МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ПЛАТФОРМЫ ДЛЯ ОНЛАЙН-КУРСОВ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 38.03.05 — Бизнес-информатика

механико-математического факультета
Батманова Данилы Алексеевича

Научный руководитель
доцент, к. э. н., доцент _____ А. Р. Файзлиев

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2026

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность темы. В 2025 году сфера онлайн-образования (EdTech) в России переживает глубокую трансформацию под воздействием технологий искусственного интеллекта. Объем рынка оценивается в сотни миллиардов рублей, однако сохраняется ключевая проблема: многие платформы остаются лишь «цифровыми библиотеками», которые хранят контент, но не помогают учащемуся выстроить целостную картину знаний. Студенты часто сталкиваются с фрагментарностью информации и высокой когнитивной нагрузкой.

Разработка серверной части платформы, способной не только хранить данные, но и интеллектуально структурировать знания с помощью LLM (больших языковых моделей), является актуальной задачей. Это позволяет автоматизировать создание глоссариев, ментальных карт и адаптивных тестов, повышая эффективность обучения.

Объект исследования — процессы проектирования и разработки серверной части веб-ориентированных платформ для онлайн-обучения.

Предмет исследования — методы и средства проектирования серверной архитектуры, обеспечивающей интеграцию ИИ-модулей для автоматизированного анализа учебных материалов.

Цель работы — разработать серверную часть образовательной платформы, поддерживающей классические LMS-функции и интеллектуальный контур структурирования знаний.

Задачи исследования:

1. Проанализировать рынок EdTech и выявить требования к современным платформам.
2. Выполнить сравнительный анализ существующих архитектурных решений.
3. Спроектировать микросервисную архитектуру системы с выделенным ИИ-контуром.
4. Реализовать серверную часть (REST API) и сервис анализа контента
5. Провести тестирование и оценку работоспособности прототипа.

1 Анализ предметной области и постановка задачи

Анализ рынка EdTech и существующих решений.

Современный рынок образовательных технологий (EdTech) в 2025 году характеризуется переходом от простого хранения контента к интеллектуальному сопровождению учащегося. Согласно аналитическим данным, объем российского рынка онлайн-образования достиг уровня 699 млрд руб. Основными игроками остаются крупные экосистемы (LMS), однако их функционал зачастую ограничен линейным предоставлением видеоматериалов и текстовых лекций.

В ходе работы был проведен сравнительный анализ популярных платформ (Stepik, Odin, Coursera).

Выявлены следующие проблемы:

1. Фрагментарность знаний: отсутствие наглядных связей между отдельными терминами и темами курса.
2. Высокая когнитивная нагрузка: учащийся вынужден самостоятельно структурировать огромные массивы информации.
3. Зависимость от облачных ИИ: использование внешних API (OpenAI, Anthropic) создает риски утечки данных и зависимости от зарубежных сервисов.

Требования к разрабатываемой системе.

На основании анализа были сформулированы ключевые функциональные требования к серверной части платформы:

1. Управление контентом: создание и хранение иерархической структуры курсов (модули, главы, уроки).
2. Интеллектуальный анализ: автоматическое извлечение ключевых понятий из загружаемых материалов.
3. Визуализация знаний: формирование данных для построения интерактивных ментальных карт и глоссариев.
4. Безопасность: разграничение прав доступа (преподаватель/студент) и защита персональных данных.

Постановка задачи разработки.

Основная задача дипломного проекта — разработка серверной архитектуры, которая объединяет классические функции системы управления обу-

чением (LMS) с автономным модулем интеллектуального анализа. Серверная часть должна обеспечивать:

1. Обработку запросов от клиентской части через REST API.
2. Интеграцию с локально развернутой языковой моделью (LLM) для обработки текстов.
3. Стабильную работу с базой данных для хранения прогресса пользователей и структуры курсов.

Таким образом, разрабатываемая платформа должна представлять собой не просто «цифровую библиотеку», а систему, формирующую смысловую модель знаний, помогающую пользователю в навигации и повторении материала.

2 Проектирование архитектуры и выбор технологий

Архитектурное решение системы.

Для обеспечения гибкости и возможности независимого масштабирования интеллектуальных функций была выбрана трехуровневая архитектура показанная на рисунке 1:

1. Клиентская часть (SPA): отвечает за визуализацию интерфейса и взаимодействие с пользователем.
2. Основной сервер (REST API): центральный узел, управляющий бизнес-логикой, авторизацией и доступом к данным.
3. ИИ-сервис (Glossary Service): изолированный Python-сервис, предназначенный исключительно для тяжелых вычислений и работы с локальной языковой моделью (LLM).

