**СОДЕРЖАНИЕ**

| Введение |  | 3 |
| --- | --- | --- |
| Описание специфики предметной области. |  | 4 |
| Актуальность проекта |  | 7 |
| Идея проекта |  | 10 |
| Требования к проекту и учитываемые ограничения. |  | 13 |
| Теоретическое обоснование проекта |  | 16 |
| Практическое обоснование проекта |  | 18 |
| Собственные теоретические разработки при реализации проекта. |  | 20 |
| Особенности разработки проекта. Уникальные решения |  | 23 |
| Функциональные возможности проекта |  | 28 |
| Пользовательский интерфейс проекта |  | 31 |
| Организация тестирования проекта. |  | 35 |
| Библиографический список |  | 38 |

### 

### **Введение**

### Современная система высшего образования активно внедряет принципы индивидуализации и адаптации образовательных программ под запросы студентов и требования рынка труда. Одним из ключевых направлений этой трансформации является переход к моделям, подобным "2+2+2", реализуемым Тюменским государственным университетом (ТюмГУ). В рамках этой модели студенты получают возможность проектировать свои образовательные траектории, включая выбор элективных курсов, которые становятся важным инструментом персонализации обучения.

### Тем не менее, управление элективными курсами остаётся сложным и ресурсоёмким процессом. Центр управления индивидуальными образовательными траекториями (Центр УИОТ), являющийся основным оператором системы Modeus, сталкивается с рядом проблем, связанных с предварительной сверкой данных и распределением нагрузки перед внесением их в платформу. Эти задачи требуют значительных временных затрат и высокой точности, что создаёт риски возникновения ошибок.

### Проект "Элиз" направлен на автоматизацию предварительных этапов работы с элективами. Его цель — сократить временные затраты, улучшить точность данных, повысить прозрачность процессов и упростить подготовку сводной документации.

### **Описание специфики предметной области**

Элективные курсы представляют собой важный элемент современного образования, способствующий гибкости и персонализации образовательных программ. Эти курсы позволяют студентам формировать индивидуальные траектории обучения, дополняя основной учебный план дисциплинами, соответствующими их карьерным интересам. Тюменский государственный университет активно развивает эту систему в рамках модели "2+2+2", предоставляя студентам возможность выбирать из сотен элективных курсов.

#### **Ключевые особенности элективных курсов:**

* **Гибкость учебных планов.** Образовательные учреждения разрабатывают вариативные программы и модули, которые учитывают академические и профессиональные интересы студентов.
* **Список дисциплин, из которых можно сделать выбор.** Каждый семестр в нашем университете предлагаются новые курсы, что требует их интеграции в общий график занятий.
* **Рациональное использование ограниченных ресурсов.** Нужно учитывать загруженность преподавателей, доступность аудиторий и технического оборудования.

#### **Проблемы управления элективными курсами:**

1. **Сложность координации.** Возникают трудности с предотвращением конфликтов расписаний неосновного блока, учётом нагрузки преподавателей, равномерным распределением студентов и интеграцией элективов в основное расписание.
2. **Высокий объём данных.** Управление курсами требует обработки и анализа множества параметров — от специалиста и численности студентов до требований к аудиториям и другим ресурсам.
3. **Фрагментированное хранение данных.** Использование отдельных Excel-файлов для управления элективами может приводить к:
   * Дублированию информации;
   * Ошибкам при синхронизации данных;
   * Затруднению доступа к актуальной информации.

В итоге, отсутствие единой платформы затрудняет оперативное принятие решений и координацию между участниками процесса.

**Потребности в автоматизации:**

Современные образовательные учреждения, включая ТюмГУ, нуждаются в системах, которые:

1. **Централизуют хранение данных.** Это помогает устранить дублирование информации, минимизировать ошибки и повысить эффективность управления.
2. **Автоматизируют проверку данных.** Внедрение процессов ETL (Extract, Transform, Load) обеспечит высокую точность данных.
3. **Обеспечивают гибкий анализ.** Динамическое представление данных и отчёты позволяют более эффективно оптимизировать управление элективами.

Примером частичного решения являются LMS (Learning Management Systems, системы управления обучением), которые предоставляют базовый функционал для управления учебным процессом. Однако такие системы, как Moodle или Canvas, не решают задач согласования нагрузки преподавателей и управления элективными курсами, а также не предоставляют детализированного анализа нагрузки.

#### **Цель автоматизации:**

Проект направлен на:

* **Сокращение временных затрат:** автоматизация рутинных операций позволит снизить нагрузку на административный персонал.
* **Улучшение точности данных:** устранение ошибок за счёт стандартизации и алгоритмов проверки.
* **Повышение прозрачности:** предоставление актуальной информации в режиме реального времени.
* **Расширение интеграции:** система будет гибкой и способной к взаимодействию с образовательными платформами.

#### **Перспективы внедрения:**

Системный подход к автоматизации управления элективными курсами обеспечит:

* Оптимизацию процессов планирования и администрирования;
* Формирование единой базы данных для долгосрочного анализа и прогнозирования;
* Устойчивость образовательного процесса за счёт адаптивной структуры системы.

### **Актуальность проекта.**

#### **Текущее состояние управления элективными курсами.**

В университетах управление элективными курсами остаётся сложным и трудоёмким процессом. Основная часть работы, связанной со сбором и обработкой данных элективов, выполняется вручную. На данный момент менеджеры по управлению индивидуальными образовательными траекториями (УИОТ) взаимодействуют с преподавателями через электронные письма и большие онлайн-таблицы, что, как правило, обрабатываются вручную, что приводит к следующим проблемам:

* **Дублирование данных.** Информация часто вводится повторно, что увеличивает риск ошибок.
* **Потеря данных.** Файлы могут быть утеряны или устареть, что затрудняет анализ.
* **Значительные временные затраты.** Процесс ручной обработки данных занимает много времени и снижает эффективность.

Эти недостатки замедляют административные процессы и затрудняют создание полной картины для анализа и стратегического планирования.

#### **Пользователи системы**

Ключевыми пользователями разрабатываемой системы являются менеджеры УИОТ, которые выполняют следующие задачи:

* Планирование расписания элективных курсов.
* Координация преподавательского состава и ресурсов.
* Анализ эффективности реализации курсов.

Существующая система ограничивает их возможности из-за фрагментации данных и отсутствия аналитических инструментов.

