

**КОЖЕВНИКОВА П. В., ЯСЕНОВЕЦ А. В.**  
**LLM-ПЛАТФОРМА ГЕНЕРАЦИИ ОБРАЗОВАТЕЛЬНЫХ КУРСОВ**  
*УДК 004.8.032.26:37, ГРНТИ 28.23.39*

LLM-платформа генерации  
образовательных курсов

LLM platform for generating  
educational courses

**П. В. Кожевникова<sup>1</sup>,**  
**А. В. Ясеновец<sup>2</sup>**

**P. V. Kozhevnikova<sup>1</sup>,**  
**A. V. Yasenovets<sup>2</sup>**

<sup>1</sup>Ухтинский государственный  
технический университет, г. Ухта;  
<sup>2</sup>Чунцинский Университет почты и  
телекоммуникаций; г. Чунцин, Китай

<sup>1</sup>Ukhta State Technical University,  
Ukhta;  
<sup>2</sup>Chongqing University of Posts and  
Telecommunications; Chongqing,  
China;

*В данной статье рассматривается использование больших языковых моделей (LLM) для улучшения онлайн-образования. Представляется система автоматизации онлайн-обучения, основанная на принципах программированного обучения и LLM, которая позволяет генерировать курсы, проверять актуальность информации, создавать тесты и адаптировать учебные программы к уровню знаний студентов. Описаны ключевые проблемы онлайн-образования и предлагаемые решения, а также рассмотрены существующие аналоги платформ.*

*This article examines the use of large language models (LLMs) to enhance online education. A system for automating online learning based on programmed learning principles and LLMs is introduced. The system enables course generation, information relevance checks, test creation, and course adaptation to students' knowledge levels. Key issues in online education are discussed along with proposed solutions, and existing platform analogs are reviewed.*

**Ключевые слова:** онлайн-платформа, программированное обучение, LLM, образование

**Keywords:** online platform, programmed learning, LLM, education

## **Введение**

Онлайн-образование позволяет достичь более широкого круга студентов, включая тех, кто из-за географических, экономических или других причин не может посещать традиционные учебные заведения. Это обеспечивает доступ к образованию для всех, независимо от их местоположения или финансовых возможностей.

В процессе поиска и усвоения информации человек сталкивается с множеством доступных ресурсов и существующих материалов на различных

платформах. В большинстве случаев найденные им материалы нерелевантные или информация в них не является актуальной.

Человек прибегает к самостоятельному поиску информации и ее структуризации, что выливается в большие временные затраты для него и уменьшает возможности усвоения информации.

Таким образом, самостоятельное онлайн-образование имеет следующие проблемы:

- Большое количество неструктурированной информации;
- Необходимость в дополнительной проверке найденной информации и ее актуальности;
- Необходимость в самостоятельной проверке своих знаний;
- Низкая адаптация учебных программ к уровню знаний студента;
- Отсутствует возможность отслеживать свои результаты;
- Отсутствует возможность поделиться курсом с другими.

Обозначенные проблемы свидетельствуют об актуальности темы.

За последние несколько лет мы стали свидетелями необычайной способности больших языковых моделей (далее – LLM), генерировать ответы, которые не только креативны и разнообразны, но и легко адаптируются к различным потребностям пользователей. Например, исследователи могут предложить ChatGPT сжимать длинные статьи в краткие аннотации для быстрого усвоения; в то время как разработчики видеоигр могут создавать подробные профили персонажей с богатыми личностными качествами, предысторией и уникальными способностями по запросу, просто динамически запрашивая LLM с учетом игрового контекста и предпочтений игроков. Несмотря на то, что конечные пользователи ценят неограниченную креативность LLM, недавние полевые исследования, посвященные разработке приложений на базе LLM, неоднократно демонстрировали необходимость налагать ограничения на результаты LLM. Например, пользователю может потребоваться, чтобы краткое изложение статьи было «строго менее 20 слов», чтобы соответствовать ограничениям по длине, или чтобы сгенерированный профиль персонажа видеоигры был «действительным JSON, который может быть проанализирован Python» для конвейера разработки [1].

Целью системы является автоматизация процесса онлайн-обучения с помощью LLM, которая позволит генерировать структуру и наполнять содержание образовательных курсов под потребности обучающегося, а также давать ему обратную связь.

Данная система основывается на принципах программированного обучения. Программированное обучение можно определить как обучение, которое осуществляется с помощью запрограммированного текста, или обучающей машины (в данном случае мы используем модели машинного обучения), при котором учащемуся в логической последовательности предъявляется множество небольших учебных фрагментов информации. Главное требование - разбить учебный материал на небольшие шаги и обеспечить немедленную обратную связь и подтверждение каждого шага, включающего представление информации,

вопросы и ответы ученика. Основные принципы программированного обучения показаны на Рисунке 1.

Данный подход значительно усовершенствует опыт обучающихся и решит следующие проблемы:

— Большого количества неструктурированной информации за счет подбора актуальной из сети Интернет и генерации структуры на основе LLM;

— Необходимости в дополнительной проверке найденной информации и ее актуальности за счет модерации найденных материалов на основе LLM;

— Необходимости в самостоятельной проверке своих знаний за счет авто-составления контрольных тестов;

— Низкой адаптации учебных программ к уровню знаний студента за счет сопровождения в обучении на основе LLM;

— Невозможности отслеживать свои результаты за счет формирования статистики при прохождении контрольных тестов;

— Невозможности поделиться курсом с другими за счет формирования ленты курсов и дополнительных настроек.