Такой подход позволяет разгрузить основной сервер: пока ИИ-модуль анализирует текст лекции и строит ментальную карту, основная платформа продолжает стабильно обслуживать запросы других пользователей.

Обоснование технологического стека.

Выбор инструментов продиктован требованиями к скорости разработки и производительности:

1. Backend (Основной): Node.js / FastAPI (в зависимости от конкретной реализации модулей). Выбор обусловлен высокой скоростью обработки асинхронных запросов, что критично для образовательной платформы с элементами реального времени.
2. Database: PostgreSQL. Выбрана как надежная реляционная СУБД для хранения сложных структур курсов, связей между модулями и прогресса студенто
3. AI Stack: Python, библиотеки для работы с LLM (например, LangChain или интеграция с Ollama). Python является стандартом для ИИ-разработки, обеспечивая доступ к современным методам NLP (обработки естественного языка).
4. Frontend: React 19 + TypeScript. Использование TypeScript минимизирует количество ошибок при передаче данных от сервера к клиенту за счет строгой типизации.

Модель данных.

В основе серверной части лежит спроектированная схема базы данных, включающая следующие ключевые сущности:

1. Users и Profiles: данные пользователей и их роли (Студент/Преподаватель).
2. Courses и Content: иерархия «Курс — Модуль — Урок», где каждый урок может содержать текст, видео и задания.
3. Knowledge Map: специфическая таблица для хранения извлеченных ИИ понятий (терминов) и их семантических связей, на основе которых строится визуальный глоссарий.

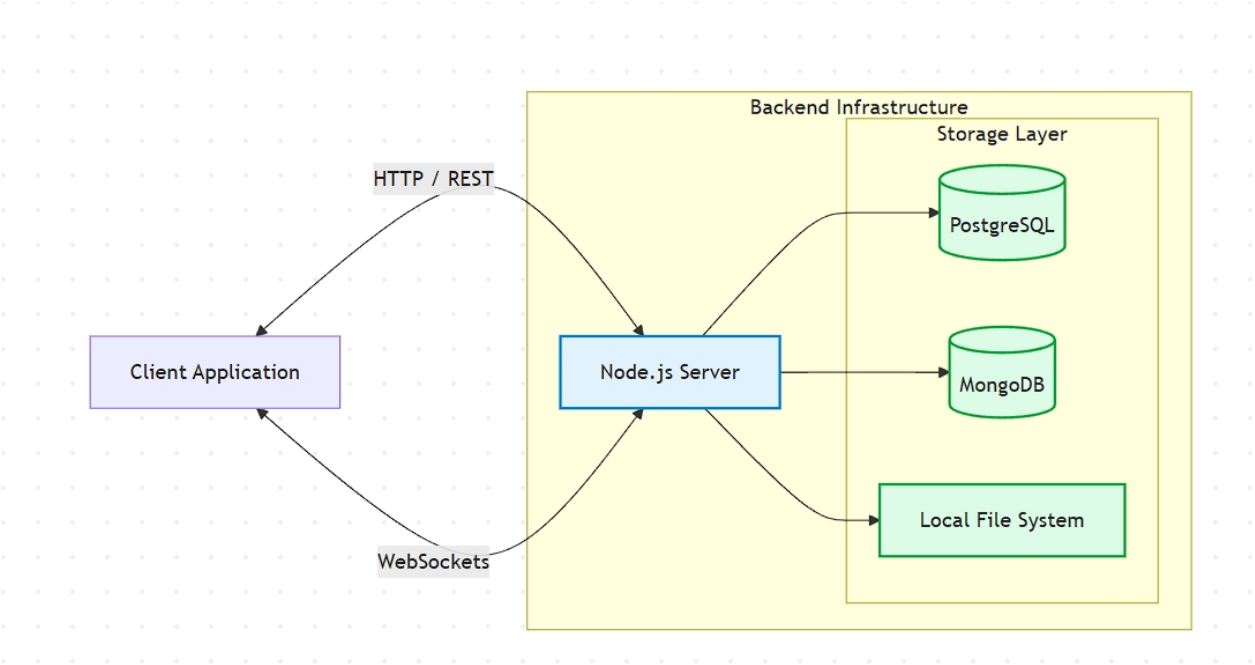

Рисунок 1 – Структура работы проекта.