#### **Проблемы текущей системы**

1. **Высокая трудоёмкость.** Ручная обработка больших объёмов данных требует значительных усилий.
2. **Фрагментация данных.** Информация хранится в разных таблицах, что приводит к несогласованности и дублированию данных.
3. **Отсутствие аналитических инструментов.** Современные задачи управления требуют автоматизированных решений для анализа и визуализации данных.
4. **Недостаточная прозрачность.** Отсутствие единой базы данных затрудняет доступ к актуальной информации.
5. **Зависимость от ручных операций.** Отсутствие автоматизации повышает вероятность ошибок и делает систему уязвимой к сбоям.

#### **Требования к системе**

Проектируемая система должна соответствовать следующим требованиям:

* **Централизация данных.** Единая база данных, содержащая информацию об элективах, обеспечит удобный доступ и согласованность данных.
* **Автоматизация процессов.** Минимизация ручного труда, автоматизация аспекты сбора, проверки и анализа данных.
* **Интеграция аналитики.** Пользователи ожидают наличие инструментов для построения отчётов и анализа данных.
* **Гибкость и адаптивность.** Система должна быть масштабируемой и способной адаптироваться к изменениям нормативных и организационных требований.

#### **Обоснование актуальности**

С увеличением числа элективных курсов традиционные методы управления становятся менее эффективными.

* **Необходимость снижения ошибок.** Автоматизация процессов минимизирует человеческий фактор, снижая вероятность ошибок.
* **Повышение требований к аналитике.** Современные университеты нуждаются в инструментах для анализа и оценки эффективности образовательных программ.
* **Сокращение временных затрат.** Оптимизация процессов позволит менеджерам сосредоточиться на стратегически важных задачах.
* **Прозрачность процессов.** Централизованная база данных обеспечит доступ к актуальной информации для всех заинтересованных сторон.

#### **Ожидаемые результаты**

Внедрение автоматизированной системы управления элективными курсами позволит:

* Сократить время на выполнение рутинных операций.
* Повысить точность и актуальность данных.
* Упростить взаимодействие между подразделениями университета.
* Улучшить процесс планирования и распределения ресурсов.
* Повысить конкурентоспособность университета за счёт эффективного управления образовательными программами.

Проект направлен на оптимизацию работы менеджеров УИОТ и повышение качества управления элективными курсами, что будет способствовать улучшению образовательного процесса и удовлетворению современных требований.

### **Идея проекта**

Для устранения существующих проблем управления элективными курсами предлагается разработать автоматизированную систему, которая будет базироваться на следующих принципах и функциональных решениях:

1. **Централизация данных.**

Система создаст унифицированную базу данных, аккумулирующую необходимую информацию об элективах, преподавателях и студентах. Это позволит следующее:

* Исключить дублирование данных, сократить временные затраты на их поиск и обновление.
* Упростить интеграцию между подразделениями университета, обеспечивая единую экосистему управления образовательным процессом.
* Предоставить гибкие возможности поиска и фильтрации данных. Например, менеджеры УИОТ смогут оперативно находить информацию о конкретном элективе, преподавателе или группе студентов через интуитивно понятный интерфейс.

1. **Автоматизация процессов.**

Система будет включать модули для автоматизированного сбора, проверки и анализа данных. Ключевые функции автоматизации:

* Упрощение подачи данных преподавателями с использованием стандартизированных форм.
* Автоматическая проверка введённой информации на корректность.
* Генерация уведомлений о необходимости предоставления данных, о сроках подготовки отчётов.
* Ускорение обработки данных, исключая ошибки, связанные с человеческим фактором.

1. **Интеграция аналитических инструментов.**

Встроенные аналитические модули позволят визуализировать данные, формировать отчёты и проводить оценку ключевых показателей эффективности. Среди возможностей:

* Создание графиков, диаграмм и таблиц, которые помогут анализировать популярность элективов, загруженность преподавателей и использование учебных ресурсов.
* Формирование отчётов для управленческих решений, например, распределение студентов по группам или сравнение элективов по показателям востребованности.
* Динамическая оценка заполненности курсов, позволяющая оперативно корректировать планирование.

**4. Гибкость и адаптивность.**

Архитектура системы будет спроектирована с учётом возможных изменений в образовательной среде. Возможность настройки правил распределения студентов, графиков подачи данных и форматов отчётов. Поддержка модульного управления доступом. Например, менеджеры УИОТ получат полный доступ к редактированию и анализу данных, а преподаватели — возможность вводить данные по своим курсам. Масштабируемость системы, что позволит адаптировать её к увеличению числа студентов, курсов и преподавателей.

#### **Практическая ценность и результаты внедрения**

Основная идея проекта заключается в создании функциональной платформы, способной автоматизировать рутинные операции, повысить точность данных и улучшить качество управления элективными курсами. Внедрение системы обеспечит:

* Сокращение времени на выполнение задач, связанных с обработкой и анализом данных.
* Уменьшение количества ошибок при планировании и реализации элективов.
* Повышение прозрачности процессов за счёт единого хранения и обработки информации.
* Улучшение координации между подразделениями университета.
* Создание более эффективной системы управления, соответствующей требованиям современных образовательных стандартов.

Проект направлен на совершенствование управления элективными курсами, что, в свою очередь, поспособствует повышению конкурентоспособности университета и улучшению качества образовательного процесса.

### **Требования к проекту и учитываемые ограничения**

Проектируемая система управления элективными курсами должна учитывать современные вызовы образовательной среды, предоставляя эффективные инструменты для автоматизации процессов, анализа данных и интеграции с существующими информационными платформами. Её разработка направлена на решение комплексных задач, связанных с оптимизацией образовательного процесса, повышением эффективности работы сотрудников и улучшением взаимодействия между всеми участниками образовательной среды.

#### **Требования к системе**

##### **1. Менеджеры УИОТ**

Менеджеры УИОТ, являющиеся ключевыми пользователями системы, требуют функционала, обеспечивающего выполнение их задач с минимальными временными затратами и высокой точностью:

* **Автоматизация рутинных операций.** Система должна заменить или упростить ручной ввод данных автоматизированными процессами. Это включает сбор данных об уже существующих курсах, таких как названия, преподаватели, количество групп, учебные аудитории и их занятость. Автоматизация также должна охватывать проверку предоставленных данных на корректность, формирование аналитических отчётов и распределение студентов по элективам
* **Сокращение временных затрат.** Интеграция функций поиска и фильтрации данных обеспечит быстрый доступ к актуальной информации, позволяя менеджерам сосредоточиться на аналитической работе.
* **Аналитическая поддержка.** Интеграция с визуализацией данных, включая графики и диаграммы, позволит менеджерам принимать решения на основе объективных показателей (популярность курсов и заполненность групп.).
* **Понятный пользовательский интерфейс.** Он должен быть удобным для выполнения задач без необходимости длительного обучения.