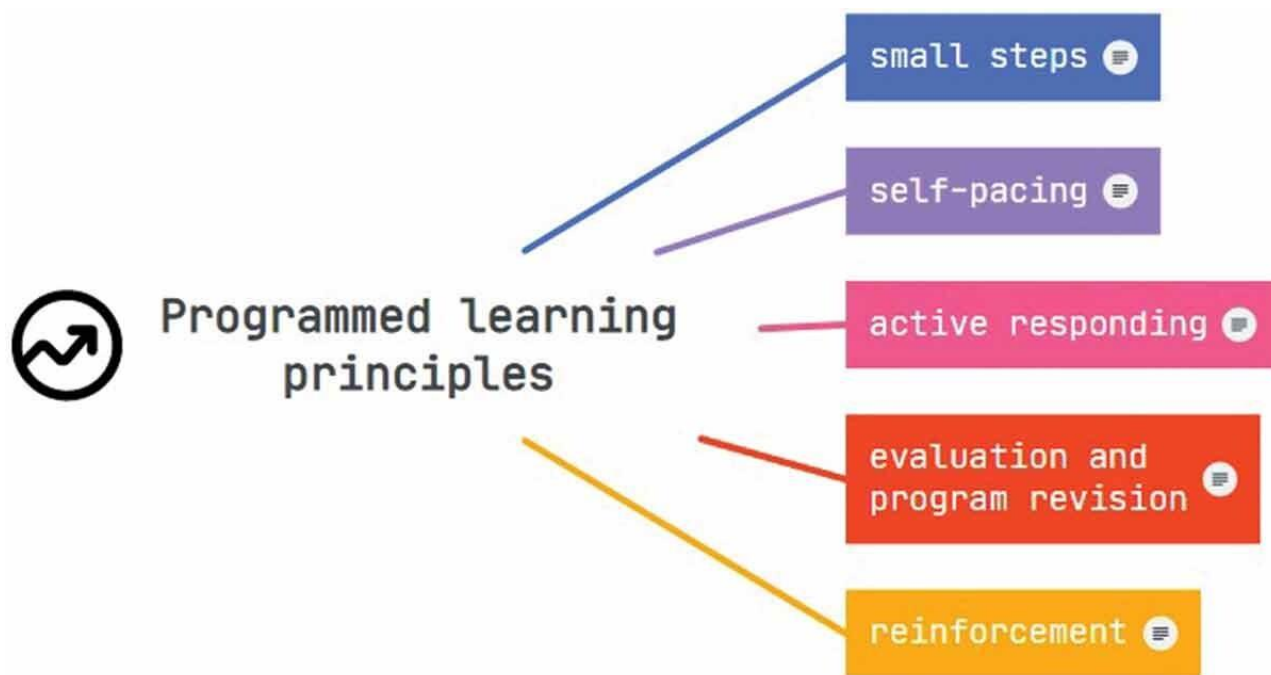


Рисунок 1. Основные принципы программированного обучения [2]

Все это будет способствовать повышению эффективности онлайн-обучения, утилизируя опыт платформ-предшественников, интеграции технологии больших языковых моделей и применения принципов программированного обучения.

В процессе изучения большинства подобных систем было выделено несколько ключевых разделов:

- информационный раздел (каталог курсов);
- система дистанционного обучения;
- профиль пользователя;
- конструктор курсов.

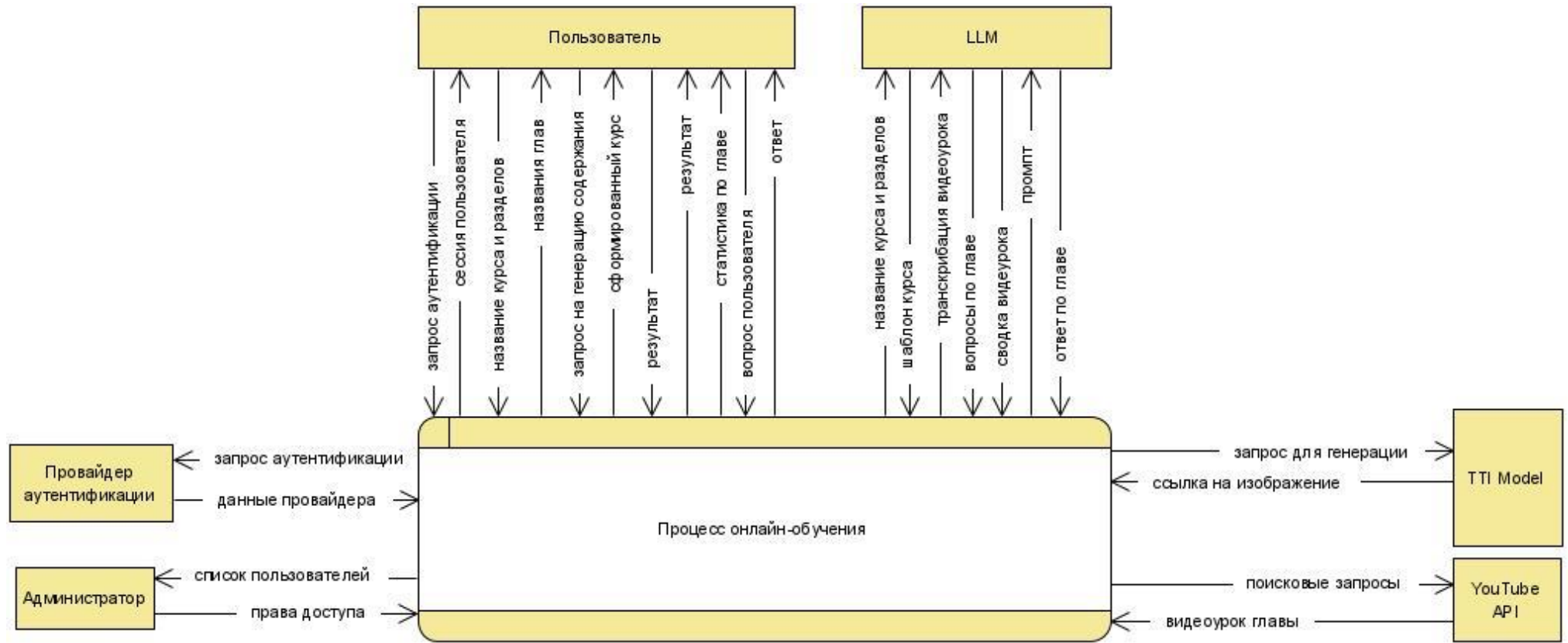


Рисунок 2. Модель «Как будет»

В данном процессе взаимодействуют шесть внешних сущностей:

— Пользователь использует платформу для самостоятельного образования, проходя процесс регистрации и/или аутентификации, генерируя для себя курсы и просматривая видеоуроки, решает тесты.

— Администратор имеет возможность использовать платформу как пользователь, а также ему доступны дополнительные действия модерации над пользователями и курсами.

— Провайдер аутентификации позволяет пользователям зарегистрироваться и/или аутентифицироваться в системе, предоставляя данные из стороннего сервиса.

— Large Language Model слушает запросы приложения удаленно, обрабатывая запросы пользователей во время всего процесса формирования курса, позволяет подготовить структуру курса, а также запросы к другим сущностям и тесты, для наполнения курса учебными материалами.

— Text-To-Image Model (далее – ТТИ) позволяет генерировать превью-изображение курса на шаге подготовки структуры курса, основываясь на сформированном запросе GPT-модели и первичной информации о курсе, указанной пользователем.

— YouTube API позволяет найти релевантные видеоуроки на основе поисковых запросов, сформированных LLM, а также получить их транскрипцию для дальнейшего формирования краткой сводки видеоурока и теста на основе сводки.

Следующим шагом после построения контекстной диаграммы является процесс декомпозиции основного процесса – LLM-платформы «АкадемИИЯ» и создание модели потоков данных (см. Рисунок 3).

Описание модели потоков данных необходимо для предварительного анализа используемых данных в процессах, для дальнейшего проектирования моделей базы данных.

На основании этого были выделены основные требования, согласно которым система должна предоставлять возможность:

- Регистрации пользователей;
- Редактирования аккаунта;
- Формирование курса;
- Настройки курса;
- Решения тестов;
- Формирования ленты курсов;
- Модерации ИС;
- Формирования отчетов по успеваемости;
- Обращения к ИИ-ассистенту в процессе обучения.

