успеваемости по дисциплине» (рисунок 19).



ФИО студента	КТ1	КТ2	КТ3	КТ сред.	Экзамен	Итоговый балл	Итог
Головина М. А.	48	47	21	38	36	74	Удовлетворительно
Петров А. Е.	23	37	34	31	44	75	Хорошо
Гусева И. М.	28	22	45	31	26	57	Удовлетворительно
Петровская А. М.	43	20	40	34	38	72	Удовлетворительно
Панова В. М.	24	26	42	30	37	67	Удовлетворительно
Жукова А. А.	21	41	45	35	43	78	Хорошо
Александрова В. К.	36	32	21	29	43	72	Удовлетворительно
Калнин И. И.	30	28	31	29	36	65	Удовлетворительно
Андреева Е. Д.	39	31	24	31	49	80	Хорошо

Рисунок 19 – Страница «учёт успеваемости по дисциплине»

Также со страницы «выбор группы» преподавателю доступна функция автоматической проверки посещаемости. При нажатии кнопки «проверка посещаемости» откроется всплывающее окно с настройкой отметки посещаемости (рисунок 20).



Настройка проверки посещаемости

Технологии программирования (6 семестр)

Введите время проверки от 00:10 до 15:00

01 00

02 05

Выберите день недели

Числитель/знаменатель

Время занятия

На занятии должно присутствовать 15 студентов

ЗАПУСТИТЬ ПРОВЕРКУ ПОСЕЩАЕМОСТИ

Рисунок 20 – Всплывающее окно с настройкой автоматической проверки посещаемости

После настройки и нажатия кнопки «запустить проверку посещаемости», данное всплывающее окно сменится на окно с процессом проверки посещаемости (рисунок 21).

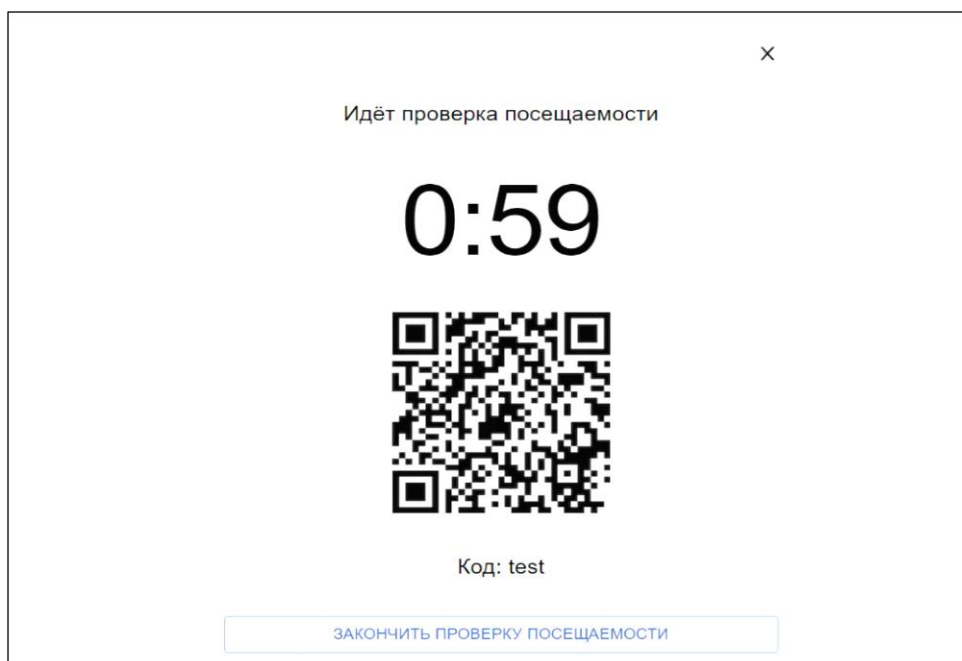


Рисунок 21 – Всплывающее окно с процессом проверки посещаемости

При завершении или прерывании процесса проверки посещаемости, путём нажатия соответствующей кнопки, всплывающее окно сменится на окно с итогом проверки посещаемости (рисунок 22).