##### **2. Преподаватели**

Преподаватели, как поставщики данных о курсах, могут ожидать, что система в будущем упростит выполнение их обязанностей: подачи информации о курсах, включая возможность редактирования ранее введённых данных, уведомления для снижения риска пропуска важных задач, подача и отслеживания статуса заявок на элективы для процесса адаптации образовательного процесса к изменениям и Интеграцию с расписанием

**3. Университет**

Для университета система должна предоставлять стратегические решения, направленные на улучшение качества управления образовательным процессом:

* **Масштабируемость и адаптивность.** Система должна быть готова к росту числа пользователей, курсов и объёма данных. Её архитектура должна предусматривать возможность внедрения новых модулей и функций.
* **Безопасность данных.** Все данные студентов, преподавателей и курсов должны быть защищены от несанкционированного доступа. Система должна соответствовать нормам защиты персональной информации.
* **Соответствие нормативным требованиям.** Система должна быть гибкой для адаптации к изменениям в законодательстве и образовательных стандартах.

#### **Ограничения**

##### **Технические ограничения:**

Система должна быть разработана с учётом существующей IT-инфраструктуры университета, что минимизирует затраты на обновление оборудования.

##### **Временные ограничения:**

Сроки реализации проекта должны учитывать этапы планирования, разработки, тестирования и внедрения. Завершение всех этапов должно быть выполнено в рамках утверждённого графика.

##### **Ограничения на обучение пользователей:**

Интерфейс системы должен быть интуитивно понятным, что позволит минимизировать временные затраты на обучение пользователей.

Разрабатываемая система должна не только соответствовать требованиям современных образовательных учреждений, но и предусматривать перспективы её дальнейшего развития. Её внедрение станет ещё одним шагом на пути к оптимизации управления курсами, улучшению качества образовательного процесса и повышению конкурентоспособности университета на образовательном рынке.

### **Теоретическое обоснование проекта**

Разработка автоматизированной системы управления элективными курсами опирается на несколько теоретических подходов и моделей, обеспечивающих её эффективность и устойчивость в рамках образовательного процесса.

**1. Теория управления и оптимизация процессов**

Один из ключевых аспектов проекта — это оптимизация процессов управления, что основывается на классической теории управления и теории оптимизации. Управление образовательным процессом может быть рассмотрено как процесс, требующий оптимального распределения ресурсов, таких как преподаватели, студенты и учебные материалы.

В рамках теории управления применяются методы анализа и оптимизации потоков информации, что позволяет снижать затраты времени и ресурсов. Использование алгоритмов оптимизации для распределения студентов по элективным курсам и преподавателей по группам может значительно повысить эффективность работы системы. Для более глубокого понимания можно обратиться к исследованиям, посвящённым применению различных алгоритмов оптимизации для решения задач распределения в образовательной среде.

#### **2. Теория информационных систем и интеграция данных**

Основой информационной части проекта является теория информационных систем (ИС), включающая методы сбора, хранения и обработки данных. В рамках этой теории осуществляется построение унифицированных баз данных, которые обеспечивают централизованное хранение и обработку информации о курсах, преподавателях и студентах. Такой подход позволяет значительно улучшить процесс принятия решений и сделать систему более гибкой и масштабируемой.

Вопросы интеграции данных между различными информационными системами в образовательных учреждениях подробно рассмотрены в исследованиях, которые анализируют методы интеграции и их роль в унификации и консолидации данных. Это позволяет минимизировать разрозненность информации и повысить её актуальность.

#### **3. Теория пользовательского интерфейса и взаимодействия с системой**

Важным аспектом успешной реализации системы является теоретическая база, связанная с проектированием интерфейсов и взаимодействием пользователя с системой. В этом контексте применяются принципы юзабилити, ориентированные на создание интуитивно понятных и удобных интерфейсов.

Теория взаимодействия с пользователем (HCI — Human-Computer Interaction) акцентирует внимание на создании таких интерфейсов, которые минимизируют затраты времени на обучение и использование системы, делая её доступной для различных категорий пользователей. Исследования в области пользовательских интерфейсов для образовательных платформ подчёркивают важность удобства и эффективности взаимодействия, что помогает улучшить восприятие системы преподавателями и студентами.

#### **4. Теория распределённых вычислений и облачные технологии**

Для обеспечения масштабируемости и надёжности система использует подходы распределённых вычислений. Это позволяет обрабатывать большие объёмы данных и обеспечивать доступность информации в любое время, что особенно важно для образовательных учреждений.

### **Практическое обоснование проекта**

Разработка автоматизированной системы управления элективными курсами базируется на современных технологиях и методах, обеспечивающих её функциональность, масштабируемость и надёжность.

#### **1. Веб-технологии**

Система реализуется как клиент-серверное веб-приложение:

* **Frontend**: React TSX с Material UI обеспечивает динамичный интерфейс, удобный для редактирования и отображения данных.
* **Backend:** Node.js с Express.js отвечает за обработку запросов, взаимодействие с базой данных и выполнение бизнес-логики.

#### **2. Работа с базой данных**

Для хранения данных о курсах, авторах, преподавателях и других элементах системы будет использоваться РСУБД **PostgreSQL**. Эта бвзв данных выбрана за её надёжность и высокую производительность при работе с большими объёмами данных.

Для хранения и моделирования информации будут разработаны для курсов, преподавателей, студентов и дополнительные сущности для связи между ними, такие как расписания и группы студентов. Для работы с базой данных будет использован **Sequelize** (для Node.js) — популярная ORM (Object-Relational Mapping), которая позволяет работать с базой данных через JavaScript, абстрагируя SQL-запросы и обеспечивая удобное взаимодействие с реляционными таблицами.

#### **3. Интеграция и API**

#### REST API обеспечивает взаимодействие между клиентом и сервером, а также поддержку стандартизированных форматов данных (JSON, CSV) для импорта и экспорта. Архитектура системы подготовлена для будущей интеграции с LMS и другими образовательными платформами.

