|  |  |
| --- | --- |
| Слайд 1 | Здравствуйте, уважаемые члены Государственной Экзаменационной комиссии!  Меня зовут Макаркина Василина, и я представляю вам свою дипломную работу на тему "Проектирование и разработка веб-приложения для организации учебного процесса в ВУЗе".  В первую очередь, хочу подчеркнуть актуальность выбранной темы. Она заключается в необходимости модернизации устаревших методов организации учебного процесса в университетах. Несмотря на развитие цифровых технологий, во многих вузах до сих пор используются неэффективные инструменты, что снижает удобство и эффективность образовательного процесса. Разработка веб-приложения для автоматизации управления учебной деятельностью позволит упростить взаимодействие между участниками образовательного процесса, повысить его прозрачность и доступность. |
| Слайд 2 | Основной целью моей дипломной работы было спроектировать и создать веб-приложение для организации учебного процесса в университете, которое обеспечит автоматизацию процессов управления расписанием, оценками и взаимодействием участников образовательного процесса.  Для достижения этой цели были поставлены несколько задач:   1. Провести анализ предметной области и существующих аналогичных решений. 2. Определить требования к веб-приложению и обосновать выбор технологий. 3. Спроектировать базу данных для хранения информации о студентах, группах, занятиях и прочих составляющих учебного процесса. 4. Разработать серверную часть приложения на NestJS с созданием API и механизма аутентификации. 5. Создать клиентскую часть на Next.js с удобным интерфейсом для пользователей. |
| Слайд 3-6 | В процессе работы были рассмотрены такие готовые решения, как 1С:Университет, LMS Moodle и LMS HSE. И на основе их анализа составлена сравнительная таблица, которая отображает преимущества и недостатки по выявленным критериям. |
| Слайд 7 | Для разработки серверной части приложения был выбран Node.js — это платформа с открытым исходным кодом, позволяющая запускать JavaScript на сервере. Она обеспечивает высокую производительность и хорошо подходит для обработки большого количества одновременных запросов.  В качестве основного фреймворка выбран Nest.js. Это прогрессивный фреймворк, который использует принципы модульности, инъекцию зависимостей и поддержку TypeScript. Он позволяет создавать масштабируемую и хорошо структурированную серверную архитектуру. |
| Слайд 8 | Для взаимодействия с базой данных я использовала Prisma ORM — это современный инструмент, который позволяет работать с базой данных через TypeScript, исключая необходимость написания SQL-запросов вручную. Это ускоряет разработку и повышает безопасность.  В качестве СУБД выбрана PostgreSQL — мощная объектно-реляционная система управления базами данных, которая обеспечивает надежность, масштабируемость и широкие возможности для работы с данными. |
| Слайд 9 | Фронтенд-приложение разработано с использованием Next.js — это фреймворк на основе React, который позволяет создавать быстрые, SEO-оптимизированные веб-приложения с поддержкой серверного рендеринга и маршрутизации «из коробки».  Также я использовала TypeScript — это надстройка над JavaScript, добавляющая статическую типизацию. Она позволяет писать более надежный код и упрощает поддержку проекта в будущем. |
| Слайд 10 | Для верстки применены Tailwind CSS и Radix UI. Tailwind обеспечивает гибкость и высокую скорость разработки интерфейса, а Radix UI предоставляет готовые, настраиваемые и доступные компоненты, что особенно важно при создании пользовательских интерфейсов.  Для работы с сервером используется Axios — удобная библиотека для отправки HTTP-запросов. А Zustand отвечает за управление состоянием внутри приложения. Вместе они обеспечивают эффективное взаимодействие клиента с сервером и удобное хранение данных на фронтенде. |
| Слайд 11 | Рассмотрим схему базы данных и UML-диаграмму прецедентов, которые играют ключевую роль в проектировании и разработке информационных систем.  Они позволяют заранее определить структуру и поведение системы, что особенно важно при создании комплексных веб-приложений.  На данном слайде представлена UML-диаграмма прецедентов (use-case), которая показывает взаимодействие пользователей с системой и их зоны доступа. |
| Слайд 12 | На этом слайде представлена схема базы данных — это основа хранения и обработки информации в веб-системе. Здесь показаны основные сущности:  Пользователь.  Роль.  Группа, к которой принадлежит студент.  Дисциплина, привязанная к преподавателю.  Расписание, где указаны день недели, номер пары, аудитория, группа, преподаватель и дисциплина.  Занятие, связанное с конкретным расписанием.  Оценка, включающая статус посещения и значение, связанная с занятием и студентом.  Портфолио и достижения, где студент может загружать свои достижения и хранить информацию о внеучебной активности. |
| Слайд 13 | Демонстрация |
| Слайд 14 | Заключение |

В заключение, хочу выразить благодарность всем, кто помогал мне в процессе работы над дипломом: моему научному руководителю [Имя научного руководителя] за поддержку и советы, а также всем преподавателям и коллегам за их помощь и вдохновение.  
Я буду рад ответить на ваши вопросы и обсудить результаты моей дипломной работы. Спасибо за внимание!

ЗАМЕТКИ

1С: Университет

+

Комплексность: Охватывает административное и учебное управление

Интеграция: Полная совместимость с другими продуктами 1С

Гибкость: Поддержка разных моделей образовательного процесса

Отчетность: Возможность формирования разнообразной отчетности для анализа и контроля процессов.

-

Устаревший интерфейс

Сложность освоения

Ограниченные возможности для дистанционного обучения: Система не предназначена для полноценного онлайн-обучения и интерактивного взаимодействия.

LMS Moodle

+

Открытость и расширяемость: Возможностью добавления плагинов для кастомизации под нужды конкретного вуза.

Мобильная поддержка: Наличие мобильного приложения с push-уведомлениями.

Разнообразие учебных материалов

Оценка и аналитика

Гибкие настройки: Возможность настройки доступа, дедлайнов, оценок и попыток в тестах.

-

Ограниченные возможности административного управления: Moodle ориентирован в первую очередь на учебный процесс, что делает его менее подходящим для административных задач.

Слабая интеграция с внешними системами: Встроенные модули не всегда совместимы с корпоративными системами ВУЗов, например, бухгалтерией или кадровыми решениями.

LMS HSE

+

Персонализация: Индивидуальные учебные планы, расписание и электронные зачётные книжки для каждого студента.

Модульность: Включает общеуниверситетские сервисы, такие как «Антиплагиат», «Оцени свои курсы» и дисциплины по выбору.

Удобство: Современный и минималистичный интерфейс.

-

Ограниченная доступность: Создана специально для ВШЭ.

Закрытость системы: Не поддерживает установку внешних плагинов и модулей, что ограничивает её расширяемость.