3 Программная реализация серверной части и ии-модуля

Реализация основного API.

Серверная часть платформы функционирует как центральный узел, обеспечивающий взаимодействие между базой данных и пользовательским интерфейсом. В рамках работы реализованы следующие ключевые контроллеры:

1. Auth Controller: реализует механизм регистрации и авторизации пользователей с использованием защищенных токенов (JWT), что обеспечивает безопасный доступ к личным кабинетам.
2. Course Controller: отвечает за CRUD-операции (создание, чтение, обновление, удаление) учебных материалов. Поддерживает иерархическую структуру «курс — модуль — урок».
3. Progress Controller: отслеживает прохождение уроков студентами и фиксирует время, затраченное на обучение, для последующей аналитики.

Алгоритм работы сервиса анализа контента (Glossary Service).

Особенностью данной работы является интеграция асинхронного сервиса для интеллектуальной обработки текстов. Процесс генерации смыслового слоя состоит из нескольких этапов:

1. Извлечение текста: сервер получает текстовое содержимое урока (или расшифровку видео лекции).
2. Запрос к LLM: текст передается локально развернутой языковой модели с использованием структурированного промпта. Модель выделяет ключевые термины и их определения.
3. Построение графа понятий: сервис анализирует связи между терминами (например, «понятие А является частью понятия Б»).
4. Валидация: полученные данные сохраняются в базу данных со статусом «черновик» для последующей проверки и корректировки преподавателем.

Технологии обмена данными.

Для обеспечения «бесшовного» пользовательского опыта используются следующие решения представленные на рисунке 2:

1. REST API: основной протокол для передачи данных между фронтендом и бэкендом.

2. WebSockets: используются в модуле «Виртуальный класс» для мгновенной передачи текстовых сообщений и обновлений статуса пользователей в реальном времени.
3. CORS & Security: настроены политики кросс-доменных запросов и механизмы защиты от типичных веб-угроз (SQL-инъекции, XSS).

Реализованная архитектура позволяет системе обрабатывать запросы в асинхронном режиме: пока ИИ-модуль выполняет ресурсоемкую операцию анализа, основной поток API остается свободным для других пользователей.

3.1 Проектирование базы данных и обеспечение безопасности

Инфологическая модель данных.

Для хранения информации выбрана реляционная модель, реализованная в СУБД PostgreSQL. Структура БД спроектирована таким образом, чтобы поддерживать целостность данных при сложных связях между контентом и интеллектуальными метаданными. Ключевые таблицы включают:

1. Users & Auth: хранение хешированных паролей (использование Argon2/BCrypt) и сессионных данных.
2. Courses & Modules: иерархическая структура, где каждый элемент имеет порядковый номер для корректного отображения последовательности обучения.
3. Glossary_Items: таблица, содержащая термины, их определения и веса релевантности, присвоенные ИИ-модулем.
4. Semantic_Links: связующая таблица для построения графа знаний (связи «целое-часть», «причина-следствие»).

Обеспечение безопасности и отказоустойчивости.

Разработка серверной части включала реализацию нескольких уровней защиты:

1. Аутентификация и авторизация: применение стандарта JWT (JSON Web Tokens). Токены разделены на access (короткоживущие) и refresh (для обновления сессии), что минимизирует риск кражи доступа.
2. Валидация входящих данных: на уровне API реализована строгая проверка всех входящих JSON-объектов. Это исключает возможность проведения атак типа SQL-инъекций или переполнения буфера.

3. Изоляция ИИ-контура: сервис обработки текстов вынесен в отдельную среду. Это гарантирует, что даже при экстремальной нагрузке на модель (например, при загрузке очень длинного текста) основной сервер API будет доступен для навигации других пользователей.

Оптимизация запросов.

Для ускорения работы платформы применены механизмы индексации по часто используемым полям (ID пользователя, ID курса) и реализована ленивая загрузка (lazy loading) для тяжелых медиа-материалов, что снижает нагрузку на серверную часть при первичном обращении к API.