#### **4. Масштабируемость и производительность**

#### Проект поддерживает горизонтальную масштабируемость и асинхронную обработку данных, что позволяет обрабатывать растущие объёмы данных без значительной потери производительности и выполнять ресурсоёмкие операции в фоновом режиме, минимизируя нагрузку на основные потоки.

#### **5. Безопасность данных**

Для обеспечения защиты личных данных и системы в целом, будут внедрены следующие меры безопасности:

* **Шифрование**: Использование HTTPS и стандартных криптографических методов для защиты передаваемой информации.
* **Регулярные обновления безопасности**: Установлен процесс обновления библиотек и зависимостей для защиты от уязвимостей, выявленных в сторонних компонентах.

#### **6. Тестирование и качество кода**

Для обеспечения надёжности и качества кода будут применяться методы тестирования:

* **Юнит-тестирование**: Использование Jest или Mocha (для Node.js) для проверки отдельных компонентов системы.
* **Интеграционное тестирование**: Написание тестов для проверки взаимодействия между компонентами системы.

### **Собственные теоретические разработки при реализации проекта**

В процессе разработки автоматизированной системы управления элективными курсами были предложены несколько теоретических решений, направленных на решение специфических задач, связанных с обработкой, хранением и управлением данными. Эти разработки касаются структуризации процессов работы с данными и обеспечения гибкости системы для её дальнейшего развития.

#### **1. Модель обработки данных.**

Основная задача системы — это сбор, обработка, хранение и редактирование данных о курсах, преподавателях и студентах. Для этого была разработана модель, которая включает следующие этапы:

* **Сбор данных**: Стандартизация форм ввода информации преподавателями и менеджерами УИОТ для упрощения последующей обработки.
* **Оценка доступных ресурсов**. Оценка преподавателей, аудитории и других ресурсов для каждого курса, что помогает избежать перегрузки.
* **Хранение данных**: Использование реляционной базы данных PostgreSQL для организации структурированной информации. Таблицы и связи между ними обеспечивают целостность и быстродействие системы.
* **Редактирование данных:** Веб-интерфейс, созданный на React TSX, предоставляет пользователям удобный инструмент для изменения информации без необходимости использования внешних средств.

#### **2. Интеграция с внешними образовательными платформами**

Для повышения гибкости и улучшения взаимодействия между системой и потенциально другими образовательными платформами (например, LMS), была предложена модель **API-ориентированного взаимодействия**.

* **Стандартизированные интерфейсы**. Модели взаимодействия через API были разработаны для упрощения обмена данными между платформами. Включение в систему модуля API позволяет интегрировать информацию вне системы в режиме реального времени.
* **Поддержка различных форматов данных**. Система предусматривает использование стандартов данных, таких как **JSON** и **CSV**, для обмена информацией с другими системами. Это позволяет синхронизировать данные между различными модулями системы, минимизируя избыточность и повышая актуальность информации.

#### **3. Теория адаптивности и расширяемости системы**

Для обеспечения возможности долгосрочного использования системы была разработана модель, поддерживающая её адаптацию к изменениям в образовательной среде.

* **Модульная архитектура**: Каждая функциональная часть системы (например, модуль редактирования данных или модуль генерации отчётов) изолирована, что упрощает добавление новых функций или изменение существующих.
* **Масштабируемость**: Система проектируется с учётом роста объёмов данных и числа пользователей. Это позволяет в будущем расширять ресурсы, такие как серверы и базы данных.
* **Обратная совместимость**. Система будет спроектирована таким образом, чтобы старые данные и процессы могли быть интегрированы в новые версии системы, что гарантирует её долгосрочную актуальность.

**4. Теоретические подходы к безопасности**

Так как система будет использоваться в локальной сети, особых мер безопасности не требуется. Тем не менее, реализованы следующие базовые меры защиты:

* **Шифрование данных**: Использование HTTPS для обеспечения безопасности передаваемой информации внутри сети.
* **Единый уровень доступа**: На данном этапе все пользователи системы имеют одинаковые права, что упрощает реализацию и эксплуатацию.
* **Подготовка к масштабированию безопасности**: В будущем можно добавить разграничение ролей пользователей и систему журналирования действий.

### 

### **Особенности разработки проекта. Уникальные решения.**

Проект автоматизированной системы управления элективными курсами разработан с учётом специфики образовательного процесса, требований пользователей и технических особенностей. В данном разделе представлены основные аспекты архитектуры системы, её взаимосвязей и уникальных решений.

#### **1. Архитектура системы**

Система построена на клиент-серверной архитектуре для разделения обработки данных и интерфейса, обеспечивая гибкость и масштабируемость.

* **Frontend (Клиентская часть)**: Клиентская часть отвечает за взаимодействие с пользователем, отображение данных и отправку запросов на сервер. Реализация на React TSX с использованием MUI обеспечивает удобный и динамичный интерфейс. Пользовательская часть позволяет загружать данные, отображать ошибки и предупреждения, а также выполнять базовую валидацию данных.
* **Backend (Серверная часть)**: Серверная часть, реализованная на Node.js с использованием Express.js, обрабатывает запросы от клиента, выполняет валидацию данных, взаимодействует с базой данных и обеспечивает базовые аналитические функции.
* **База данных**: Хранение осуществляется в PostgreSQL, что обеспечивает надёжность и целостность данных.

Система использует архитектурный подход, при котором каждый компонент (клиент, сервер, база данных) выполняет свою четко определённую функцию, что обеспечивает изоляцию проблем и упрощает поддержку и расширение системы.

#### **2. Взаимосвязь элементов системы**

Структура обеспечивает изоляцию функций и упрощает поддержку. Компоненты системы взаимодействуют через API:

1. **Клиентская часть** (Frontend):
   * Отправляет запросы на сервер через **REST API** для получения и отправки данных.
   * Отображает данные и сообщения, предоставляемые сервером
   * Поддерживает загрузку данных и их передачу на сервер для последующей обработки.
2. **Серверная часть** (Backend):
   * Принимает запросы от клиента и выполняет операции, такие как проверка данных, их сохранение в базе данных и генерация ответов.
   * Реализует логику обработки данных, включая валидацию и форматирование для клиента.
3. **База данных**:
   * Хранит информацию о курсах, преподавателях, группах, аудиториях и их взаимодействиях.
   * Обеспечивает доступ к данным через запросы, реализуемые с помощью ORM Sequelize.
4. **API**:
   * Служит связующим звеном между клиентом и сервером.
   * Обеспечивает стандартизированные методы для взаимодействия.