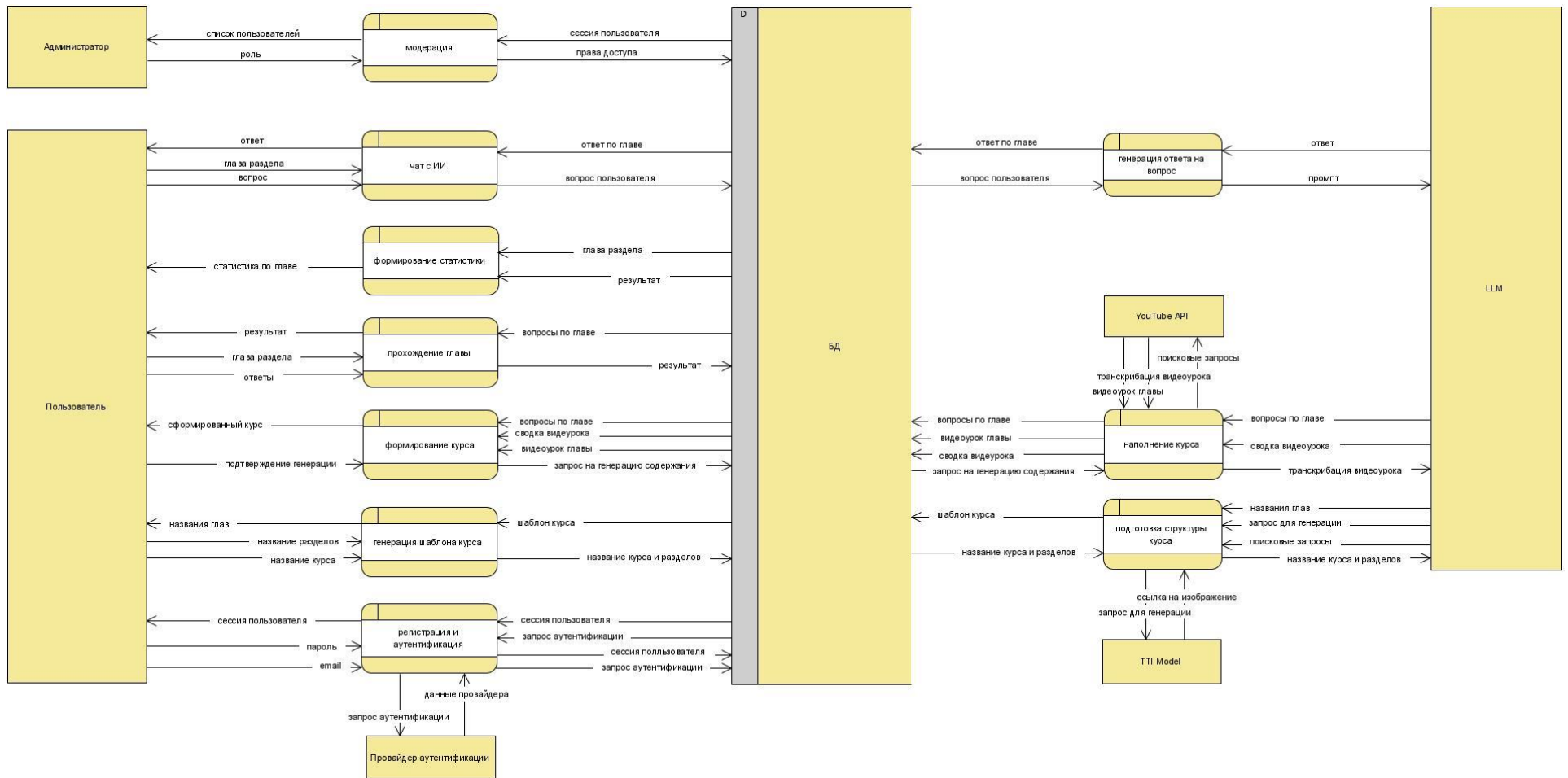


Рисунок 3. Декомпозиция процесса «LLM-платформы «Академия»»

## Обзор аналогов

При поиске программ для возможного решения задачи LLM-платформы были рассмотрены такие системы как:

- «Coursable» – платформа генерации персонализированных курсов;
- «YouLearn» – индивидуальный репетитор по видео с Youtube;
- «Skillbox» – популярная в СНГ онлайн-платформа для обучения.

Сравнение аналогов проводилось по функциональным требованиям, за исключением модерации, из-за невозможности проверить это требование.

В качестве главного аналога был взят «Coursable», далее рассмотрим его более подробно, остальные аналоги являются косвенными и были описаны вкратце.

«Coursable» – иностранная платформа для генерации персонализированных курсов с утвержденными учебными материалами из сети Интернет.

«Coursable» предоставляет возможность создать свой курс на сайте указав в запросе название курса (см. Рисунок 4).

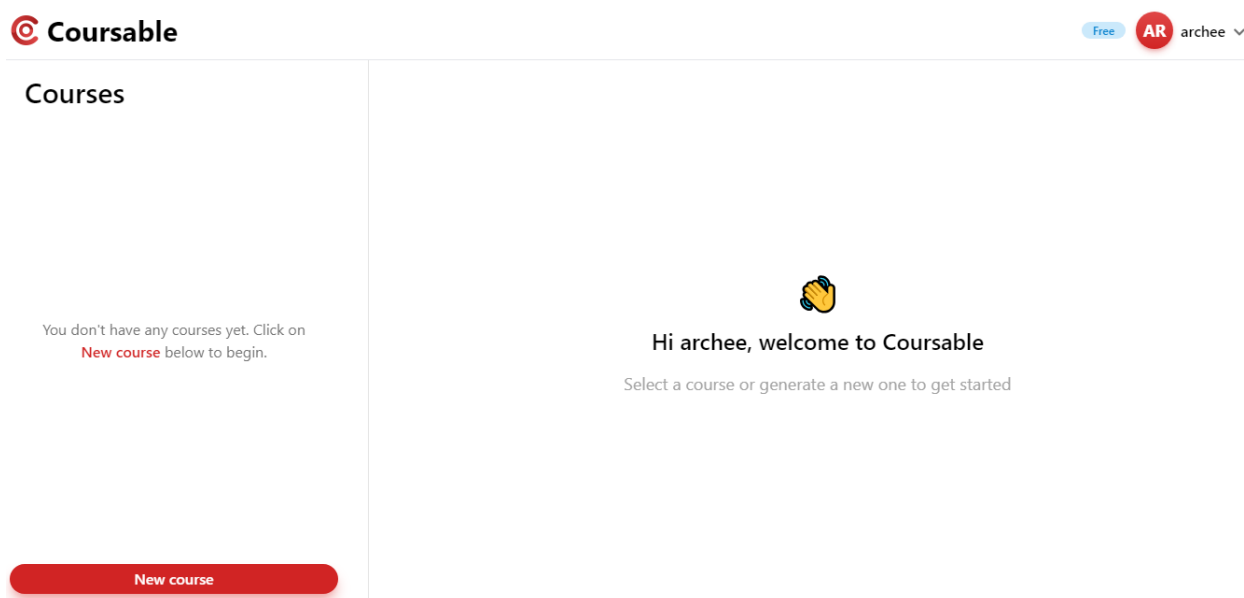


Рисунок 4. Меню личного кабинета ИС «Coursable»

Создадим курс по теме «Информационная безопасность», из результата видно, что платформа использует поиск в Интернете и, следовательно, генерирует в процессе поисковые запросы для поиска материалов. Также можно увидеть окно персонализации учебной программы, что соответствует требованию генерации адаптивных учебных программ (см. Рисунок 5).

