**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc199375038)

[ГЛАВА 1 ОБЗОР СУЩЕСТВУЮЩИХ СИСТЕМ КОНСУЛЬТИРОВАНИЯ ОБУЧАЮЩИХСЯ 6](#_Toc199375039)

[1.1 Классификация систем консультирования обучающихся 6](#_Toc199375040)

[1.2 Технологии искусственного интеллекта в системах консультирования 17](#_Toc199375041)

[1.3 Примеры внедрения систем консультирования 17](#_Toc199375042)

[1.3.1 Российский опыт 18](#_Toc199375043)

[1.3.2 Зарубежный опыт 18](#_Toc199375044)

[1.4 Проблемы и недостатки современных решений 19](#_Toc199375045)

[1.5 Тенденции развития систем в образовании 21](#_Toc199375046)

[ГЛАВА 2 ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К СИСТЕМЕ КОНСУЛЬТИРОВАНИЯ ОБУЧАЮЩИХСЯ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА 26](#_Toc199375047)

[2.1 Функциональные требования 26](#_Toc199375048)

[2.2 Требования к технической части 27](#_Toc199375049)

[2.3 Выбор инструментов и технологий разработки 29](#_Toc199375050)

[2.4 Обзор процесса разработки 30](#_Toc199375051)

[2.5.1 Схема базы данных 31](#_Toc199375052)

[2.5.2 Оптимизация базы данных 34](#_Toc199375053)

[2.6 Реализация серверной части 35](#_Toc199375054)

[2.7 Архитектура серверной части 35](#_Toc199375055)

[2.8 Интеграция модуля ИИ 37](#_Toc199375056)

[2.8.1 Логика обработки запросов 37](#_Toc199375057)

[2.8.2 Пример кода модуля ИИ 38](#_Toc199375058)

[2.9 Обновление пользовательского интерфейса 41](#_Toc199375059)

[2.9.1 Пример кода интерфейса 41](#_Toc199375060)

[2.9.2 Улучшение UX/UI 44](#_Toc199375061)

[2.10 Тестирование и отладка 45](#_Toc199375062)

[2.10.1 Пример тестов 45](#_Toc199375063)

[2.10.2 Выявленные проблемы и решения 45](#_Toc199375064)

[2.11 Возможные улучшения системы 46](#_Toc199375065)

[2.12 Итоги реализации 47](#_Toc199375066)

[ЗАКЛЮЧЕНИЕ 48](#_Toc199375067)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 50](#_Toc199375068)

[Приложение А 55](#_Toc199375069)

[Приложение Б 56](#_Toc199375070)

ВВЕДЕНИЕ

Актуальность исследования

Компьютерные технологии все больше проникают в различные сферы деятельности современного общества: бизнес, финансы, средства массовой информации, науку и образование. На общем фоне развития телекоммуникаций в нашей стране постепенно проявляется и становится заметным процесс внедрения компьютерных технологий в сферу образования [40]. компьютерные технологии получают все большее признание у преподавателей, потому что они:

* стимулируют процесс изучения любого предмета и облегчают обмен опытом преподавания различных учебных дисциплин;
* существенно повышают интерес учащихся к учебному процессу, в частности к овладению иноязычной речевой деятельностью на уроках иностранного языка;
* расширяют коммуникативную практику учащихся;
* делают возможным использование новых методических приемов, основанных на сопоставлении собственных данных учащихся и тех данных, которые они получают в результате общения [6].

Современное образование сталкивается с серьёзными задачами: нужно сделать обучение более индивидуальным и обеспечить студентов быстрой, качественной поддержкой. Чтобы справиться с этим, всё чаще используют различные информационные системы для помощи студентам. Несмотря на их известность и разнообразие, они нередко подводят, когда дело доходит до точных и полезных рекомендаций для студентов [14], [28].

В чём проблема? Эти системы не всегда учитывают, что каждый студент уникален — у всех разный уровень знаний, свои учебные программы и цели. Во-вторых, их ответы часто бывают слишком общими, потому что им не хватает глубокого понимания академического контекста [26]. Они опираются на широкие, универсальные данные, которые сложно приспособить к конкретным образовательным задачам. Представьте: вы спрашиваете, как подготовиться к экзамену по математике, а в ответ получаете стандартные советы, которые не имеют ничего общего с вашим курсом или уровнем подготовки. В итоге такие системы оказываются не слишком полезными в учёбе, где важны точность и персональный подход [26], [31].

Из-за этих ограничений возникает потребность в разработке новой системы консультирования обучающихся, которая сможет устранить указанные проблемы. Такая система должна предлагать персонализированные рекомендации, учитывать контекст учебных программ и обеспечивать точный подбор материалов, например, через специализированные базы данных, либо, что рассматривается в данной работе – через интеграцию с искусственным интеллектом. Искусственный интеллект имеет множество потенциальных приложений и возможностей в образовании. Он может анализировать информацию о детях, такую как их уровень образования и стиль обучения, и предоставлять персонализированные учебные материалы и инструкции, которые помогают каждому ребенку учиться в своем собственном темпе [1].

Решение этой задачи способно не только повысить эффективность обучения, но и сделать образовательные ресурсы более доступными, помочь студентам работать самостоятельно и разгрузить преподавателей, автоматизировав рутинные запросы [5], [41].

Цели и задачи работы

Цель работы

Разработка и реализация системы консультирования обучающихся с использованием технологий искусственного интеллекта, включающей базу данных, серверную и логическую части, а также пользовательский интерфейс, для повышения эффективности образовательного процесса путём предоставления ответов и рекомендаций.

Задачи работы

* Провести анализ современных систем консультирования и технологий ИИ в образовании.
* Сформулировать требования к системе, включая функциональные и нефункциональные аспекты.
* Спроектировать архитектуру системы, охватывающую базу данных, серверную часть, логическую обработку и интерфейс.
* Разработать базу данных для хранения запросов, ответов и пользовательских данных.
* Реализовать серверную часть для обработки запросов и взаимодействия с базой данных.
* Интегрировать алгоритмы ИИ для анализа запросов и генерации рекомендаций.
* Доработать прототип пользовательского интерфейса для динамического взаимодействия с сервером.
* Провести тестирование системы и проанализировать результаты её работы.

Эти задачи обеспечивают комплексный подход к созданию системы, ориентированной на потребности студентов и преподавателей в условиях цифрового обучения.

ГЛАВА 1 ОБЗОР СУЩЕСТВУЮЩИХ СИСТЕМ КОНСУЛЬТИРОВАНИЯ ОБУЧАЮЩИХСЯ

1.1 Классификация систем консультирования обучающихся

Под системами консультирования обучающихся будем понимать любые информационные системы, платформы и программы, которые тем или иным образом помогают студентам (школьникам, учащимся в колледжах и ВУЗах) освоить материал. Современные системы консультирования обучающихся являются важным элементом цифровизации образовательного процесса, обеспечивая автоматизированную поддержку студентов, преподавателей и административного персонала [21]. Эти системы используют информационные технологии, включая искусственный интеллект (ИИ), для повышения доступности образовательных ресурсов, персонализации обучения и оптимизации взаимодействия между участниками образовательного процесса. Их внедрение способствует снижению нагрузки на преподавателей, улучшению качества консультаций и адаптации к индивидуальным потребностям обучающихся. В данной главе рассматриваются основные типы систем консультирования, их функциональные возможности, используемые технологии, примеры внедрения, а также проблемы и перспективы развития [3], [4].

Системы консультирования обучающихся можно классифицировать по нескольким критериям: функциональности, технологической основе и целевой аудитории. По функциональности выделяются следующие категории:

**Системы управления обучением** (LMS), которые представляют собой интегрированные программные комплексы, обеспечивающие организацию учебного процесса, включая регистрацию пользователей, структурирование образовательного контента, мониторинг успеваемости и предоставление отчетов преподавателям [12]. Такие системы позволяют создавать, хранить и многократно использовать учебные материалы, такие как тексты, презентации и мультимедийные ресурсы, адаптируя их под различные дисциплины и группы обучающихся.