Чёткая структура взаимодействия между компонентами системы позволяет разделить логику пользовательского интерфейса и бизнес-процессов, что улучшает масштабируемость и упрощает поддержку.

#### **3. Взаимодействие с внешними программными продуктами**

На текущем этапе система не взаимодействует с внешними платформами, такими как LMS, однако её архитектура предусматривает возможность интеграции в будущем.

* **Форматы данных**: Система поддерживает JSON и CSV для импорта и экспорта данных, что обеспечивает совместимость с другими системами.
* **API для интеграции**: В будущем предусмотрена возможность добавления интерфейсов для обмена данными с платформами управления обучением, такими как Moodle.
* **Интеграция с системами отчётности**: Система будет использовать возможность генерации отчётов в форматах, совместимых с другими системами (например, **Excel** или **PDF**), что позволит администраторам и преподавателям получать отчёты, которые можно использовать в других приложениях или платформах.

#### **4. Особенности реализации и поддержка масштабируемости**

Система была спроектирована с учётом роста числа пользователей и объёмов данных:

* **Масштабируемость и гибкость**. Система спроектирована модульно, что позволяет добавлять новые функции без необходимости переработки существующего функционала. Поддерживается горизонтальная масштабируемость, что позволяет расширять ресурсы для работы с увеличенными объёмами данных.
* **Производительность**. Асинхронные операции обеспечивают быструю обработку запросов, минимизируя время отклика. Использование кэширования помогает оптимизировать работу с часто запрашиваемыми данными.

#### **5. Безопасность и защита данных**

Так как система работает в локальной сети, основные меры безопасности включают:

* **Шифрование данных**: Использование HTTPS для защиты данных при их передаче.
* **Единый уровень доступа**: На данном этапе все пользователи имеют равные права, что упрощает эксплуатацию системы.
* **Защита от SQL-инъекций** и других уязвимостей с помощью ORM и валидации входных данных.
* **Подготовка к будущей безопасности**: В будущем возможно добавление разграничения ролей пользователей и системы журналирования действий для повышения прозрачности и безопасности.

Разработка системы управления основывается на принципах модульности, гибкости и простоты интеграции. Чёткая структура компонентов, эффективное взаимодействие через API и база данных, а также поддержка стандартных форматов данных делают систему удобной для эксплуатации и адаптируемой к будущим изменениям в образовательной среде.

6. **Собственные практические решения**

В ходе разработки были найдены и реализованы следующие уникальные решения, повышающие эффективность системы:

* **Упрощение структуры данных:** Стандартизация форматов данных (JSON, CSV) и минимизация сложности схемы базы данных для ускорения разработки и упрощения поддержки.
* **Минимизация инфраструктурных затрат:** Отказ от сложной аутентификации в пользу единого уровня доступа для всех пользователей, что снизило временные затраты на настройку и обучение.
* **Модульная архитектура:** Изоляция бизнес-логики и пользовательского интерфейса, что упростило тестирование и возможность добавления новых функций.
* **Асинхронная обработка данных:** Обеспечение обработки больших объёмов данных в фоновом режиме, минимизируя задержки.
* **Подготовка к интеграции:** Реализация всех функций через REST API, что позволяет легко подключать внешние системы и расширять функционал в будущем.
* Гибкий интерфейс: Создание интуитивно понятного пользовательского интерфейса, оптимизированного для редактирования и валидации данных.

Эти решения обеспечивают надёжность, гибкость и адаптивность системы, что делает её подходящей для современных образовательных учреждений.

### **Функциональные возможности проекта**

Разрабатываемая система управления элективными курсами предлагает набор функций для улучшения управления учебным процессом и повышения эффективности взаимодействия между участниками образовательной среды.

#### **1. Управление данными о курсах**

Система позволяет централизованно управлять информацией о курсах, что является основой для успешной координации учебного процесса. Ключевые функции включают:

* **Добавление, редактирование и удаление курсов**: Менеджеры УИОТ могут добавлять новые курсы, редактировать существующие и удалять их при необходимости. Все изменения автоматически синхронизируются с базой данных.
* **Работа с данными о преподавателях и группах**: Система предоставляет возможность обновления информации о преподавателях.
* **Импорт и экспорт данных**: Поддержка загрузки данных в формате JSON и CSV, что облегчает управление и анализ информации.

#### **2. Работа с преподавателями и студентами**

Система предоставляет функционал, ориентированный на облегчение работы как преподавателей, так и студентов. Основные возможности включают:

* **Управление информацией о преподавателях**: Менеджеры могут обновлять данные о преподавателях и связывать их с конкретными курсами.
* **Управление элективными курсами**: Преподаватели и авторы могут подавать заявки на проведение новых элективных курсов, добавлять к ним описание, информацию о требованиях.
* **Уведомления и напоминания**: Система автоматически информирует преподавателей о важных событиях, например, сроках обновления данных.

#### **3. Проверка и валидация данных**

Система включает механизмы для проверки данных на корректность, что помогает избежать ошибок и несоответствий. Основные функции:

* **Проверка корректности данных**: Система автоматически проверяет введённые данные, например, корректность форматов и наличие обязательных полей.
* **Автоматическая валидация при загрузке**: При импорте данных осуществляется проверка их соответствия установленным правилам.

#### **4. Генерация отчётов**

Один из ключевых элементов системы — это возможность автоматического формирования отчётов, что значительно облегчит работу менеджеров УИОТ. Основные возможности:

* **Создание отчётов по курсам**: Генерация отчётов о текущем состоянии курсов, включая данные об авторстве, ведущих преподавателях и заполняемости.
* **Генерация статистики по студентам**: Система позволяет собирать статистику по успеваемости студентов, их записи на курсы и выполнению заданий.
* **Экспорт данных**: Отчёты могут быть экспортированы в форматы CSV, Excel, PDF для дальнейшего анализа или печати.

#### **5. Интеграция с внешними системами**

Хотя на текущем этапе система работает автономно, её архитектура предусматривает возможности интеграции в будущем. Все функции системы реализованы через REST API, что обеспечивает модульность и упрощает дальнейшее расширение.