Нажмем на кнопку “Подтвердить” и дождемся завершения генерации курса, она занимает приблизительно одну минуту, что вполне приемлемо.

После завершения генерации курса он будет доступен в меню личного кабинета, однако курсы других пользователей недоступны (см. Рисунок 6).

Пользователю будут доступны все разделы нового курса с прикрепленными видео и веб-материалами, а также финальный тест по всему курсу,

что соответствует требованию формы контроля усвоения материалов (см. Рисунок 7).

**Comprehensive Cybersecurity Fundamentals** Cancel Confirm

Free 0/5 personalizations for this course

Here is your full generated course. Ask AI to make changes to personalize it exactly for your needs. Once you are done, click **Confirm** in the top right corner and get to it!

**Changes**

If you have any desired changes for the course outline, type them out here.

**Complexity**

More basic Standard More detailed

**Personalize**

**1. Understanding Cybersecurity**  
Learn the basics of cybersecurity, including its importance, key concepts, and common threats.  
[Materials](#)

**2. Network Security**  
Explore the principles and best practices of securing computer networks, including firewalls, VPNs, and intrusion detection systems.  
[Materials](#)

**3. Data Encryption**  
Study the methods and techniques used to protect data through encryption, including symmetric and asymmetric encryption algorithms.  
[Materials](#)

**4. Risk Management and Compliance**  
Discover how organizations assess and mitigate cybersecurity risks, ensure compliance with regulations, and create security policies.  
[Materials](#)

Рисунок 5. Процесс создания

## Courses

### Advanced Cybersecurity and Binary Exploitation Fundamentals

Completion: 0%

Рисунок 6. Курс создан

Delve into the fundamental concepts of binary exploitation, including buffer overflows, format string vulnerabilities, and memory corruption.

**10. Shell Code Execution**  
Learn about the techniques and considerations involved in executing shell code, including payload creation and bypassing security mechanisms.

**Final Quiz**  
10 questions

secure systems and applications against unauthorized access and exploitation.

**Materials**

- [Part 1: Shellcode Execution with Python | Joff Thyer](#) Video 25 minutes
- [Shellcode Execution \(ret2shellcode\) - pwn104 - PWN101 | TryHackMe](#) Video 21 minutes
- [Local Shellcode Execution without Windows APIs](#) Web

**Completed topic**

Рисунок 7. Процесс обучения



Рассмотрев платформу «Coursable», можно сказать, что на ней:

- Есть регистрация пользователей;
- Есть редактирование аккаунта;
- Есть формирование курса;
- Нет настройки курса на бесплатном плане;
- Есть решение тестов;
- Нет формирования ленты курсов;
- Есть формирование отчетов по успеваемости;
- Нет обращения к ИИ-ассистенту в процессе обучения.

Далее был рассмотрен косвенный аналог «YouLearn», представляющий из себя веб-сайт, который подгружает видео из YouTube (см. Рисунок 8) и предоставляет услуги ИИ-репетитора, который объяснит концепции из видеоролика (см. Рисунок 9).

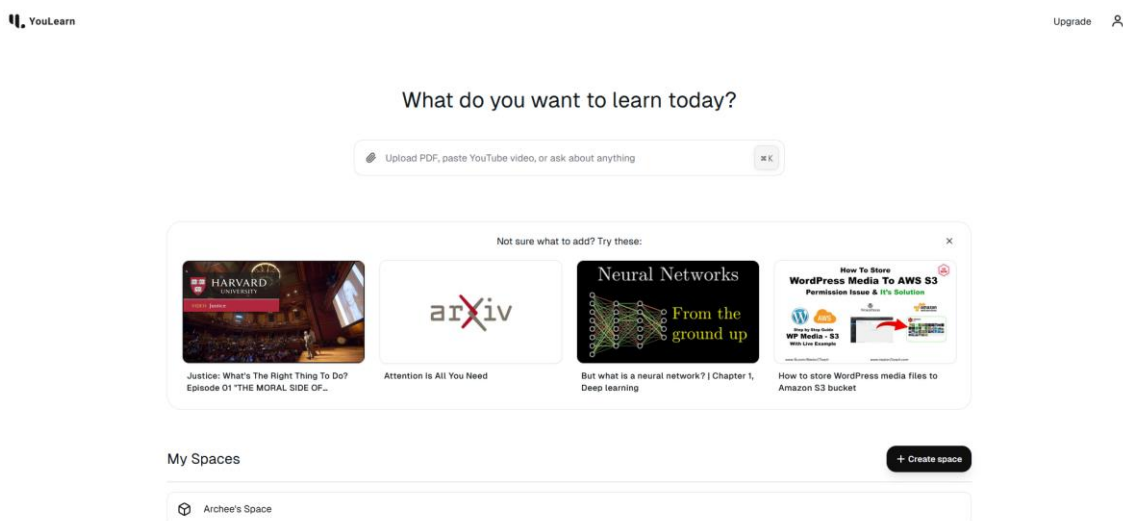


Рисунок 8. Главная YouLearn

Ниже представлена страница видеоролика, справа расположен чат и сводка, а слева видео и ключевые моменты видео (см. Рисунок 9).

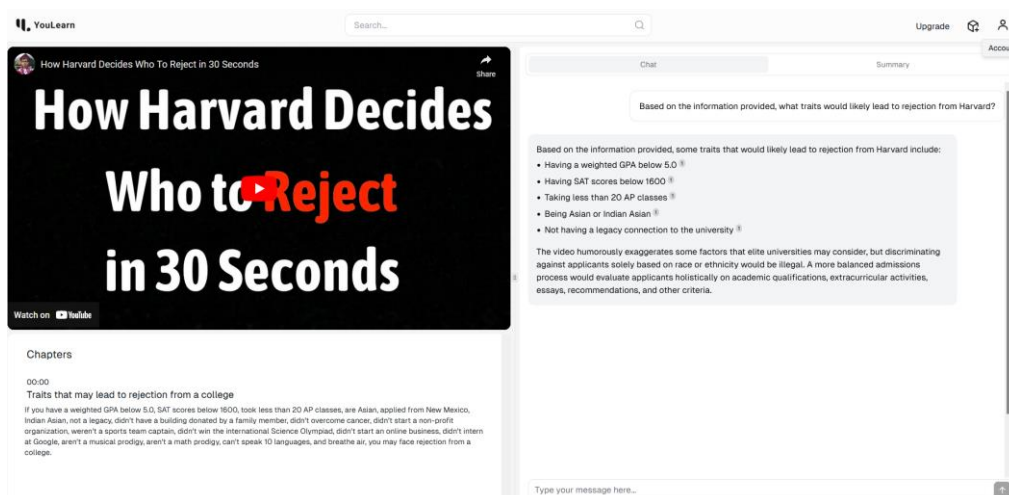


Рисунок 9. Диалоговое окно с ИИ-репетитором

Рассмотрев платформу «YouLearn», можно сказать, что на ней:

- Есть регистрации пользователей;
- Есть редактирование аккаунта;
- Нет формирования курса;
- Нет настройки курса;
- Нет решения тестов;
- Есть формирование ленты курсов;
- Нет формирования отчетов по успеваемости;
- Есть обращения к ИИ-ассистенту в процессе обучения.