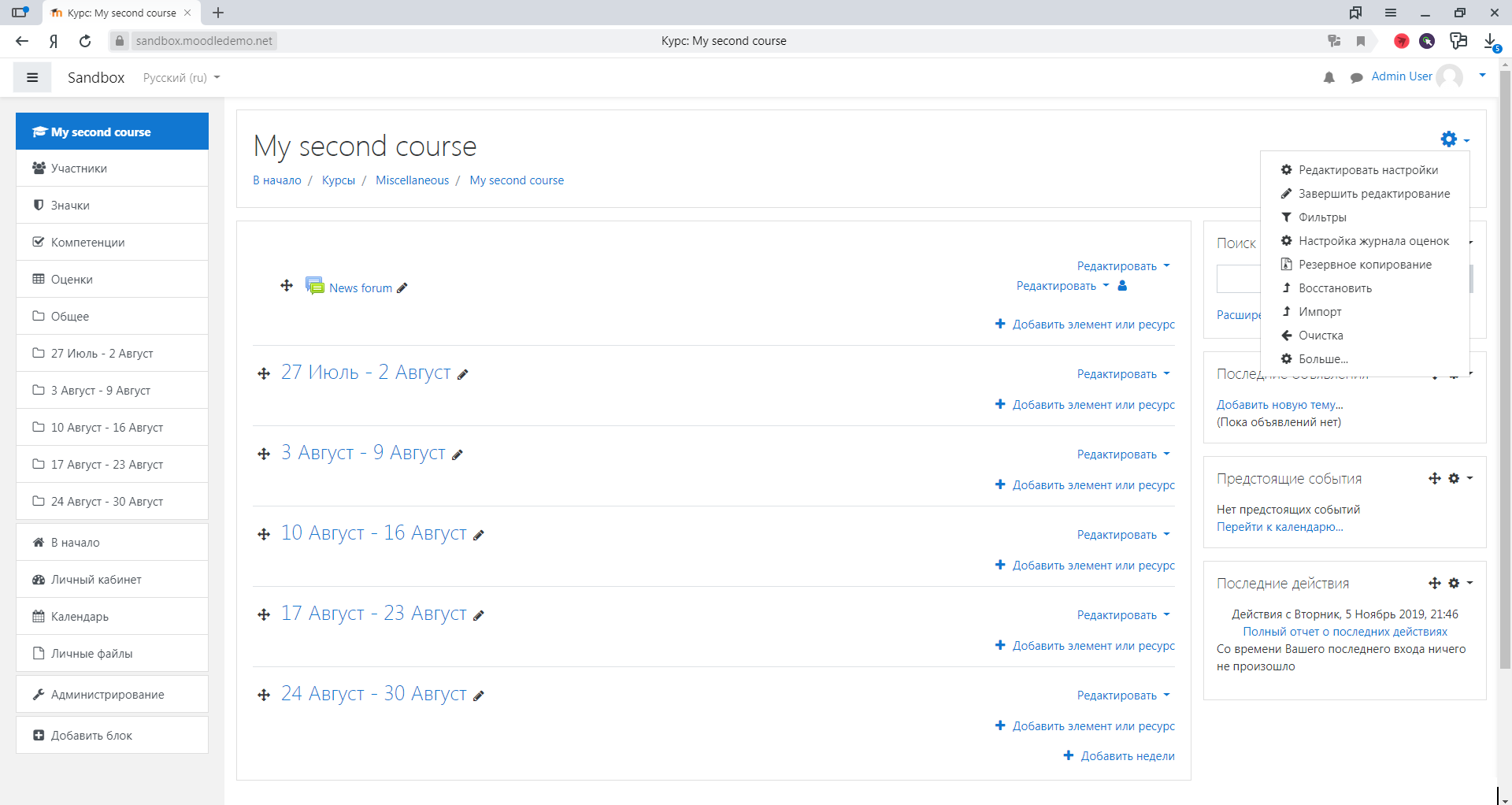


Рисунок 1 Интерфейс Moodle

**Moodle** (Рисунок 1) — это система управления курсами (LMS — Learning Management System). Она также известна как система управления содержанием или как виртуальная среда обучения. Moodle представляет собой свободное (распространяющееся по лицензии GNU GPL) веб-приложение, которое дает возможность создавать сайты для онлайн-обучения, разработанное на языке программирования php и с системой управления базами данных MySQL, PostgreSQR, Oracle, Assess, Interbase. Предоставляется как ПО с открытым исходным кодом под стандартной общественной лицензией GNU General Public License, пользователи которого имеют право на неограниченную установку, использование, распространение и изменение Moodle без каких-либо лицензионных сборов., которая предоставляет такие функции, как форумы, вики и тесты. Она позволяет студентам взаимодействовать с учебным контентом и друг с другом. Однако её возможности по обработке сложных запросов ограничены, так как она не оснащена искусственным интеллектом (ИИ) для анализа контекста или предоставления персонализированных ответов.

Пример: представим студента, который изучает курс по машинному обучению и задаёт вопрос: "Какие ресурсы лучше всего подходят для понимания нейронных сетей?" Moodle не сможет предложить конкретные рекомендации, учитывающие уровень знаний студента или специфику курса. Вместо этого система предоставит доступ к общим материалам или перенаправит студента на форум, где он будет ждать ответа от преподавателя или однокурсников. Это может занять часы или даже дни.

Другой проблемой, на которую указали студенты и преподаватели, описывая свой опыт работы в Moodle, — это проблема с регистрацией. Несмотря на, казалось бы, простой интерфейс, студенты испытывают затруднения при регистрации и часто обращаются за помощью к преподавателям или IT-специалистам университета

**Blackboard** (Рисунок 2) — система дистанционного обучения (СДО, англ.: Learning Management System (LMS)), обеспечивающая создание и функционирование единой интерактивной среды обучения, взаимодействия, обмена информацией между обучаемыми и преподавателями. Это хорошо развитое программно-аппаратное обеспечение, помогает управлять виртуальной средой обучения, создавать электронные образовательные ресурсы, обеспечивать удаленный доступ к образовательным ресурсам учебного заведения, осуществлять контроль образовательного процесса, предоставлять платформы для курсов дистанционного обучения и обладает закрытым исходным кодом. Для поддержки функционирования СДО BlackBoard необходим администратор, который обеспечивает установку и следит за работоспособностью системы, а также осуществляет управление и обработку учетных записей пользователей. Подавляющее число американских учебных заведений используют именно эту платформу для электронного образования. Однако, как и Moodle, она не способна эффективно обрабатывать сложные запросы, оставляя студентов зависимыми от внешней помощи.

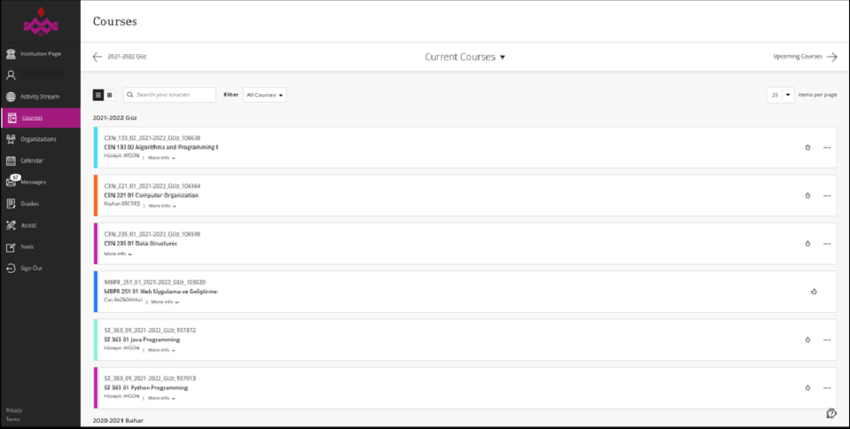


Рисунок 2 Интерфейс Blackboard

Пример: Студент работает над проектом по анализу данных и спрашивает: "Какой алгоритм лучше использовать для кластеризации этого набора данных?" Blackboard не сможет проанализировать специфику вопроса или предложить подходящий алгоритм (например, K-means или DBSCAN). Студенту придётся либо самостоятельно искать информацию в учебных материалах, либо обращаться к преподавателю, что замедляет процесс обучения.