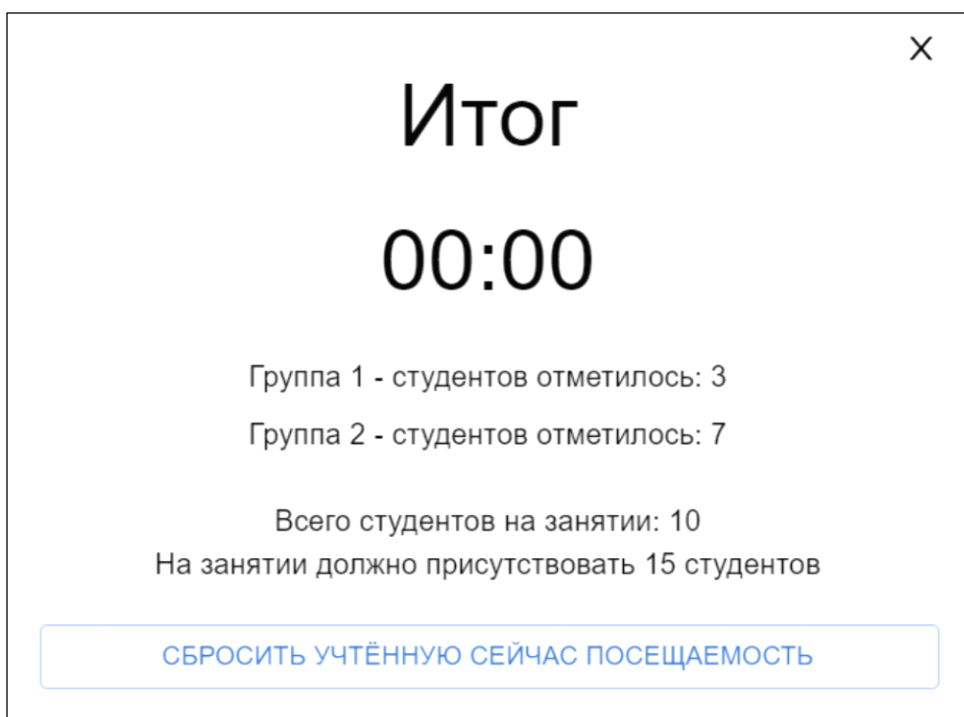


Рисунок 22 – Всплывающее окно с итогом проверки посещаемости

Студент после авторизации попадает на страницу «еженедельное расписание» (рисунок 23).

СМЕНИТЬ ПАРОЛЬ ВЫХОД

## Еженедельное расписание

Иванов Иван Иванович

КОД ПОСЕЩАЕМОСТИ

ОТКРЫТЬ  
УСПЕВАЕМОСТЬ

← Месяц Март →

27	08:00 - 09:35		
	09:45 - 11:20		
	11:30 - 13:05		
	13:25 - 15:00		
	15:10 - 16:45		
пн	16:55 - 18:30		
	18:40 - 20:00		
	20:10 - 21:30		
28	08:00 - 09:35	Квантовая теория Стадная Р. П. 303п	Н
	09:45 - 11:20	Обработка изображений Фертиков В. В. 297	О
	11:30 - 13:05	Информационные поисковые системы Сычев А. В. 314п	
	13:25 - 15:00	Информационные поисковые системы Сычев А. В. 385	
	15:10 - 16:45		

← Месяц Март →

30	08:00 - 09:35	Технологии интернет вещей Зуев С. А. 307п	Н
	09:45 - 11:20	Обработка изображений Фертиков В. В. 297	
	11:30 - 13:05	ПАОД Фертиков В.В. 292	
	13:25 - 15:00		
	15:10 - 16:45		
чт	16:55 - 18:30		
	18:40 - 20:00		
	20:10 - 21:30		
31	08:00 - 09:35	Инфокоммуникационные системы и сети Коваль А.С. (ДО)	Н
	09:45 - 11:20	Обработка изображений Фертиков В. В. 297	
	11:30 - 13:05		
	13:25 - 15:00	Информационные поисковые системы Сычев А. В. 385	
	15:10 - 16:45		

Рисунок 23 – Страница «еженедельное расписание»

С данной страницы он может перейти на страницу «моя успеваемость» (рисунок 24) или нажать на кнопку «код посещаемости», чтобы открыть всплывающее окно для введения кода посещаемости (рисунок 25). Со страницы «моя успеваемость» также можно открыть данное всплывающее окно.