Далее был рассмотрен косвенный аналог «Skillbox», представляющий из себя популярную в СНГ платформу для обучения разными специальностям (см. Рисунок 10).

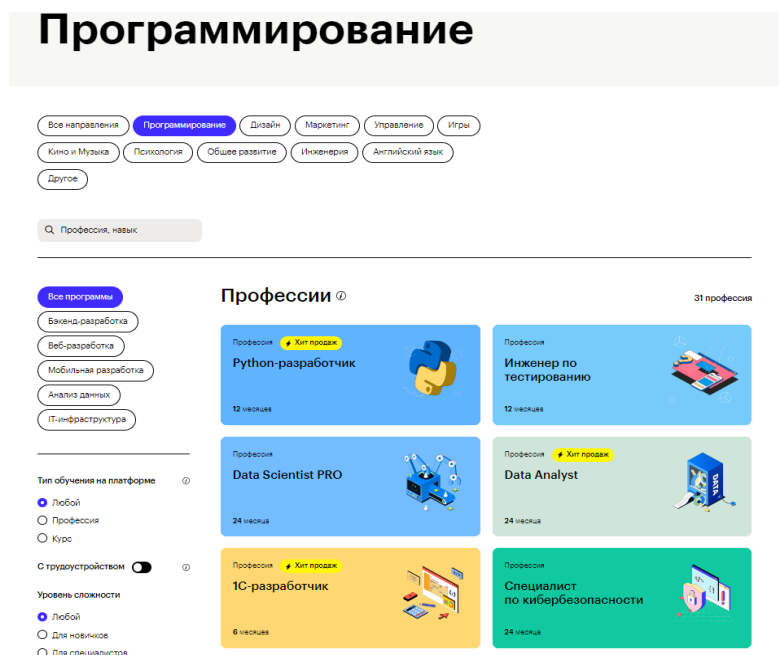


Рисунок 10. Страница с курсами

Рассмотрев платформу «Skillbox», можно сказать, что на ней:

- Есть регистрация пользователей;
- Есть редактирование аккаунта;
- Нет формирования курса;
- Нет настройки курса;
- Есть решения тестов;
- Есть формирование ленты курсов;
- Есть формирование отчетов по успеваемости;
- Нет обращения к ИИ-ассистенту в процессе обучения.

В сравнении с существующими аналогами LLM-платформа «АкадемИИя» обладает рядом преимуществ, которые приведены в Таблице 1.

Таблица 1. Критерии и результаты сравнения

Критерии	Coursable	Youlearn	Skillbox	LLM-платформа «АкадемИИЯ»
Регистрация пользователей	+	+	+	+
Редактирование аккаунта	+	+	+	+
Формирование курса	+	-	-	+
Настройки курса	-	-	-	+
Решение тестов	+	-	+	+
Формирование ленты курсов	-	+	+	+
Формирование отчетов по успеваемости	+	-	+	+
Обращение к ИИ-ассистенту в процессе обучения	-	+	-	+

### Технологический раздел

Для LLM-платформы «АкадемИИЯ» была выбрана гибридная архитектурная модель программного комплекса. В данном случае компоненты ИС развернуты в контейнерах Docker на одном физическом сервере, где контейнер ИС «АкадемИИЯ» является монолитным ядром, которое содержит основную бизнес-логику ИС, микросервисы “scraper” и “validations” предоставляют дополнительные функции и работают с внешними API, а MySQL контейнер содержит базу данных. Архитектура разрабатываемой системы представлена на диаграмме развертывания ниже.

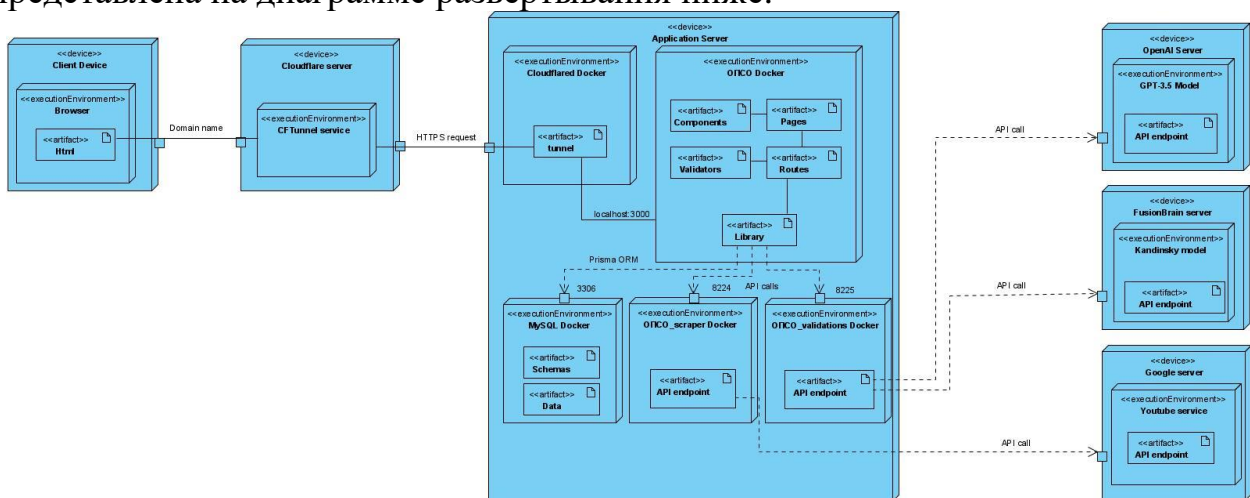


Рисунок 11. Диаграмма развертывания

Для разработки моделей диаграммы потоков данных и проектирования БД было выбрано CASE средство Visual Paradigm. В качестве среды была выбрана интегрированная среда разработки Visual Studio Code. На роль СУБД для LLM-платформы «АкадемИИЯ» была выбрана свободная объектно-реляционная система управления базами данных MySQL.