**Canvas** (Рисунок 3) - Даная платформа предоставляет индивидуальный и совместный доступ зарегистрированным пользователям. Преподаватель имеет возможность создать отдельный курс под конкретную дисциплину и конкретную группу. Параллельные группы потока можно объединять в общем онлайн-классе, но в разных подгруппах. То же относится и к преподавателям одной дисциплины: при желании можно создать один курс для всех преподавателей и групп потока, при этом объявления и материалы будут в общем доступе, а задания и тестовые мероприятия могут выполняться как в едином пространстве, если мы говорим о совместных проектах, так и в подгруппах. Под совместными проектами подразумеваются студенческие работы и место, где преподаватели смогут делиться наблюдениями, методическими наработками, материалами и т.д. Совместный доступ преподавателей к заданиям, контрольным и экзаменационным работам студентов способствует повышению объективности оценивания с помощью процедуры двойной проверки. Система выделяется своим удобным интерфейсом и возможностями интеграции с другими образовательными инструментами. Тем не менее, он также не обеспечивает глубокую поддержку для сложных запросов, вынуждая студентов искать помощь за пределами системы.

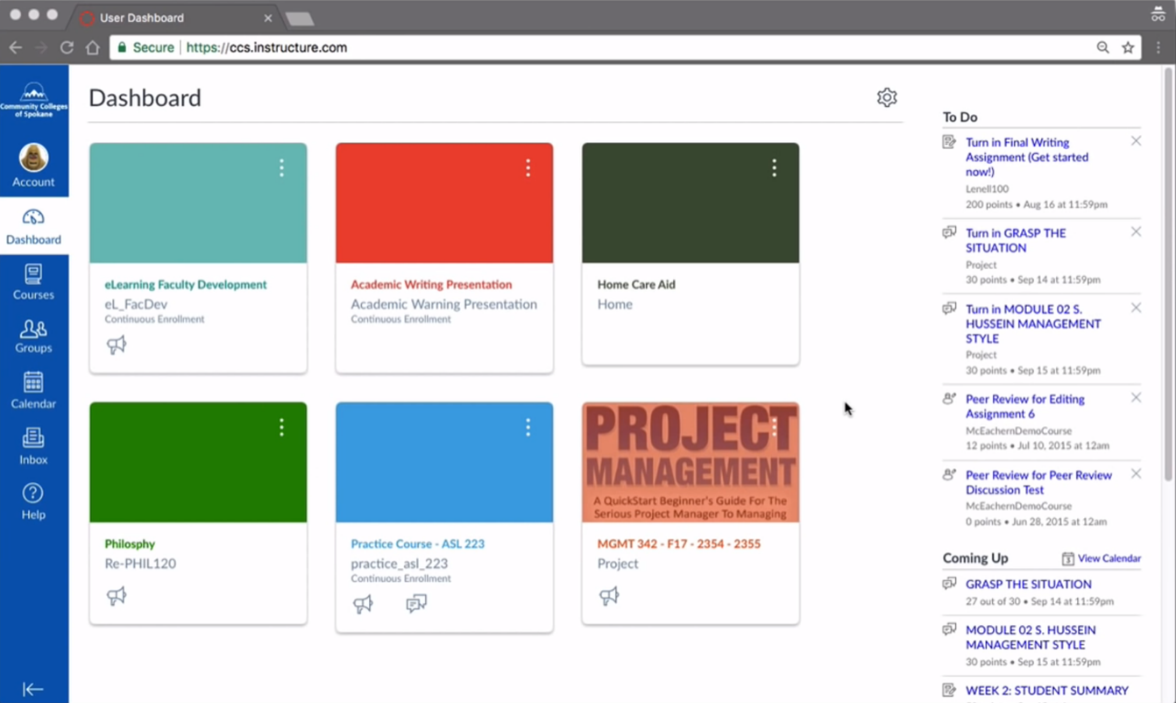


Рисунок 3 Интерфейс Canvas

Пример: Студент, изучающий программирование, задаёт вопрос: "Почему мой код не работает?" Canvas не способен проанализировать код, выявить ошибку (например, синтаксическую или логическую) или предложить исправление. Студенту придётся разместить вопрос на форуме или отправить его преподавателю, что может привести к задержкам в решении проблемы.

Основная цель систем, таких как Moodle, Blackboard и Canvas, — это управление контентом и организация учебного процесса, а не обработка сложных запросов. Их ключевые недостатки включают:

* Недостаток контекстного понимания
* Отсутствие диалоговой поддержки
* Зависимость от преподавателей

Эти ограничения особенно заметны, когда студенты сталкиваются с задачами, требующими персонализированной помощи или быстрого ответа.

**Чат-боты на основе ИИ** (Рисунок 4): автоматизированные системы, использующие технологии обработки естественного языка (NLP) для анализа текстовых запросов и генерации ответов в реальном времени Чат-боты представляют собой программы, разрабатываемые человеком и для человека на основе технологии машинного обучения и нейросетей под определённый набор целей. Фактически чат-бот — это автоматизированный и персонализированный чат между компьютерной системой и пользователем, своеобразный «младший» партнёр [18], [38].

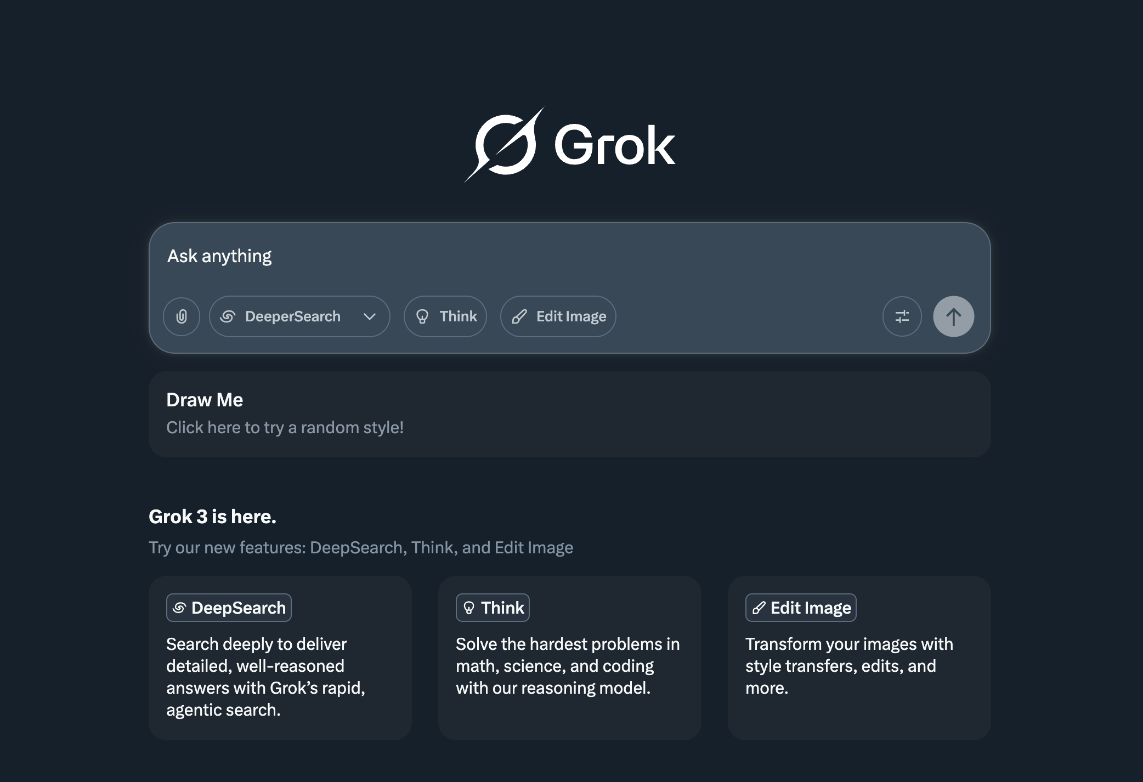


Рисунок 4 Интерфейс GPT Чата

Чат-боты сочетают в себе две важные составляющие: они многозадачны, так как позволяют автоматизировать целый ряд процессов (осуществление консультаций, проведение контрольных мероприятий и экзаменов, проверку результатов тестирования, ведение опросов обучающихся для выявления слабых мест в образовательных программах), и удобны для взаимодействия с пользователем за счёт комфортного формата общения, имитирующего разговор с собеседником [10], [37].

Одной из современных сфер применения чат-ботов является образование и уже создано и используется множество чат-ботов, способствующих получению и закреплению знаний, а также проверке их усвоения. Существуют исследования эффективности использования чат-ботов в образовательном процессе, в ходе которых установлено, что большинство участников экспериментов учились, усваивали информацию и общались с ботами так же, как если бы это были настоящие люди. Более того, на основании изучения общения пользователей с чат-ботами исследователи сообщили о появлении особой социотехнической системы «Пользователь-бот» как системы подходов к организации труда в контексте взаимодействия человека и чат-ботов [5], [46].