#### **6. Адаптивность и масштабируемость**

Система разработана с учётом возможного роста числа пользователей и данных. Архитектура системы позволяет увеличивать ресурсы для работы с большим количеством данных.

#### **7. Пользовательский интерфейс и удобство работы**

Интерфейс системы был спроектирован с учётом принципов юзабилити, что обеспечивает пользователям удобный и быстрый доступ ко всем необходимым функциям:

* **Интуитивно понятный интерфейс**: Для удобства работы с системой все ключевые функции доступны через простой и логичный интерфейс.
* **Поддержка мобильных устройств**: Система может работать с ней через мобильные устройства. (Возможна интеграция с React Native)

Функциональные возможности разрабатываемой системы обеспечивают решение основных задач, стоящих перед образовательными учреждениями, а также позволяют эффективно управлять элективными курсами, минимизируя затраты времени и повышая точность данных. Автоматизация ключевых процессов, включая сбор данных, их валидацию и генерацию отчётов, помогает улучшить организацию учебного процесса и повысить качество образования.

### **Пользовательский интерфейс проекта**

Пользовательский интерфейс является неотъемлемой частью разрабатываемой системы, так как он обеспечивает взаимодействие между пользователями и функционалом системы. Интуитивно понятный интерфейс позволяет сократить время на обучение пользователей и повысить общую эффективность работы с системой. В данном проекте интерфейс был спроектирован с учётом потребностей ключевых пользователей, таких как преподаватели, студенты и менеджеры УИОТ.

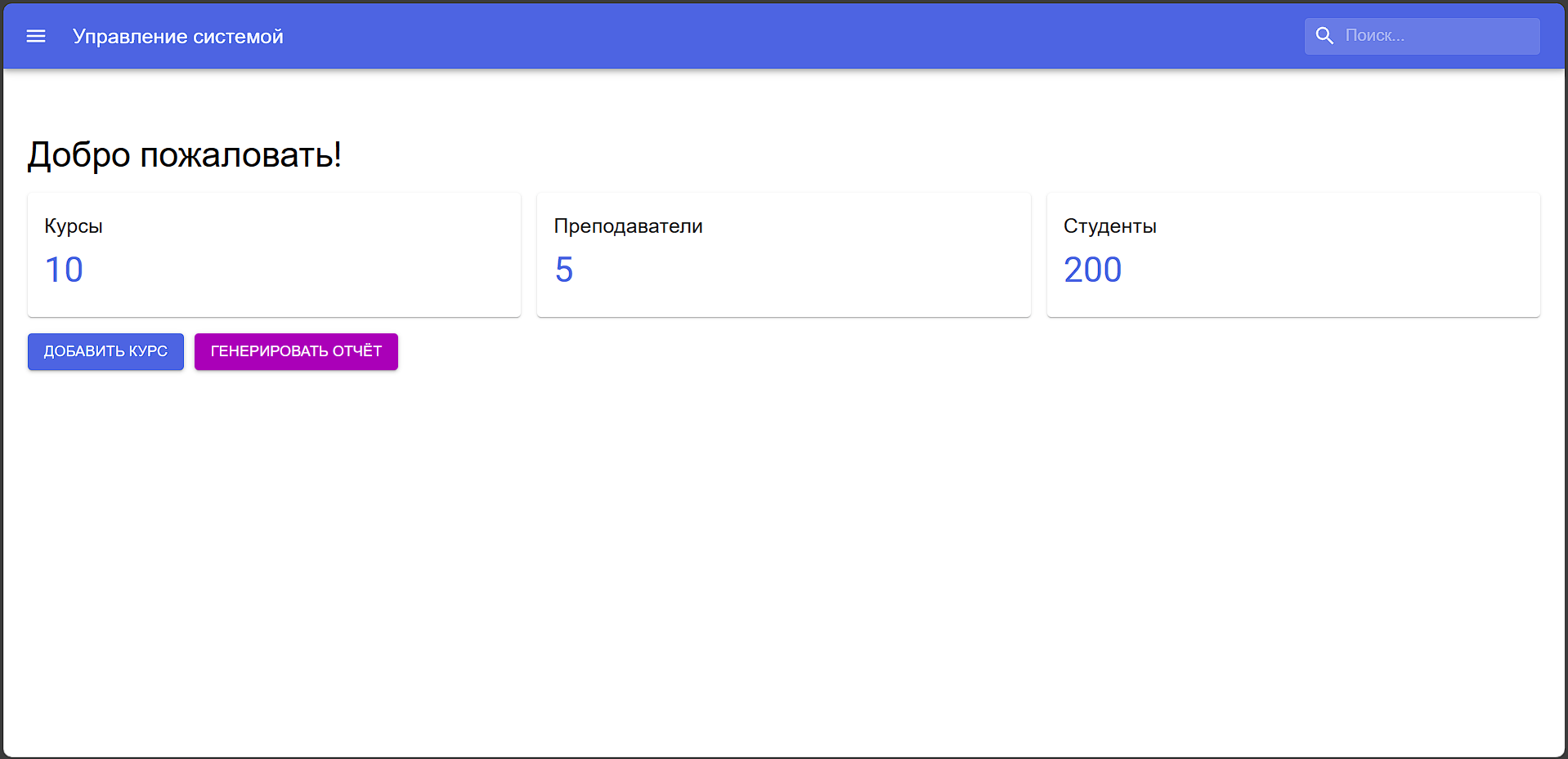
#### **1. Основные требования к интерфейсу**

* **Интуитивно понятный дизайн**: Интерфейс должен быть таким, чтобы пользователи могли легко найти нужные функции без необходимости искать инструкции или проходить длительное обучение.
* **Минимизация шагов для выполнения задач**: Процесс выполнения задач должен быть упрощён до минимума.
* **Представление ошибок и уведомлений**: Понятное отображение ошибок и рекомендаций по их исправлению.

#### **2. Структура интерфейса**

Интерфейс системы разделён на несколько основных разделов, каждый из которых отвечает за свою часть функционала:

* **Главная страница**: На главной странице представлены краткие сведения о текущем статусе элективных курсов, списке доступных курсов и статистике. Менеджеры и преподаватели могут с главной страницы перейти к более детализированным разделам.

  
Рис. 1 — Главная страница

* **Страница курса**: Каждому курсу соответствует отдельная страница, на которой отображается информация о курсе (например, преподаватель, количество групп, расписание). Преподаватели могут редактировать информацию о курсе, а студенты — записываться на курсы.