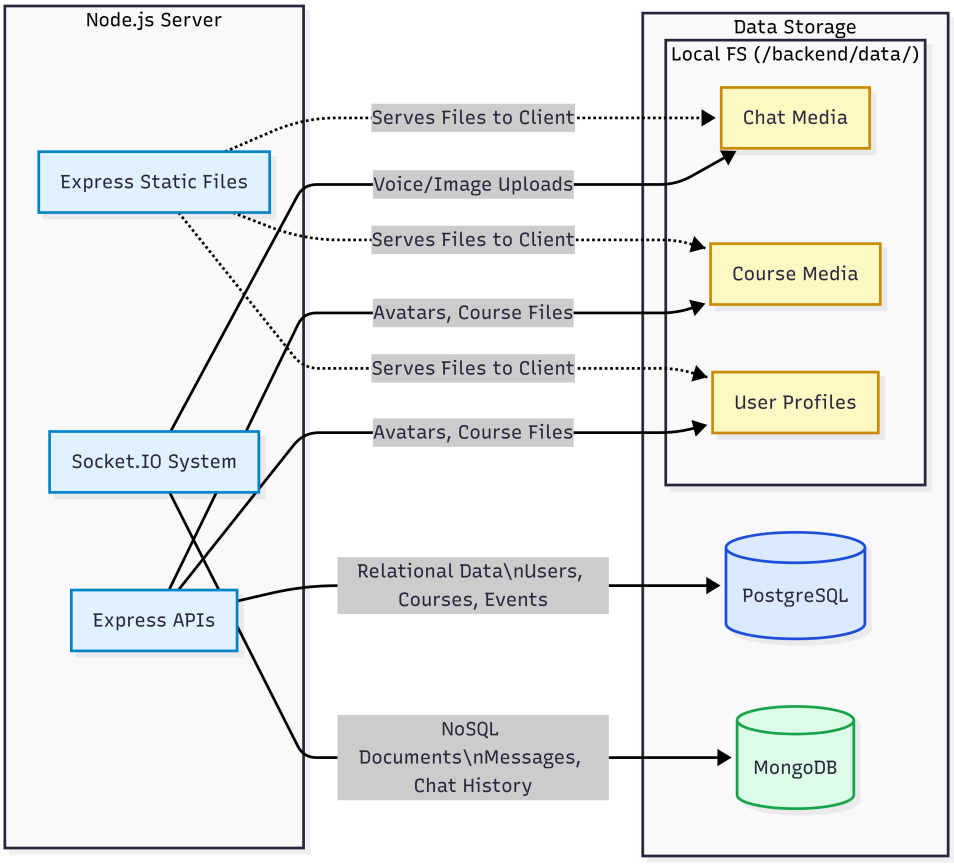

Рисунок 2 – Структура работы с базой данных.

4 Интеграция с локальной языковой моделью (LLM)

Принципы работы интеллектуального слоя. В отличие от стандартных решений, использующих внешние API, данная разработка опирается на локальный запуск моделей (семейства Llama или Mistral). Это решает две задачи:

1. Конфиденциальность: учебные материалы и данные пользователей не покидают контур сервера.
2. Автономность: работа системы не зависит от доступности зарубежных сервисов или ограничений по количеству запросов.

Конвейер обработки текста (Pipeline). Процесс автоматического формирования «смыслового слоя» курса выглядит следующим образом:

1. Препроцессинг: очистка текста от лишних символов и разметки.
2. Промпт-инжиниринг: формирование специализированного системного запроса, который заставляет модель возвращать ответ в строгом формате JSON.
3. Постпроцессинг: проверка полученного JSON на соответствие схеме и сохранение в базу данных с привязкой к конкретному уроку.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была реализована серверная часть образовательной платформы, которая успешно решает задачу перехода от простого хранения контента к созданию смысловых моделей знаний.

Основные результаты работы:

1. Спроектирована и реализована микросервисная архитектура, разделяющая классический функционал LMS (управление курсами, пользователями и прогрессом) и интеллектуальный контур обработки текстов.
2. Разработано серверное API, обеспечивающее высокую скорость обработки запросов и надежную авторизацию на основе JWT-токенов.
3. Реализован автономный ИИ-сервис (Glossary Service), который с помощью локально развернутых языковых моделей (LLM) автоматически извлекает термины и строит семантические связи из учебных материалов. Это позволило автоматизировать создание глоссариев и ментальных карт, снижая нагрузку на преподавателя.
4. Обеспечена безопасность данных за счет локального контура обработки ИИ-запросов, что исключает передачу конфиденциальной информации во внешние облачные сервисы.