Программирование и запуск собственного чат-бота, основанного на машинном обучении, является сложным процессом, предполагающим наличие квалифицированных разработчиков и специалистов по интерфейсам, а также значительных временных и ресурсных затрат. Однако в настоящее время существует возможность быстро создать простой чат-бот, не требующий особых технических навыков и знания языков программирования [25].

Одна из причин активного интереса к применению чат-ботов в различных областях жизнедеятельности — высокая эффективность их использования, так как чат-бот, в отличие от человека, может работать круглосуточно, обеспечивая, например, предоставление информации в режиме 24/7. Этот фактор важен и в случае с образованием, поскольку чат-бот может работать как круглосуточная служба поддержки, позволяя учителям избежать необходимости отвечать на повторяющиеся вопросы, на которые легко и быстро может дать ответ чат-бот. Помимо этого, чат-боты помогают вовлечь студентов в образовательный процесс, заинтересовать обучающихся в усвоении материала за счёт постоянной коммуникации с пользователем, которая является важным средством получения обратной связи и, как следствие, способствует повышению качества образовательных программ Однако внедрению чат-ботов в образование всегда должно предшествовать предварительное обсуждение, независимо от того, как планируется применять чат-бот — в учебном процессе или для решения каких-то организационных задач. Важно подчеркнуть, что включение чат-ботов в информационно-образовательную среду образовательной организации не заменит преподавательский состав, администрацию и обслуживающий персонал, но бот может взять на себя выполнение некоторых из их задач. На данный момент чат-боты не смогут заменить учителя, они должны взять на себя низкоинтеллектуальные повторяющиеся задачи, что позволит учителю, свободному от выполнения таких задач, посвятить свое время более высоким, в когнитивном плане, задачам [29], [42].

Чат-боты имеют достаточно хорошие перспективы использования в сфере образования. Их применение дает возможность предоставить учащимся доступ к различным учебным материалам, интерактивным задачникам и т. п. в круглосуточном режиме [35].

GPT бот – это продвинутая система искусственного интеллекта. Он может понимать и генерировать текст на естественном языке, что делает его мощным инструментом для решения различных проблем в образовании, коммуникации, исследованиях, разработках и других областях. В образовании его можно использовать для автоматизации задач и выполнения многих других задач. Его способности проистекают из огромного количества информации и обучения на миллионах текстовых примеров. Однако важно помнить, что бот не имеет реального понимания и не имеет собственного сознания. Он работает на основе статистических закономерностей в тексте и не может учиться на собственном опыте или испытывать чувства. Поэтому при его использовании, особенно в критических средах, необходимо тщательно проверять создаваемый им контент. Однако его эффективное использование требует от пользователей точности и ответственности, чтобы максимизировать его преимущества и минимизировать возможные недоразумения или ошибки [22], [45].

**Адаптивные образовательные технологии** являются одним из наиболее перспективных направлений в образовании, позволяя индивидуализировать обучение для каждого учащегося. Адаптивные образовательные технологии представляют собой системы, способные адаптироваться к индивидуальным потребностям, стилю обучения и темпу усвоения знаний каждого учащегося [3]. Эти технологии основаны на использовании современных информационных и коммуникационных технологий, таких как искусственный интеллект, машинное обучение и анализ данных [2].

Преимущества адаптивных образовательных технологий включают:

* Индивидуализация обучения: Системы адаптивного обучения позволяют каждому учащемуся учиться в соответствии с его уровнем знаний, интересами и потребностями. Это способствует более эффективному усвоению материала.
* Повышение мотивации: Подходы адаптивного обучения часто включают элементы игр и наград, что способствует повышению мотивации учащихся к учебной деятельности.
* Более эффективное использование времени: Системы адаптивного обучения позволяют более быстро пройти материал, который учащийся усвоил хорошо, и уделить больше времени сложным темам.

**Примеры платформ**

* **Khan Academy** (Рисунок 5): это платформа, предоставляющая образовательные материалы в виде видеоуроков и практических заданий. Система адаптивного обучения Khan Academy анализирует ответы учащегося на задачи и, основываясь на этих данных, предоставляет дополнительные упражнения для закрепления материала или перехода к более сложным темам. Специализируется на математике и использует ИИ-помощника Khanmigo, который применяет сократский метод, помогая студентам самостоятельно находить решения. «Khanmigo использует подход, направленный на развитие критического мышления, а не на предоставление готовых ответов».
* **Duolingo**: Фокусируется на изучении языков, применяя большие языковые модели для создания персонализированных уроков и моментальной обратной связи. Использование больших языковых моделей позволяет генерировать разнообразные учебные материалы, адаптированные к уровню пользователя.
* **Coursera**: Предлагает курсы по программированию и другим дисциплинам, используя ИИ для рекомендаций курсов, виртуального коучинга и создания контента. ИИ на Coursera помогает формировать персонализированные рекомендации и автоматизировать создание учебных материалов.

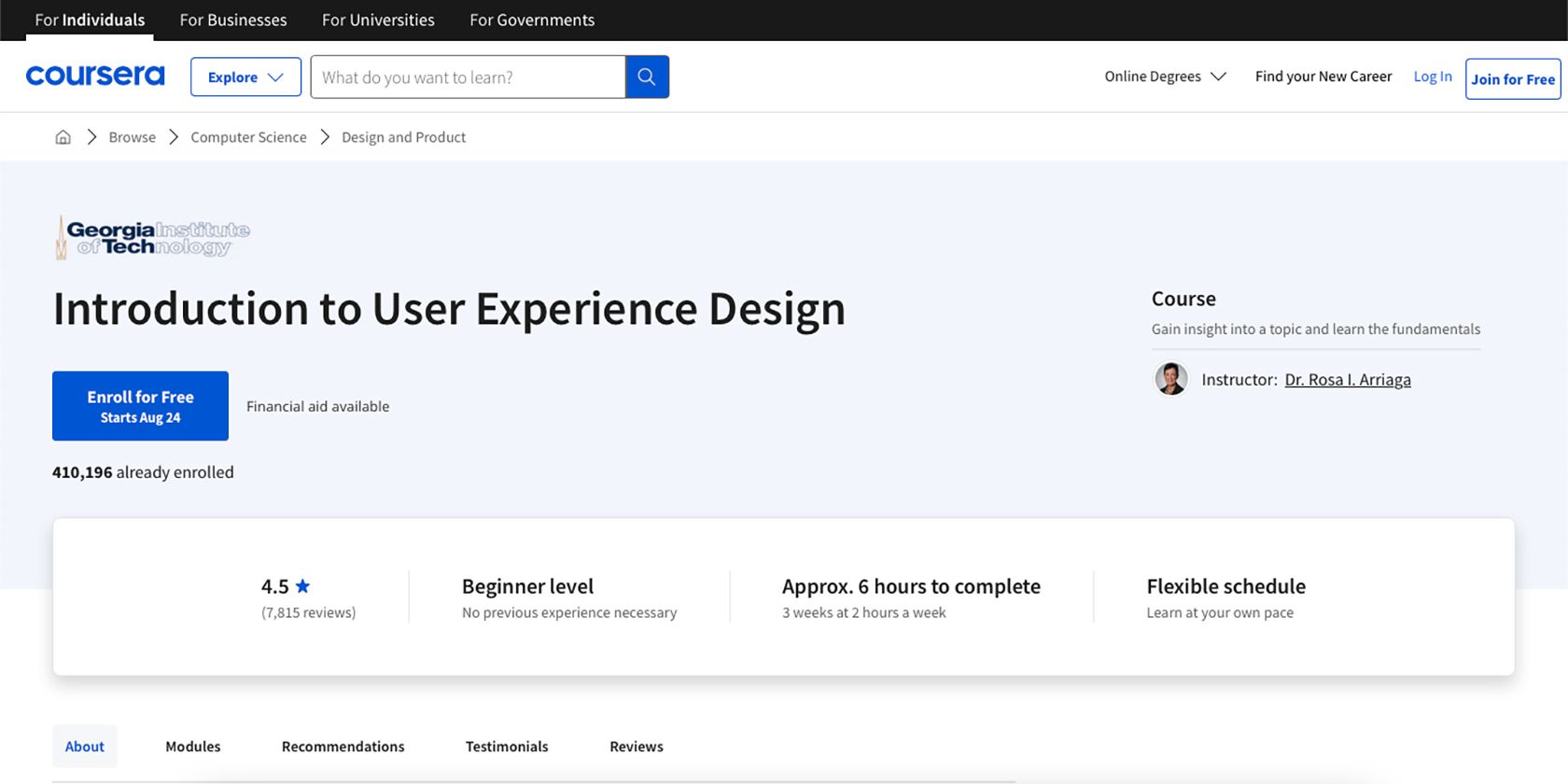


Рисунок 5 Платформа персонализированного обучения

Искусственный интеллект (ИИ) и машинное обучение играют ключевую роль в реализации адаптивных образовательных технологий. Алгоритмы машинного обучения способны анализировать данные обучающего процесса, выявлять паттерны в ответах учащихся и определять их уровень понимания материала. Эта информация затем используется для настройки содержания уроков и заданий в соответствии с потребностями каждого конкретного ученика [2].