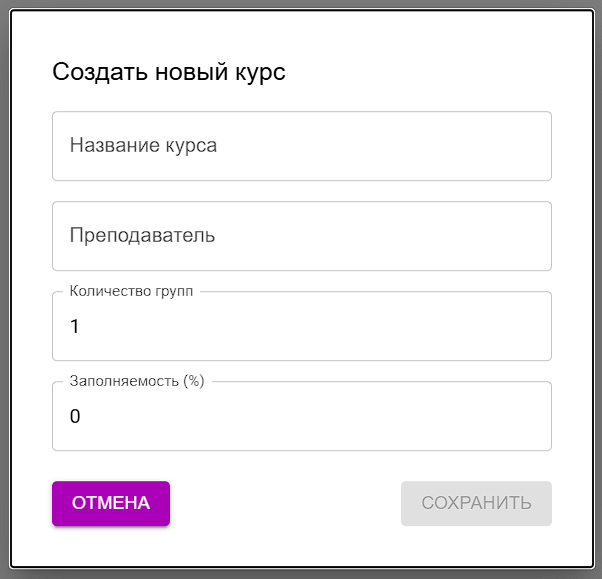
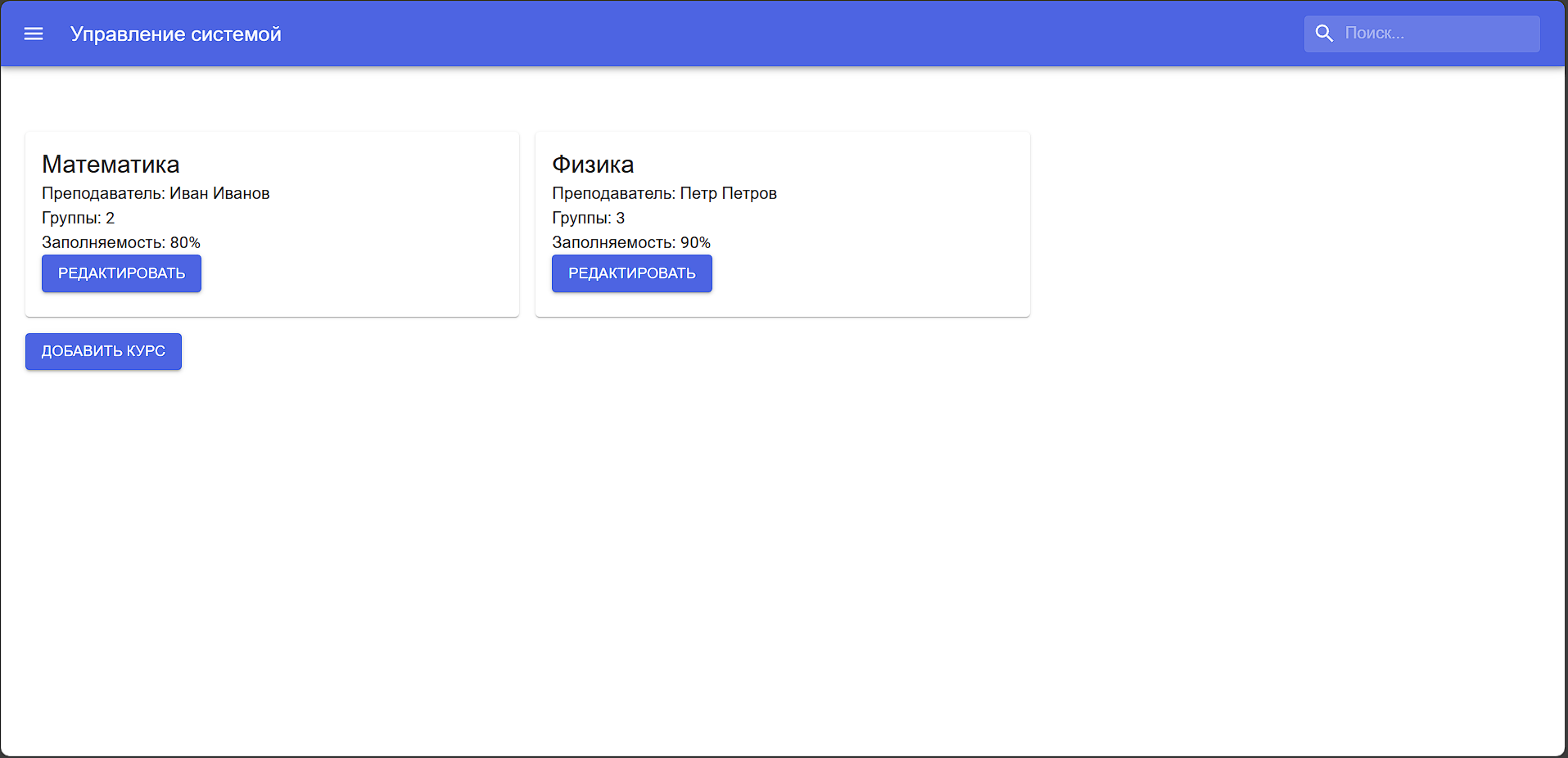


Рис. 2 и 3 — Страница управления курсами и модальная форма для создания

* **Страница преподавателей**: Преподаватели могут просматривать информацию о своих курсах, проверять загруженные материалы и получать уведомления о новых задачах.
* **Страница отчётов**: Для менеджеров УИОТ предусмотрена страница, на которой они могут генерировать отчёты о состоянии курса, студентах и преподавателях, а также экспортировать их в форматах **Excel** или **PDF**.
* **Ошибки и уведомления**: Вся информация о ошибках, таких как некорректно введённые данные, будет отображаться в виде всплывающих сообщений. Эти сообщения можно будет закрыть или исправить, а ошибки будут выделены для упрощения поиска.

#### **3. Визуальные элементы интерфейса**

Интерфейс был спроектирован с акцентом на простоту и удобство, при этом используя минималистичный подход, который включает:

* **Карточки и списки**: Для отображения информации о курсах, преподавателях и студентах используются карточки и списки, которые помогают быстро просматривать и искать необходимую информацию.
* **Модальные окна**: Для редактирования данных, подачи заявок или подтверждения действий используются модальные окна, которые позволяют пользователю не покидать текущую страницу и не перегружать интерфейс.
* **Иконки и кнопки**: Простые иконки для часто используемых действий (например, «Добавить курс», «Редактировать» или «Загрузить файл») делают интерфейс более понятным и удобным.