Для разработки LLM-платформы «АкадемИИЯ» использовался стек технологий, включающий Next.js, TypeScript, Prisma и Auth.js (ранее NextAuth.js). Он предназначен для быстрой и безопасной разработки полноценных веб-приложений.

Для реализации микросервисов «Поиска релевантных материалов» и «Генерации учебных программ», способствующих функционированию основного веб-приложения, был выбран язык программирования Python.

## Проектирование и реализация

Следующим шагом является создание логической (см. Рисунок 12) и физической моделей базы данных (см. Рисунок 13). На основе спецификаций сущностей, сформулированных ранее, были созданы таблицы, в которых находятся описания всех атрибутов сущностей и ограничения, накладываемые на эти атрибуты.

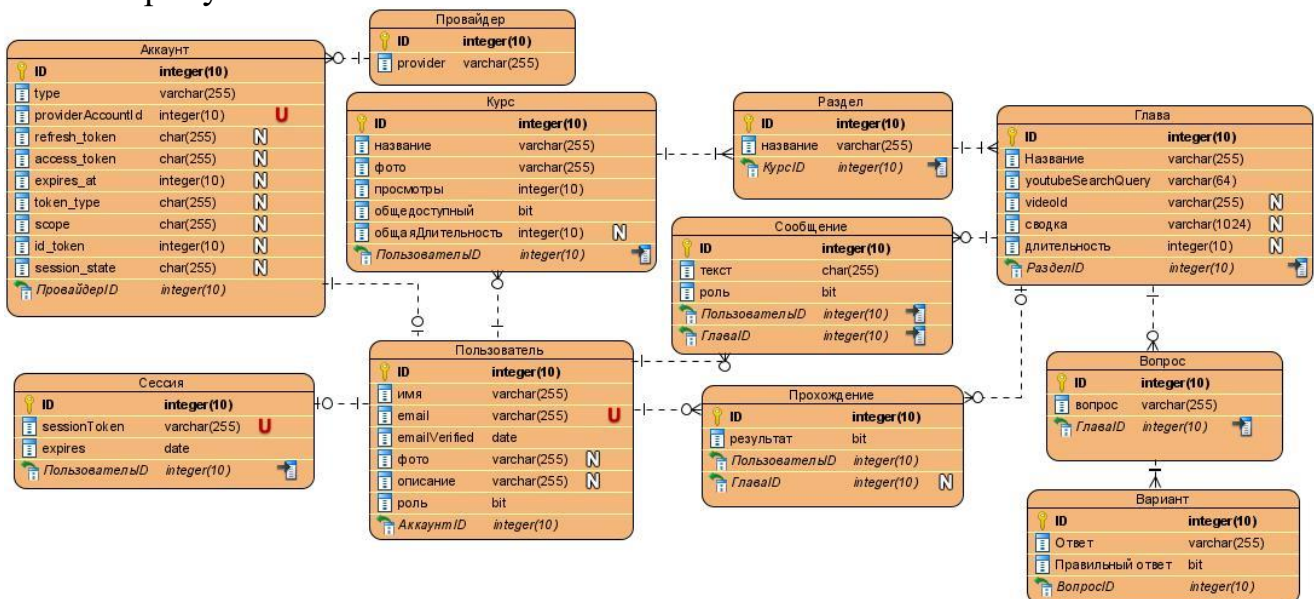


Рисунок 12. Логическая модель базы данных

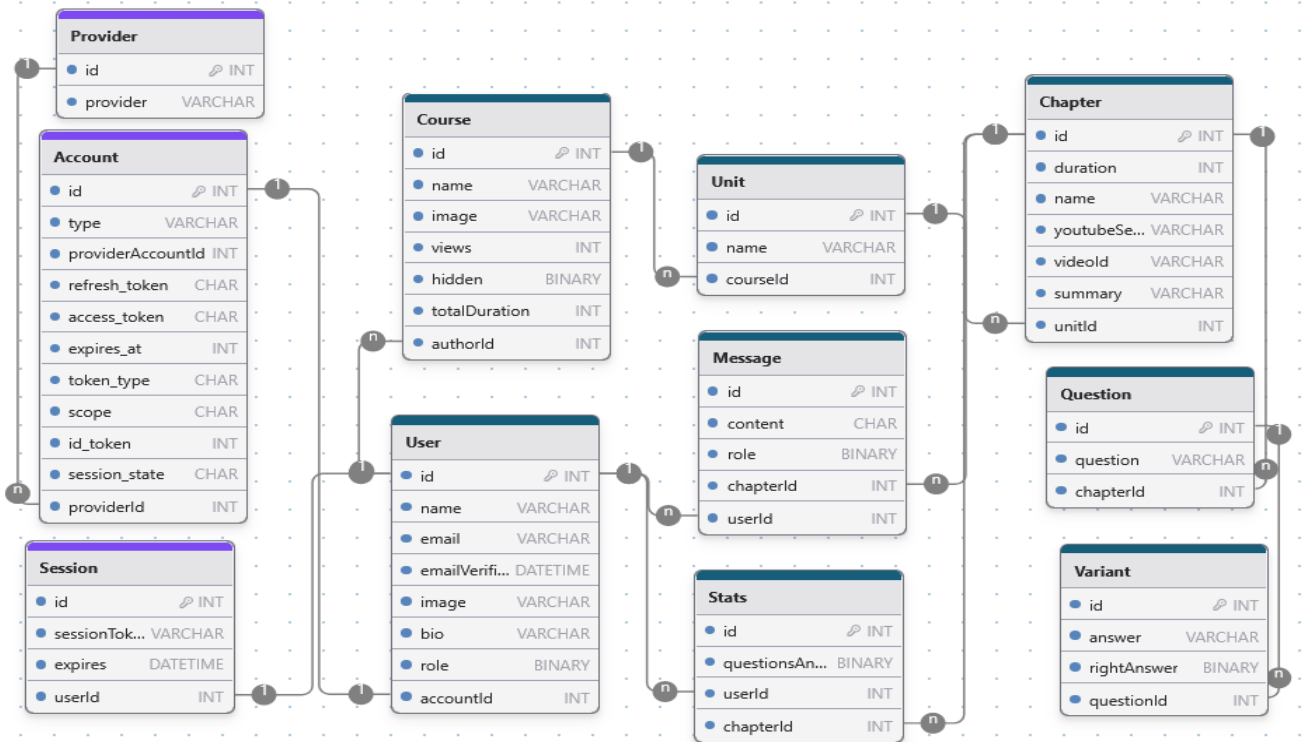


Рисунок 13. Физическая модель базы данных

В созданной системе непосредственно используются 11 таблиц, первые 3 из них являются стандартными таблицами библиотеки Auth.js [9]:

- Таблица Provider – справочник доступных провайдеров аутентификации.
- Таблица Account – создана для хранения данных о провайдере аутентификации пользователя;
- Таблица Session – создана для хранения данных о текущей сессии пользователя;
- Таблица User – создана для хранения данных пользователей;
- Таблица Course – создана для хранения созданных курсов;
- Таблица Unit – создана для хранения разделов курса;
- Таблица Chapter – создана для хранения глав курса;
- Таблица Question – создана для хранения вопросов по сводке видео;
- Таблица Variant – создана для хранения ответов на вопросы по сводке;
- Таблица Message – создана для хранения сообщений пользователей;
- Таблица Stats – создана для хранения результатов пользователей по прохождению глав.