Одной из главных составляющих успешной адаптивной системы обучения является анализ данных. Платформы собирают информацию о прогрессе каждого ученика, его реакциях на различные уроки и задания, а также времени, затраченном на изучение материала. На основе этих данных системы создают профиль обучающегося, который позволяет определить его индивидуальные потребности и оптимальные способы обучения.

Персонализация материала включает в себя предоставление дополнительных упражнений для тех тем, которые вызывают затруднения, или переход к более сложным темам для более продвинутых учеников. Этот подход позволяет каждому ученику двигаться вперед с оптимальной скоростью и наилучшим пониманием.

Несмотря на значительные достижения в области адаптивных образовательных технологий, существуют и некоторые вызовы. Например, системы адаптивного обучения нуждаются в надежных методах оценки знаний учащихся, чтобы точно определить их уровень понимания. Кроме того, важно обеспечить баланс между технологическими инновациями и ролью учителя, который остается важным источником мотивации и поддержки для учащихся [3].

С учетом динамичного развития технологий и образовательных методик, будущие исследования в области адаптивных образовательных технологий в обучении могут расширить существующее понимание. Одним из направлений может стать более глубокое исследование влияния адаптивных технологий на мотивацию и интерес учащихся в получении новых знаний. Также стоит исследовать эффективность адаптивных систем в разных контекстах, включая разные уровни образования и культурные особенности [2],

**Гибридные системы**, такие как EdX, объединяют элементы информационных систем, чат-ботов и персонализированного обучения. «Эти системы сложнее в разработке, но обеспечивают более широкий охват образовательных задач за счёт интеграции различных функций». Например, EdX предоставляет автоматизированные ответы, рекомендации и доступ к ресурсам, что делает её более универсальной по сравнению с узкоспециализированными платформами.

По технологической основе системы делятся на традиционные (базирующиеся на статических базах данных и скриптах) и ИИ-ориентированные (использующие NLP, машинное обучение, глубокое обучение). По целевой аудитории выделяются системы для студентов, преподавателей или администрации, а также универсальные решения.

1.2 Технологии искусственного интеллекта в системах консультирования

Искусственный интеллект, особенно технологии обработки естественного языка, играет ключевую роль в современных системах консультирования. Основные технологии включают:

Обработка естественного языка (NLP): позволяет системам анализировать текстовые запросы, извлекать ключевые слова и генерировать релевантные ответы. Библиотеки, такие как spaCy, обеспечивают токенизацию, лемматизацию и анализ синтаксиса, что критично для обработки вопросов на русском языке. Например, нормализация слов (например, «математике» → «математика») повышает точность поиска литературы [17].

Машинное обучение (ML): используется для классификации запросов, предсказания намерений пользователя и формирования персонализированных рекомендаций. Алгоритмы, такие как SVM или нейронные сети, помогают системам адаптироваться к индивидуальным профилям студентов [16].

Чат-боты на основе больших языковых моделей (LLM): обеспечивают диалоговую поддержку, генерируя естественные ответы. Однако их внедрение в образовательные системы требует тонкой настройки под академический контекст.

Семантический поиск: позволяет находить релевантные ресурсы даже при неточном формулировании запроса. Это особенно важно для поиска литературы в базах данных.

Применение ИИ повышает гибкость систем, но требует значительных вычислительных ресурсов и качественных данных для обучения. В российских образовательных учреждениях технологии NLP часто адаптируются под особенности русского языка, что подчёркивает важность библиотек, таких как pymorphy3, для морфологического анализа.

1.3 Примеры внедрения систем консультирования

1.3.1 Российский опыт

В России системы консультирования активно внедряются в рамках цифровизации образования. Примеры включают:

Платформа «Сферум»: информационно-коммуникационная система для школ и вузов, включающая чат-боты для ответов на организационные вопросы (расписание, дедлайны). Система интегрируется с «Госуслугами» и использует простые алгоритмы NLP. Сферум – это защищенная платформа для организации учебного процесса, дающая возможность коммуницировать всем его участникам. Сферум сочетает в себе все необходимые для учебы функции: видеоуроки, звонки, чаты, трансляции, возможность обмениваться образовательный контентом.

Чат-бот МФТИ: разработан для поддержки студентов в навигации по учебным ресурсам и консультациях по физико-математическим дисциплинам. Использует базу знаний и ограниченные элементы ИИ [21].

Платформа «Учи.ру» разработана для школьников и предоставляет доступ к задачам по математике и информатике, сопровождаемым поддержкой чат-бота для объяснения сложных моментов. Вот только её функционал слишком ориентирован на школьную программу, что делает её недостаточно надёжной для ответа на вопросы, требующих знаний из различных дисциплин, например, установления связей между математикой и биологией [4].

1.3.2 Зарубежный опыт

За рубежом системы консультирования более разнообразны благодаря инвестициям в ИИ:

Georgia Tech’s Jill Watson: чат-бот на основе IBM Watson, внедрённый для поддержки студентов в курсе по ИИ. Отвечает на вопросы по материалам и форумам, демонстрируя высокую точность. Однако его разработка потребовала значительных затрат.

Deakin University’s Genie: австралийская платформа, объединяющая чат-бота, персонализированные рекомендации и доступ к LMS. Использует NLP для анализа запросов и ML для адаптации контента.

Duolingo’s Chatbot: интегрирован в языковые курсы, обеспечивая диалоговую практику. Ограничен языковой тематикой, но демонстрирует эффективность NLP в образовании.

Зарубежные системы часто опираются на большие языковые модели, что повышает их универсальность, но увеличивает затраты на внедрение.

1.4 Проблемы и недостатки современных решений

Современные системы консультирования, несмотря на достижения, сталкиваются с рядом проблем, ограничивающих их эффективность

Когда возможности ИИ усиливаются, этические аспекты его использования становятся еще более важными. Вот в чем суть: с большой силой приходит большая ответственность.

Одной из самых насущных проблем является подлинность работ, созданных авторами эссе ИИ. Если студент отправляет эссе, изначально созданное с помощью инструмента ИИ, можем ли мы действительно сказать, что это оригинальная работа студента? Это стирание границ между человеческими усилиями и результатами машин бросает вызов нашему традиционному пониманию авторства и оригинальности. Возникает вопрос: продвигаем ли мы непреднамеренно культуру, в которой процесс мышления, анализа и творчества передан машинам [34]?

Кроме того, эпоха ИИ представляет собой нюансированную форму вековой проблемы плагиата. Даже если инструменты ИИ могут генерировать уникальный контент, тень сомнения в его оригинальности сохраняется. Речь идет не только о извлечении контента из существующих источников; речь идет о генезисе самой идеи. И даже если технически это не плагиат, поддерживает ли он дух академической честности?

Хотя ИИ продемонстрировал замечательные способности в решении различных задач, его надежность остается предметом споров. Машины работают на основе алгоритмов и данных, которые не всегда могут отражать нюансы и сложности человеческого мышления. Если полагаться исключительно на суждения ИИ, это может привести к неправильным представлениям и неточностям.