#### **4. Реализация Frontend**

Для разработки клиентской части использованы современные технологии и подходы, чтобы обеспечить высокую производительность и лёгкость в обслуживании:

* **React** с **TypeScript**: React позволяет создавать динамичные и быстрые интерфейсы, а TypeScript гарантирует типизацию данных, что снижает вероятность ошибок при разработке.

#### **5. Отображение ошибок и уведомлений**

Одной из ключевых функций интерфейса является отображение ошибок и уведомлений, связанных с несоответствиями в данных. Ошибки отображаются в виде списка, чтобы пользователи могли быстро их идентифицировать и исправить:

#### **6. Тестирование интерфейса**

Интерфейс прошёл серию внутренних тестов на удобство и функциональность, включая:

* **Юзабилити-тестирование**: Оценка удобства использования и корректировки интерфейса по отзывам пользователей.
* **Кроссбраузерное тестирование**: Интерфейс протестирован в разных браузерах (Chrome, Firefox, Safari), что гарантирует его стабильную работу на всех популярных платформах.

### **Организация тестирования проекта**

Тестирование — важный этап в процессе разработки любого проекта, особенно когда речь идет о системах, которые должны обрабатывать и анализировать большие объёмы данных, взаимодействовать с пользователями и интегрироваться с внешними платформами. В данном проекте были применены различные виды тестирования, чтобы обеспечить качество, надёжность и эффективность функционирования системы.

#### **1. Виды тестирования, использованные в проекте**

Для подтверждения того, что проект решает все поставленные задачи и эффективно снимает обозначенные проблемы, были проведены следующие виды тестирования:

* **Юнит-тестирование**: Этот вид тестирования был использован для проверки корректности работы отдельных компонентов системы. Каждый модуль или функция проверялись на их способность правильно выполнять операции, такие как проверка данных, обработка запросов и генерация отчётов.
* **Интеграционное тестирование**: Тестирование было направлено на проверку взаимодействия между различными частями системы, например, между фронтенд- и бэкенд-компонентами, а также между сервером и базой данных. Оно позволило убедиться, что данные корректно передаются и обрабатываются между модулями.
* **Функциональное тестирование**: Проверка функциональности всей системы в целом. Это включает проверку основных функций, таких как загрузка данных, регистрация студентов и преподавателей, создание курсов, а также генерирование отчётов. Все эти функции должны работать корректно в реальных условиях использования.

#### **2. Подтверждение того, что проект решает свои задачи**

После проведения тестирования можно уверенно заявить, что проект решает все обозначенные проблемы и задачи, поскольку:

* **Автоматизация сбора и обработки данных**: Процесс загрузки, валидации и генерации отчётов теперь происходит быстро и без ошибок, что значительно сокращает время, затрачиваемое на рутинные операции. Это подтверждается результатами тестов, которые показали отсутствие сбоев при обработке данных.
* **Снижение количества ошибок**: Система успешно справляется с проверкой данных на корректность (например, проверка лимитов по кредитам и обязательных полей), что исключает вероятность ошибок при вводе информации преподавателями или менеджерами.
* **Удобство использования**: Интерфейс был протестирован на реальных пользователях, и все результаты подтвердили, что он интуитивно понятен и удобен для пользователей с разным уровнем подготовки.
* **Обработка больших объёмов данных**: Нагрузочные тесты показали, что система способна работать с большими объёмами данных и не теряет в производительности при увеличении числа пользователей и курсов.
* **Безопасность**: Безопасность системы была проверена с помощью тестов на уязвимости, что позволило выявить и устранить возможные угрозы. Все данные защищены с помощью современных методов шифрования.

#### **3. Внесение изменений и улучшений на основе тестирования**

Тестирование показало несколько областей, где можно было бы улучшить систему:

* **Ускорение загрузки данных**: Было замечено, что при загрузке больших файлов система может немного замедляться. Для решения этой проблемы была оптимизирована логика обработки файлов, а также добавлены асинхронные операции для повышения скорости.
* **Улучшение уведомлений для пользователей**: Хотя система успешно отправляет уведомления, были добавлены дополнительные функции для более точного и персонализированного оповещения пользователей о важнейших событиях (например, статус заявок на новые курсы).
* **Оптимизация пользовательского интерфейса**: На основе юзабилити-тестирования были внесены изменения в интерфейс, чтобы сделать его ещё более удобным и понятным для пользователей, особенно в части отображения ошибок и их исправлений.

**Библиографический список.**

Гаврилова Т.А., Кудрявцев Д.А. **Основы проектирования баз данных**. – Санкт-Петербург: Питер, 2020. – 384 с.

**PostgreSQL Documentation** [Электронный ресурс]. – URL:<https://www.postgresql.org/docs/> (дата обращения: 21.01.2025).

**React Documentation** [Электронный ресурс]. – URL: https://reactjs.org/docs/getting-started.html (дата обращения: 21.01.2025).

**MDN Web Docs** [Электронный ресурс]. – URL:<https://developer.mozilla.org> (дата обращения: 21.01.2025).

**Express.js Documentation** [Электронный ресурс]. – URL:<https://expressjs.com/> (дата обращения: 21.01.2025).

**Jest: JavaScript Testing Framework Documentation** [Электронный ресурс]. – URL: https://jestjs.io/docs/getting-started (дата обращения: 21.01.2025).

**JMeter Documentation** [Электронный ресурс]. – URL:<https://jmeter.apache.org/usermanual/index.html> (дата обращения: 21.01.2025).

**Locust Documentation** [Электронный ресурс]. – URL: https://docs.locust.io/ (дата обращения: 21.01.2025).

**Material UI Documentation** [Электронный ресурс]. – URL: https://mui.com/material-ui/getting-started/overview/ (дата обращения: 21.01.2025).

**TypeScript Handbook** [Электронный ресурс]. – URL: https://www.typescriptlang.org/docs/ (дата обращения: 21.01.2025).

**Bootstrap Documentation** [Электронный ресурс]. – URL: https://getbootstrap.com/docs/ (дата обращения: 21.01.2025).

**TailwindCSS Documentation** [Электронный ресурс]. – URL:<https://tailwindcss.com/docs/installation> (дата обращения: 21.01.2025).