Ниже представлена экранная форма главной страницы. Для того чтобы создать новый курс, нужно нажать иконку «Создать» в навигационной панели или на кнопку «Создать курс» (см. Рисунок 14).

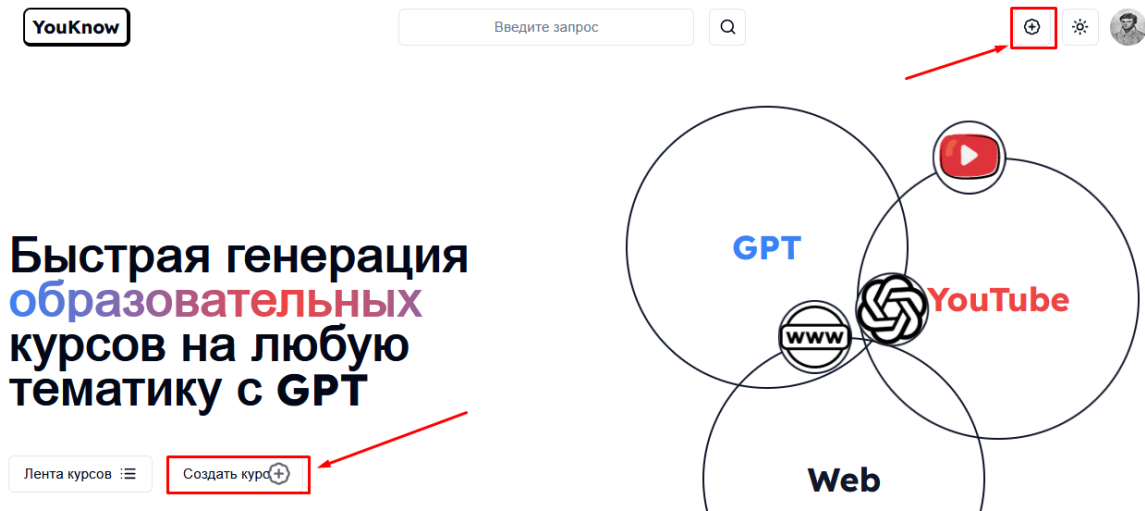


Рисунок 14. Экранная форма главной страницы

Далее представлена экранная форма создания курса. Пользователю необходимо указать название предмета и добавить разделы, которые он хочет изучить, далее нужно нажать кнопку «Поехали» и ожидать генерации будущей структуры курса (см. Рисунок 15).

Рисунок 15. Экранная форма создания курса

Далее представлена экранная форма сгенерированного содержания. Пользователю необходимо нажать кнопку «Создать» чтобы наполнить курс видеуроками и формами контроля усвоения (см. Рисунок 16).



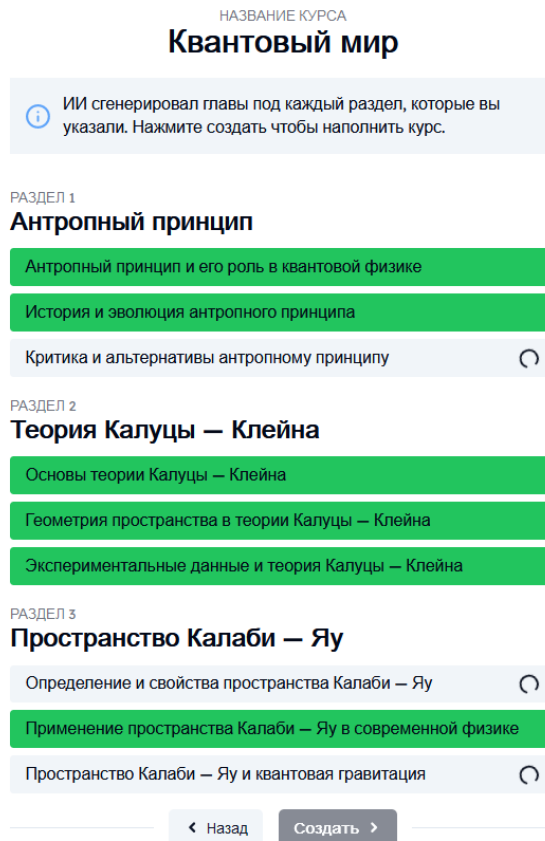


Рисунок 16. Экранная форма содержания

Ниже представлена экранная форма созданного курса (см. Рисунок 17).

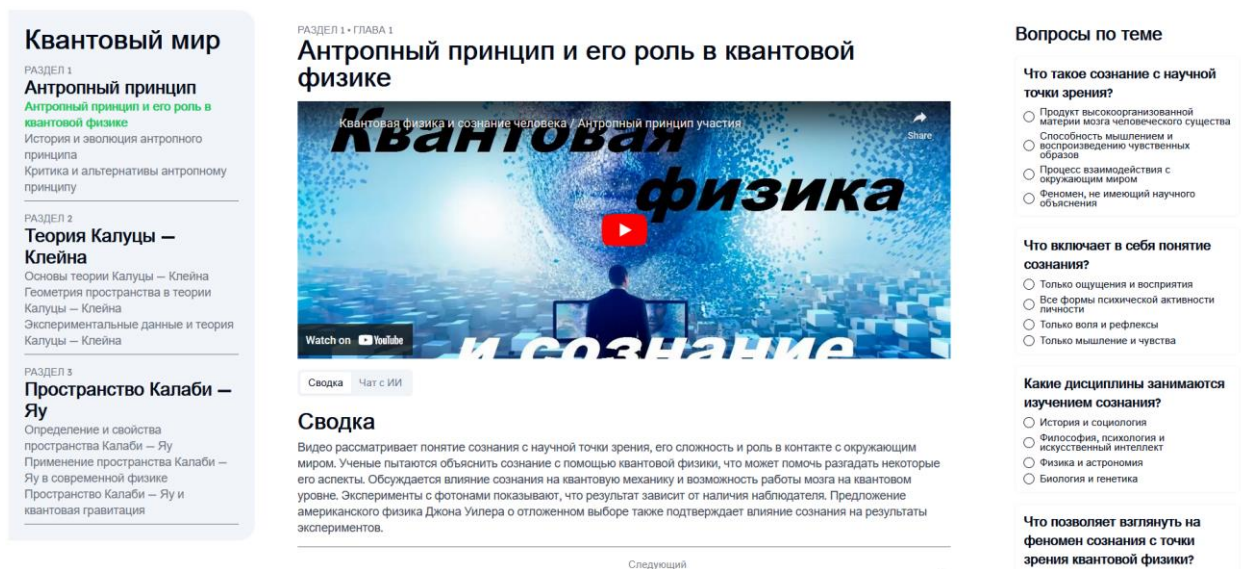


Рисунок 17. Экранная форма созданного курса

В представленном ниже коде (см. Рисунок 18) реализована функция `checkResult`, которая предназначена для анализа текста на предмет наличия запрещенных тематик и принятия решения о возможности его публикации. Функция использует фреймворк LangChain и его компоненты для структурирования вывода и обработки данных.