СМЕНИТЬ ПАРОЛЬ ВЫХОД

## Моя успеваемость

Иванов Иван Иванович

КОД ПОСЕЩАЕМОСТИ

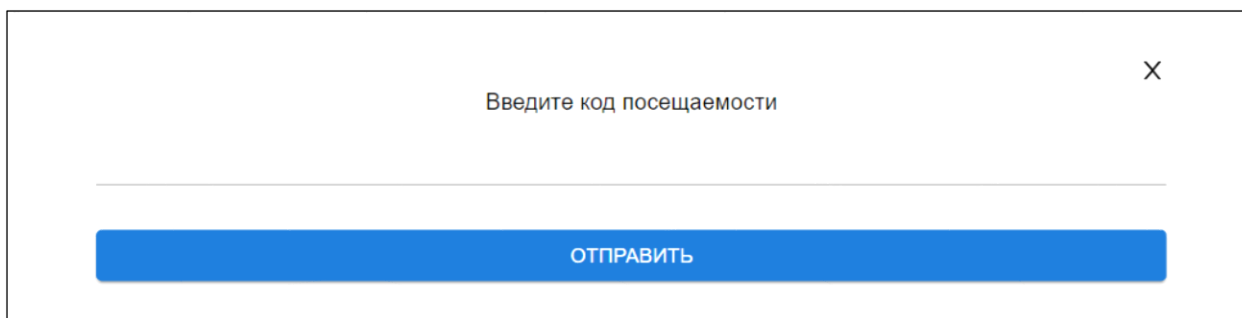
ОТКРЫТЬ  
РАСПИСАНИЕ

Семестр	Курс	Дисциплина	Отчётность	Преподаватель	КТ 1	КТ 2	КТ 3	Средний балл	Экзамен	Итог
6	3	Информационные технологии	Экзамен	Вахтин А. А.	40	40	40	40	40	Хорошо
6	3	Инфокоммуникационные системы и сети	Экзамен	Коваль А. С.	25	32	42	33	33	Удовлетворительно
6	3	Технологии программирования	Экзамен	Тарасов В. С.	45	25	38	36	35	Хорошо
6	3	Квантовая теория	Экзамен	Запрыгаев С. А.	25	25	25	25	25	Удовлетворительно
6	3	Информационно-поисковые системы	Зачёт	Сычев А. В.	35	35	35	35	-	Зачтено
6	3	Обработка изображений	Зачёт с оценкой	Фертиков В. В.	36	35	37	36	-	Хорошо
6	3	Механика и оптика	Зачёт с оценкой	Фертиков В. В.	40	40	40	40	-	Хорошо

## 4.5 Тестирование

Исходя из выбранных методов и особенностей проекта, было принято решение провести ручное тестирование. Данная форма тестирования позволяет более гибко и детально проверить функциональность и пользовательский интерфейс приложения, а также обнаружить потенциальные проблемы и дефекты.

Ниже представлены таблицы, содержащие в себе заранее составленные тест-кейсы, состоящие из определенных шагов. Данные таблицы являются



The image shows a modal window with a white background and a thin grey border. In the top right corner, there is a small grey 'X' icon for closing the window. The main text inside the window is 'Введите код посещаемости' (Enter the visit code) in a dark grey font. Below this text is a horizontal line indicating where to enter the code. At the bottom of the window is a solid blue button with the white text 'ОТПРАВИТЬ' (SEND).

руководством для тестировщика. В процессе тестирования каждый из шагов был выполнен, и результаты на каждом шаге соответствовали ожидаемым. Это подтверждает работоспособность и соответствие разрабатываемого веб-приложения требованиям.

Тестирование, основанное на предоставленном руководстве, позволило проверить различные функциональные возможности приложения, а также убедиться в его корректной работе.

Ручное тестирование предоставляет возможность более тщательно и внимательно проверить каждый аспект приложения, а также принять во внимание индивидуальные особенности и потребности пользователей. Такой подход способствует повышению качества и надежности приложения перед его внедрением и использованием в реальных условиях.

Таблица 1 – Тест-кейс «Поведение системы при провали авторизации»

<b>Шаг</b>	<b>Ожидаемый результат</b>
Ввести заведомо неправильный логин и/или пароль	Введённые данные отображаются в полях ввода
Нажать кнопку входа	На экране авторизации появляется сообщение о том, что введён неверный логин или пароль

Таблица 2 – Тест-кейс «Вход в аккаунт пользователя Администратор»

<b>Шаг</b>	<b>Ожидаемый результат</b>
Ввести заранее полученный логин и пароль	Введённые данные отображаются в полях ввода
Нажать кнопку входа	Экран авторизации заменяется на панель администратора

Таблица 3 – Тест-кейс «Вход в аккаунт пользователя Преподаватель»

<b>Шаг</b>	<b>Ожидаемый результат</b>
Ввести заранее полученный логин и пароль	Введённые данные отображаются в полях ввода
Нажать кнопку входа	Экран авторизации заменяется на страницу «выбор дисциплины»

Таблица 4 – Тест-кейс «Вход в аккаунт пользователя Студент»

<b>Шаг</b>	<b>Ожидаемый результат</b>
Ввести заранее полученный логин и пароль	Введённые данные отображаются в полях ввода
Нажать кнопку входа	Экран авторизации заменяется на страницу «еженедельное расписание»

Таблица 5 – Тест-кейс «Выход из аккаунта»

Шаг	Ожидаемый результат
Повторить шаги «входа в аккаунт»	Экран авторизации заменяется на страницу в соответствии с типом пользователя
Нажать на кнопку «выход» в шапке страницы	Экран предшествующей страницы заменяется на страницу авторизации