Сегодня данные стали новым золотом, поэтому существует множество проблемы конфиденциальности данных. Поскольку студенты все чаще обращаются к онлайн-инструментам искусственного интеллекта для академической помощи, они часто делятся личной информацией, эссе и исследованиями. Но какой ценой? Растут опасения по поводу того, как хранятся эти данные, кто имеет к ним доступ и их возможное неправомерное использование. Нарушают ли студенты непреднамеренно свою конфиденциальность в обмен на удобство, которое предлагают услуги, основанные на искусственном интеллекте?

В академической среде суть обучения заключается не только в получении информации, но и в оригинальности мысли и способности к инновациям. Нельзя отрицать, что ИИ обладает способностью генерировать огромное количество контента, часто имитируя человеческие манеры письма. Однако, хотя он может воспроизводиться, он не обязательно вводит новшества так, как это делают люди. Человеческий разум опирается на опыт, эмоции, культуру и множество других факторов, которыми ИИ пока просто не обладает. Нюансы, интуиция и полная непредсказуемость человеческого творчества сложны, если не невозможны, для ИИ полностью подражать. Может ли машина по-настоящему уловить суть момента озарения или острые ощущения от неожиданной связи? Доступность: высокая стоимость разработки и обслуживания ИИ-систем делает их недоступными для учреждений с ограниченным бюджетом. Это усиливает цифровое неравенство, особенно в регионах [34].

Интерфейсы. Сложные системы, такие как Moodle, зачастую оказываются малопонятными для пользователей с низким уровнем цифровой грамотности. Их пользовательские интерфейсы нередко требуют значительного упрощения и лучшей адаптации к мобильным устройствам.

Локализация. Во многих случаях цифровые платформы не учитывают культурные и языковые различия, что снижает их эффективность в многонациональной среде. Например, большинство зарубежных чат-ботов редко поддерживают русский язык, что ограничивает их применимость.

Ограниченная универсальность. Многие системы изначально ориентированы на решение узкоспециализированных задач — например, преподавание языков, ответы на типовые вопросы (FAQ) или помощь в одной конкретной дисциплине. Это значительно ограничивает их способность эффективно работать с междисциплинарными вопросами, требующими комплексного подхода и знаний из различных областей.

Большинство консультационных систем проектируются под определённые цели и аудитории, что порождает ряд ограничений:

Узкая направленность алгоритмов. Системы наподобие Duolingo или Khan Academy используют алгоритмы, адаптированные под конкретные задачи — будь то языковое обучение или решение задач по математике. Такие алгоритмы не справляются с запросами, выходящими за рамки их специализации.

Ограниченность используемых данных. Часто базы знаний этих систем охватывают лишь одну дисциплину или тип задач. Например, FAQ-бот, предназначенный для помощи с расписанием, не содержит информации, необходимой для решения междисциплинарных задач.

Недостаток контекстного анализа. В большинстве случаев отсутствует учет сложных взаимосвязей между предметами или индивидуальных особенностей обучающихся, что делает ответы таких систем поверхностными при работе с многослойными и нестандартными запросами.

1.5 Тенденции развития систем в образовании

Цифровизация образования стимулирует развитие систем консультирования, применяющих искусственный интеллект (ИИ) и персонализацию, что даёт возможность адаптировать образовательные процессы к индивидуальным нуждам учащихся. Эти системы всё чаще интегрируют в образовательные экосистемы, обеспечивая масштабируемость, доступность и вовлечённость. Основные тенденции включают глубокую персонализацию, многоязычную поддержку, интеграцию с мессенджерами, использование облачных технологий, геймификацию и акцент на этичный ИИ. Разрабатываемая в последующих главах система, описанная, учитывает эти направления, фокусируясь на обработке естественной речи (NLP), адаптивных интерфейсах и локализации под российский образовательный контекст [35].

**1. Глубокая персонализация**

Современные подходы к индивидуализации образовательного процесса активно используют технологии машинного обучения и анализа больших данных. Такие цифровые образовательные платформы формируют персонализированные траектории обучения, адаптируя содержание и сложность заданий в зависимости от прогресса и поведенческих характеристик обучающегося (Тихоновецкая, 2023). Например, при выявлении сложностей с определёнными темами, система автоматически предлагает обучающие материалы и упражнения, направленные на устранение пробелов. В российских школах и вузах, где уровень подготовки учащихся может значительно различаться, такие технологии способствуют сокращению разрыва в знаниях и повышению мотивации. Использование ИИ в системах образования позволяет создавать адаптивные среды, учитывающие индивидуальные когнитивные особенности студентов.

**2. Многоязычная поддержка**

Развитие глобализированного образования требует внедрения многоязычных интеллектуальных систем поддержки, особенно в университетах с высоким числом иностранных студентов. Применение моделей обработки естественного языка, таких как mBERT, позволяет обрабатывать запросы на различных языках, обеспечивая инклюзивное взаимодействие между студентами и администрацией. В российских университетах, таких как РУДН и НИУ ВШЭ, активно внедряются многоязычные чат-боты, поддерживающие русский, английский и китайский языки. Эти системы облегчают процесс адаптации и поступления иностранных студентов за счёт автоматизированной обработки запросов, связанной с приёмной кампанией, академическим расписанием и миграционными вопросами. Интеграция NLP-алгоритмов в инфраструктуру вузов демонстрирует высокую эффективность в контексте многоязычной поддержки, включая региональные языки, такие как башкирский или якутский, что особенно актуально в условиях этнолингвистического разнообразия России. [27]

**3. Интеграция с мессенджерами**

Интеграция систем цифрового консультирования с популярными мессенджерами, такими как Telegram и ВКонтакте, усиливает доступность информационных сервисов для студентов, предоставляя им возможность получать ответы в привычной и удобной среде. Особенно это актуально в российском контексте, где данные платформы являются основными каналами коммуникации молодёжи. Вузы, включая МФТИ, СПбГУ и ТПУ, внедряют чат-ботов для информирования о расписании, сроках сдачи заданий и предоставления ссылок на учебные материалы. Такие решения позволяют оперативно решать типовые вопросы и снижать нагрузку на административный персонал. Кроме того, Telegram-боты применяются в приёмных комиссиях и академических службах вузов, автоматизируя рутинные процессы и повышая удовлетворённость пользователей. Эти разработки подчёркивают роль мессенджеров как ключевых каналов цифровой трансформации образовательной среды.

**4. Облачные технологии**

Облачные вычислительные технологии, такие как Yandex.Cloud, AWS и Microsoft Azure, обеспечивают высокую масштабируемость, надёжность и доступность цифровых образовательных систем, включая интеллектуальных ассистентов и обучающие платформы. Они позволяют значительно снизить затраты на локальную инфраструктуру, что критично для образовательных учреждений с ограниченным бюджетом. Вузы России, включая Новосибирский государственный университет (НГУ), активно применяют облачные решения для размещения LMS-платформ, чат-ботов и автоматизации административных процессов. Использование таких платформ, как Stepik, на базе облачных серверов, обеспечивает бесперебойную доставку интерактивных курсов тысячам пользователей одновременно. Это создаёт условия для гибкой масштабируемой архитектуры, необходимой для внедрения ИИ-сервисов в образовательную практику, включая обработку естественного языка, адаптацию контента и мониторинг прогресса студентов [20].

**5. Геймификация**

Геймификация становится важным инструментом повышения мотивации и вовлечённости студентов в образовательный процесс. Использование игровых элементов — таких как баллы, рейтинги, достижения — формирует интерактивную среду, стимулирующую регулярное взаимодействие с образовательными платформами. Как показывают исследования, такие механизмы положительно влияют на учебную активность и когнитивную вовлечённость обучающихся. Российская платформа «Учи.ру» активно внедряет геймификацию, предлагая звёзды и медали за выполнение заданий, что способствует формированию положительной учебной мотивации у школьников. Аналогично, в университетах и колледжах чат-боты могут начислять виртуальные баллы за участие в учебных активностях, выполнение тестов или посещение занятий, что позволяет преподавателям отслеживать динамику вовлечённости и прогресса студентов. Такие цифровые подходы особенно актуальны в условиях цифровой трансформации образования и необходимости адаптации к новым форматам обучения [23].