— Инициализация и формирование запроса: сначала формируется строка `summary`, объединяющая название (`title`) и разделы курса (`units`). Затем создается схема ответа (`ResponseSchema`), определяющая ожидаемый формат вывода – булево значение (`true/false`), указывающее на решение о публикации. Эта схема добавляется в список схем ответов (`response\_schemas`).

— Настройка парсера вывода: используя `StructuredOutputParser`, создается парсер вывода на основе предоставленных схем ответов. Этот парсер будет использоваться для преобразования ответа модели в структурированный формат.

— Формирование шаблона запроса: создаётся шаблон запроса (`template\_string`), который включает инструкции для модели по формату ответа и содержит переменные для динамической подстановки данных (`summary`, `policies`, `format\_instructions`). Шаблон используется для генерации конкретного запроса к модели.

— Отправка запроса и получение ответа: запрос, сформированный с помощью шаблона, отправляется в модель через метод `self.openai.chat`. Ответ модели затем обрабатывается с помощью ранее созданного парсера вывода, преобразуя его в структурированный формат.

— Возврат результата: результат, полученный после обработки ответа модели, сериализуется в JSON-строку и возвращается из функции.

```

async def checkResult(self, title, units, policies):

    summary = title + ' ' + units
    response_schemas = []
    chapters_schema = ResponseSchema(name=f'decision', description=f'decision about publishing', type='true/false')
    response_schemas.append(chapters_schema)
    output_parser = StructuredOutputParser.from_response_schemas(response_schemas)
    format_instructions = output_parser.get_format_instructions()

    template_string = """Ты - помощник, способный проверить решение [ ] публикации курса. \
Твоя задача - проверить текст ```{summary}``` на любое содержание или/и намек на наличие запрещенных тематик из списка: \
```{policies}```.

[ ] случае соответствия текста запрещенным тематикам в ответе верни false, иначе верни true. \

[ ] ответе верни JSON объект [ ] одним ключем и одним значением.

{format_instructions}
"""
    prompt = ChatPromptTemplate.from_template(template=template_string)
    messages = prompt.format_messages(summary=summary,
                                     policies=policies,
                                     format_instructions=format_instructions)

    response = self.openai.chat(messages)

    response_as_dict = output_parser.parse(response.content)
    result_json = json.dumps(response_as_dict, indent=4)

    return result_json

```

Рисунок 18. Функция проверки содержания

Такой подход позволяет автоматизировать процесс принятия решений о публикации контента, обеспечивая строгое соблюдение установленных политик и требований к содержанию курса. Использование `StructuredOutputParser` из фреймворка LangChain для структурирования вывода [3] обеспечивает гибкость и точность в обработке данных, делая систему безопасной как с точки зрения разработки, так и с точки зрения модерации информации.



## Информационная безопасность

LLM-платформа «АкадемИИЯ» согласно приказу ФСТЭК для АС [4] является многопользовательской, разграниченные права доступа, содержит персональные данные, соответствует классу АС 1Д.

А также, согласно приказу ФСТЭК № 21 [5] имеет персональные данные типа общедоступные, количество субъектов персональных данных менее 100 000 субъектов и третий тип угроз, ИС соответствует уровню защищенности ИСПДН № 4.

В соответствии с данными показателями были выбраны меры и подобраны средства защиты информации.

## Заключение

В данной статье была рассмотрена современная сфера онлайн-обучения, ее текущие проблемы и перспективы применения подхода программированного обучения на базе LLM. Разработка системы, основанной на больших языковых моделях и принципах программированного обучения, позволяет повысить эффективность онлайн-образования. Автоматизация процесса генерации курсов, проверка актуальности информации и возможность адаптации учебных материалов под потребности учащегося помогут решить основные проблемы самостоятельного онлайн-обучения. Представленная система способна изменить подход к онлайн-образованию, сделав его более доступным и персонализированным.

## Список использованных источников и литературы

1. Лю М.Х., Лю Ф., Фианнака А.Дж., Ку Т., Диксон Л., Терри М. и Кай Сиджей (2024). «Нам нужен структурированный вывод»: к ориентированным на пользователя ограничениям вывода больших языковых моделей. [Электронный ресурс] URL: <https://doi.org/10.48550/ARXIV.2404.07362> (дата обращения 01.06.2024).

2. Тимур Садыков. Систематический обзор подхода к программированному обучению в естественнонаучном образовании. [Электронный ресурс] URL: <https://doi.org/10.1080/2331186X.2023.2189889> (дата обращения 01.06.2024).

3. Анализатор структурированного вывода. Лангчейн. [Электронный ресурс] URL: [https://python.langchain.com/v0.1/docs/modules/model\\_io/output\\_parsers/types/structured/](https://python.langchain.com/v0.1/docs/modules/model_io/output_parsers/types/structured/) (дата обращения 01.06.2024).

4. Автоматизированные системы. Защита от несанкционированного доступа к информации. [Электронный ресурс] URL: <https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/rukovodyashchij-dokument-ot-30-marta-1992-g-3> (дата обращения 01.06.2024).

5. Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных. [Электронный ресурс] URL: <https://fstec.ru/dokumenty/vse-dokumenty/prikazy/prikaz-fstek-rossii-ot-18-fevralya-2013-g-n-21> (дата обращения 01.06.2024).

## List of references

1. Liu, M. X., Liu, F., Fiannaca, A. J., Koo, T., Dixon, L., Terry, M., & Cai, C. J. (2024). «We Need Structured Output»: Towards User-centered Constraints on Large Language Model Output. arXiv. URL: <https://doi.org/10.48550/ARXIV.2404.07362> (date of access 06.01.2024).
2. Timur Sadykov. A systematic review of programmed learning approach in science education, URL: <https://doi.org/10.1080/2331186X.2023.2189889> (date of access 06.01.2024).
3. Structured output parser. LangChain, URL: [https://python.langchain.com/v0.1/docs/modules/model\\_io/output\\_parsers/types/structured/](https://python.langchain.com/v0.1/docs/modules/model_io/output_parsers/types/structured/) (date of access 06.01.2024).
4. Automated systems. Protection against unauthorized access to information, URL: <https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/rukovodyashchij-dokument-ot-30-marta-1992-g-3> (date of access 06.01.2024).
5. On approval of the composition and content of organizational and technical measures to ensure the security of personal data during their processing in personal data information systems, URL: <https://fstec.ru/dokumenty/vse-dokumenty/priказы/prikaz-fstek-rossii-ot-18-fevralya-2013-g-n-21> (date of access 06.01.2024).