Таблица 6 – Тест-кейс «Проверка посещаемости преподавателем»

Шаг	Ожидаемый результат
Повторить шаги «входа в аккаунт»	Экран авторизации заменяется на страницу «выбор дисциплины»
Выбрать нужную дисциплину	Страница «выбор дисциплины» заменяется на страницу «выбор группы»
Нажать кнопку «проверка посещаемости»	Открывается всплывающее окно с настройкой проверки посещаемости
Настроить проверку посещаемости и нажать кнопку «запустить проверку посещаемости»	Во всплывающем окне появляется таймер с обратным отсчётом, QR-код для сканирования студентами и символьный код для введения студентами
Дождаться окончания проверки посещаемости	Во всплывающем окне выводится итог проверки посещаемости

Таблица 7 – Тест-кейс «Прерывание проверки посещаемости преподавателем»

Шаг	Ожидаемый результат
Повторить шаги «входа в аккаунт»	Экран авторизации заменяется на страницу «выбор дисциплины»
Повторить шаги «проверки посещаемости преподавателем» до	Во всплывающем окне появляется таймер с обратным отсчётом, QR-код

момента запуска проверки посещаемости	для сканирования студентами и символьный код для введения студентами
Нажать кнопку «закончить проверку посещаемости»	Во всплывающем окне выводится итог проверки посещаемости

Таблица 8 – Тест-кейс «Сброс итога проверки посещаемости преподавателем»

Шаг	Ожидаемый результат
Повторить шаги «входа в аккаунт»	Экран авторизации заменяется на страницу «выбор дисциплины»
Повторить шаги «проверки посещаемости преподавателем»	Во всплывающем окне выводится итог проверки посещаемости
Нажать кнопку «сбросить учтённую сейчас посещаемость»	Только что учтённые данные обнуляются и всплывающее окно с итогами проверки посещаемости заменяется на всплывающее окно с настройкой проверки посещаемости

Таблица 9 – Тест-кейс «Отметка посещаемости студентом через QR код»

Шаг	Ожидаемый результат
Навести камеру мобильного устройства на QR-код и перейти по содержащейся в нем ссылке	Необходимость авторизации с последующим сообщением об успешном отметке посещаемости

Таблица 10 – Тест-кейс «Отметка посещаемости студентом, путём введения символьного кода»

Шаг	Ожидаемый результат
Повторить шаги «входа в аккаунт»	Экран авторизации заменяется на страницу «еженедельное расписание»

Нажать на кнопку «код посещаемости»	Страница «еженедельное расписание» заменяется страницей «ввод кода посещаемости»
Ввести код посещаемости	Введённый код отображается в поле ввода
Нажать кнопку «Отправить»	Посещаемость отмечена

Помимо ручного тестирования, в GitHub Actions был добавлен workflow file, в котором настроен запуск сборки клиента и сервера, а также выполнение проверки кода линтерами. Данные проверки автоматически запускаются при открытии нового pull request'а, а также при любом его изменении.

В случае возникновения ошибки на любом из этапов, pull request не проходит проверку и не может быть слит с целевой веткой.



## **Заключение**

В заключение, данная курсовая работа представила разработку и реализацию веб-приложения, которое объединяет функции автоматического учёта посещаемости, учёта успеваемости и просмотра расписания для студентов.

Готовое приложение было успешно развернуто на хостинге Render, который обеспечивает надежное и гибкое размещение веб-приложений. Это обеспечивает пользователям доступ к приложению в соответствии с заявленными требованиями технического задания.

В результате данной работы было создано веб-приложение, которое объединяет функции автоматического учёта посещаемости, контроля успеваемости и просмотра расписания для студентов.

В дальнейшем развитии данного проекта возможно расширение его функциональности, добавление новых модулей и интеграция с другими системами.

В целом, данная курсовая работа представляет важный шаг в разработке современных и эффективных веб-приложений для образовательных учреждений.

### **Список использованных источников**

1. TypeScript Documentation // TypeScript URL: <https://www.typescriptlang.org/docs/> (дата обращения: 05.04.23).
2. Documentation | NestJS // NestJS URL: <https://docs.nestjs.com/> (дата обращения: 05.04.23).
3. Docs | Next.js // Next.js URL: <https://nextjs.org/docs> (дата обращения: 06.04.23).
4. Getting Started | Create React App // React URL: <https://create-react-app.dev/docs/getting-started/> (дата обращения: 08.04.23).
5. Material UI - Overview // MUI URL: <https://mui.com/material-ui/getting-started/overview/> (дата обращения: 08.04.23).
6. Cloud Application Hosting for Developers | Render // URL: <https://render.com/> (дата обращения: 20.09.23).