**6. Этичный ИИ**

Этичный искусственный интеллект (ИИ) ставит во главу угла прозрачность алгоритмов, защиту персональных данных и снижение предвзятости, что приобретает особую значимость в условиях строгих законодательных требований и общественных ожиданий. В России действует Федеральный закон № 152-ФЗ «О персональных данных», который регулирует обработку личной информации, включая требования к согласию, права субъектов данных и меры безопасности. Образовательные платформы, такие как Сферум, внедряют шифрование и прозрачные политики обработки данных, чтобы соответствовать этим нормам и укреплять доверие пользователей. Применение ИИ в образовательной сфере требует разработки прозрачных алгоритмов и строгого соблюдения принципов защиты данных для минимизации рисков. Соблюдение законодательства о персональных данных и обеспечение прозрачности алгоритмов имеют решающее значение для успешного внедрения ИИ-систем в российских университетах [8].

Россия активно разрабатывает правовую базу для регулирования ИИ, уделяя внимание совершенствованию этических и технических стандартов, которые служат ориентиром для разработчиков и пользователей. В частности, Российский кодекс этики искусственного интеллекта подчеркивает приоритетность человека, справедливость, прозрачность и безопасность на всех этапах жизненного цикла ИИ. Эти меры необходимы для ответственного использования ИИ в образовании и обеспечения его соответствия как юридическим, так и социальным ожиданиям.

Эти тенденции формируют перспективы для создания универсальных систем консультирования, способных интегрироваться в образовательные экосистемы и адаптироваться к разнообразным задачам. Разрабатываемая система, описанная в последующих главах, учитывает указанные направления, включая:

ГЛАВА 2 ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К СИСТЕМЕ КОНСУЛЬТИРОВАНИЯ ОБУЧАЮЩИХСЯ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

2.1 Функциональные требования

Для разработки эффективной системы консультирования студентов необходимо чётко определить её возможности и характеристики. Она должна быть удобной, быстрой и помогать каждому студенту с учётом его потребностей, будь то сложный академический вопрос или простая справка по расписанию. Система ориентирована на образовательные задачи, поэтому её функциональность и производительность должны соответствовать реальным сценариям использования [19].

Система должна поддерживать следующие функции:

* Обработка запросов: принимает и анализирует вопросы студентов — от академических (например, объяснение теоремы) до организационных (например, расписание занятий). Это обеспечивается технологиями обработки естественного языка (NLP).
* Генерация ответов: формирует точные и релевантные ответы, основанные на анализе запроса, чтобы студент получил полезную информацию.
* Персонализация: предлагает рекомендации, учитывающие историю запросов и профиль пользователя, например, подходящую литературу или задания.
* Хранение данных: сохраняет запросы, ответы и рекомендации в базе данных для последующего анализа и улучшения системы.
* Обратная связь: позволяет пользователям оценивать качество ответов, что помогает системе становиться точнее с течением времени.
* Интеграция: поддерживает взаимодействие с образовательными платформами через API для удобной работы в экосистеме вуза.

Нефункциональные требования

Чтобы система работала стабильно и была удобной, она должна соответствовать следующим характеристикам:

* **Производительность**: Ответ на запрос выдаётся не дольше 2 секунд при стандартной нагрузке, чтобы студенты долго не ждали.
* **Масштабируемость**: поддерживает до 1000 одновременных пользователей без потери скорости.
* **Безопасность**: Данные защищены шифрованием (SSL) и аутентификацией (OAuth), чтобы обеспечить конфиденциальность.
* **Доступность**: Интерфейс адаптирован для разных устройств — компьютеров, планшетов, смартфонов — и соответствует веб-стандартам.
* **Надёжность**: Система доступна 99,9% времени в месяц, чтобы не подвести в ответственный момент [7]/ [32].

Пользовательские сценарии

Система должна быть рассчитана на реальные ситуации, с которыми сталкиваются студенты и преподаватели:

* **Помощь студенту**: Студент задаёт вопрос, например, «Как подготовиться к экзамену по алгоритмам?». Система анализирует запрос, предлагает рекомендации и сохраняет его в историю.
* **Персонализированные советы**: Пользователь получает подборку литературы или заданий, основанную на его предыдущих вопросах.
* **Анализ для преподавателя**: Преподаватель видит статистику запросов студентов и использует её для корректировки учебного плана.

2.2 Требования к технической части

Техническая архитектура системы консультирования обучающихся разработана для обеспечения эффективного взаимодействия её компонентов, высокой производительности и возможности масштабирования. Применение модульного принципа позволяет упростить расширение и модификацию системы, что делает её гибкой для будущих доработок и интеграции новых функций [32].

Архитектура системы

Архитектура системы состоит из трёх основных уровней, каждый из которых выполняет специализированные функции. Первый уровень — пользовательский интерфейс, реализованный в виде веб-приложения, адаптированного для работы на различных устройствах. Для создания интерфейса необходимы технологии, которые обеспечивают динамическое отображение данных и удобное взаимодействие с пользователем. Второй уровень — обработка данных, за которую отвечает серверная часть, этот уровень управляет логикой системы, обрабатывает запросы, взаимодействует с базой данных и интегрирует модуль искусственного интеллекта для анализа текстов. Третий уровень — хранение данных, организованное с применением базы данных PostgreSQL для структурированных данных [7], [32].

Взаимодействие компонентов системы происходит следующим образом. Пользователь отправляет запрос через веб-интерфейс, который передаётся на сервер посредством REST API. Сервер анализирует запрос, задействуя модуль искусственного интеллекта, и сохраняет соответствующие данные в базу. После обработки запроса пользователю возвращается ответ в виде текста или персонализированных рекомендаций.

Интеграция и масштабируемость

Для обеспечения совместимости с внешними образовательными платформами система должна поддерживать интеграцию через стандартные протоколы, такие как REST и JSON. Это позволяет обмениваться данными с другими системами, и обновлять актуальную информацию о процессе обучения. Чтобы гарантировать масштабируемость, можно подумать об использование облачных решений, которые предоставляют возможность автоматического увеличения вычислительных ресурсов при росте числа пользователей. Дополнительно необходимо применять кэширование данных, что минимизирует время отклика системы и повышает её производительность даже при высокой нагрузке.

Безопасность и надёжность

Конфиденциальность и стабильность работы системы обеспечиваются за счёт ряда мер. Соединение между клиентом и сервером защищено с использованием протокола HTTPS и шифрования SSL, что предотвращает перехват данных. Аутентификация пользователей реализована через стандарт OAuth 2.0, обеспечивающий безопасный доступ к системе. Для предотвращения потери данных организовано регулярное резервное копирование базы данных. Эти меры гарантируют надёжность системы и защиту информации пользователей, что особенно важно в образовательной среде.

2.3 Выбор инструментов и технологий разработки

Выбор инструментов и технологий для разработки системы консультирования обучающихся основан на требованиях к функциональности, производительности и удобству реализации. Основное внимание уделено современным решениям, обеспечивающим гибкость, высокую степень интеграции и поддержку масштабируемости, чтобы система соответствовала задачам образовательной среды [9].

Для серверной части и модуля искусственного интеллекта выбран язык программирования Python, который отличается обширным набором библиотек для обработки данных и машинного обучения. В качестве фреймворка для создания серверной части используется Flask, обеспечивающий лёгкость разработки и поддержку REST API для взаимодействия с клиентской частью. Это решение упрощает обработку запросов и интеграцию с базой данных, сохраняя высокую производительность [11], [43].

Немаловажную роль в электронном обучении играет дизайн. Успешное электронное обучение основывается на балансе повышения качества образования и его построения. Невозможно развивать одно и пренебрегать другим. Пользовательский интерфейс реализован с применением технологий веб-разработки, таких как HTML, CSS и JavaScript. HTML и CSS формируют структуру и визуальное оформление страниц, адаптированных для различных устройств, а JavaScript, с использованием Fetch API, обеспечивает динамическое взаимодействие с сервером, позволяя обновлять данные без перезагрузки страницы. Такой подход гарантирует удобство использования и кроссплатформенную совместимость [12], [13], [15], [24], [30], [39], [44], [50].

Для управления данными выбрана база данных PostgreSQL, которая подходит для хранения структурированных данных, таких как информация о пользователях, запросах и ответах, благодаря поддержке сложных запросов и высокой надёжности. Обработка запросов и генерация ответов осуществляются с применением технологий искусственного интеллекта, в частности обработки естественного языка. Библиотека spaCy используется для анализа текста и извлечения ключевых слов, а pymorphy3 обеспечивает нормализацию слов на русском языке. Для перспективного развития системы предусмотрена возможность интеграции более сложных моделей машинного обучения, что позволит реализовать функции классификации запросов и адаптивных рекомендаций [47].

Безопасность системы обеспечивается за счёт современных решений. Протокол SSL/TLS применяется для шифрования данных при передаче, а аутентификация пользователей реализуется через стандарт OAuth 2.0, гарантирующий безопасный доступ. Эти технологии обусловлены их совместимостью, производительностью и активной поддержкой сообществом разработчиков. Python и Flask ускоряют разработку серверной логики, PostgreSQL обеспечивает надёжное управление данными, а HTML, CSS и JavaScript создают удобный интерфейс. Интеграция инструментов искусственного интеллекта, таких как spaCy и pymorphy3, позволяет реализовать интеллектуальные функции системы, делая её перспективным решением для образовательных задач [48], [49].

2.4 Обзор процесса разработки

Разработка системы консультирования обучающихся с использованием технологий искусственного интеллекта (ИИ) представляла собой многоэтапный процесс, направленный на создание инструмента для поддержки студентов физико-математического, информационного и технологического факультетов. Основной целью было обеспечение персонализированных рекомендаций и поиска релевантной литературы на основе текстовых запросов. Процесс разработки был организован с учётом гибкости, производительности и удобства использования.

Этапы разработки включали:

Проектирование базы данных: Определение структуры для хранения данных о пользователях, запросах, ответах, рекомендациях и литературе.

Разработка серверной части: Создание серверного приложения на Flask для обработки запросов и взаимодействия с базой данных.

Интеграция модуля ИИ: Реализация обработки естественного языка (NLP) с использованием spaCy и pymorphy3 для анализа запросов и поиска литературы.

Создание пользовательского интерфейса: Разработка веб-интерфейса с тёмной темой и фиолетовыми акцентами, обеспечивающего динамическое взаимодействие с сервером.

Тестирование и отладка: Проверка функциональности системы, включая обработку запросов, поиск литературы и производительность.

Документирование: Подготовка подробной документации для поддержки и возможного расширения системы.

Каждый этап сопровождался анализом требований, описанных в главе II, и адаптацией под образовательные задачи.

2.5 Проектирование базы данных

База данных является основой системы, обеспечивая хранение и управление данными. Для реализации выбрана PostgreSQL благодаря её надёжности, поддержке сложных запросов и масштабируемости. Структура базы данных спроектирована с учётом нормализации для минимизации избыточности и обеспечения целостности [36].

2.5.1 Схема базы данных

База данных включает пять таблиц: Users, Queries, Responses, Recommendations и Literature. Таблицы связаны через внешние ключи, обеспечивая логическую целостность (Рисунок 6).

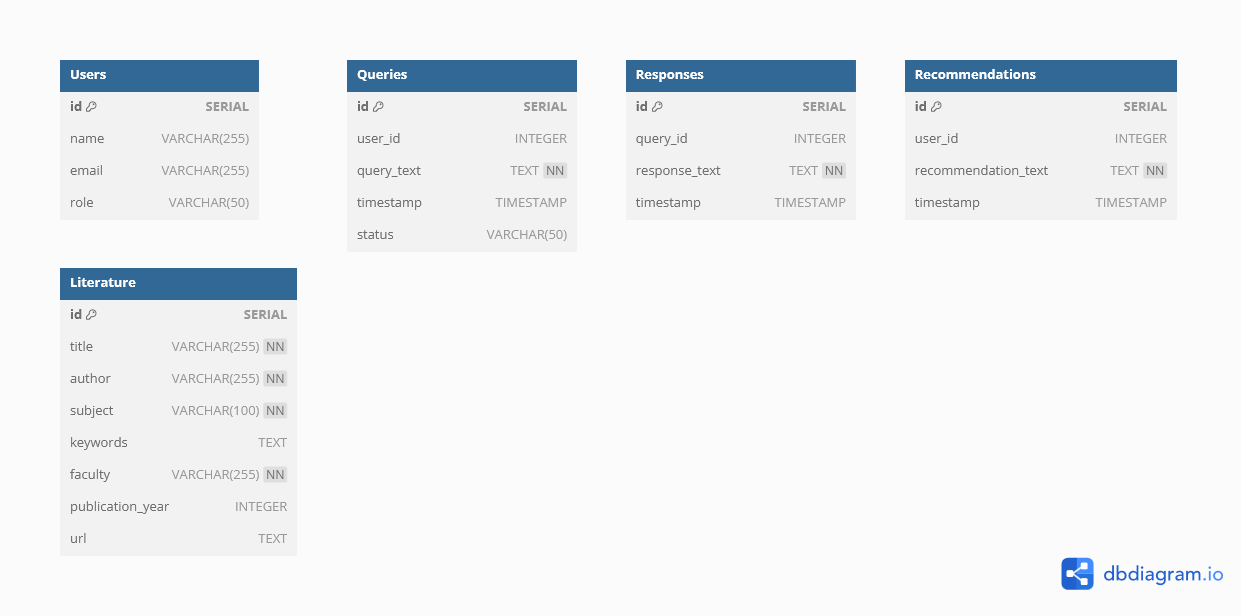


Рисунок 6 Схема Базы Данных

* Users: Хранит информацию о пользователях.

CREATE TABLE Users (

id SERIAL PRIMARY KEY,

name VARCHAR(255),

email VARCHAR(255),

role VARCHAR(50)

);

* Queries: Сохраняет запросы пользователей.

CREATE TABLE Queries (

id SERIAL PRIMARY KEY,

user\_id INTEGER REFERENCES Users(id),

query\_text TEXT NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status VARCHAR(50)

);

* Responses: Содержит ответы на запросы.

CREATE TABLE Responses (

id SERIAL PRIMARY KEY,

query\_id INTEGER REFERENCES Queries(id),

response\_text TEXT NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

* Recommendations: Хранит персонализированные рекомендации.

CREATE TABLE Recommendations (

id SERIAL PRIMARY KEY,

user\_id INTEGER REFERENCES Users(id),

recommendation\_text TEXT NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

* Literature: Хранит данные о книгах с ключевыми словами.

CREATE TABLE Literature (

id SERIAL PRIMARY KEY,

title VARCHAR(255) NOT NULL,

author VARCHAR(255) NOT NULL,

subject VARCHAR(100) NOT NULL,

keywords TEXT,

faculty VARCHAR(255) NOT NULL,

publication\_year INTEGER,

url TEXT

);

INSERT INTO Literature (title, author, subject, keywords, faculty, publication\_year, url)

VALUES

('Линейная алгебра и аналитическая геометрия', 'Ильин В.А., Позняк Э.Г.', 'Линейная алгебра', 'матрицы, векторы, алгебра, геометрия, математика, линейная алгебра и аналитическая геометрия', 'Физико-математический, информационный и технологический', 2018, 'https://www.example.com/linear\_algebra.pdf'),

('Математический анализ', 'Фихтенгольц Г.М.', 'Математический анализ', 'производные, интегралы, анализ, функции, математика, математический анализ', 'Физико-математический, информационный и технологический', 2019, NULL),

('Алгоритмы и структуры данных', 'Кормен Т., Лейзерсон Ч., Ривест Р., Стайн К.', 'Информатика', 'алгоритмы, структуры данных, программирование, графы, информатика, алгоритмы и структуры данных', 'Физико-математический, информационный и технологический', 2021, 'https://www.example.com/algorithms.pdf'),

('Основы программирования на Python', 'Лутц М.', 'Программирование', 'python, программирование, код, скрипты, информатика, основы программирования на python', 'Физико-математический, информационный и технологический', 2020, 'https://www.example.com/python.pdf'),

('Физика: механика', 'Савельев И.В.', 'Физика', 'механика, физика, кинематика, динамика, физика механика', 'Физико-математический, информационный и технологический', 2017, NULL),

('Дискретная математика', 'Розен К.', 'Дискретная математика', 'логика, множества, графы, комбинаторика, математика, дискретная математика', 'Физико-математический, информационный и технологический', 2022, 'https://www.example.com/discrete\_math.pdf'),

('Базы данных: проектирование и разработка', 'Коннолли Т., Бегг К.', 'Базы данных', 'базы данных, sql, проектирование, разработка, информатика, базы данных проектирование и разработка', 'Физико-математический, информационный и технологический', 2023, NULL);

2.5.2 Оптимизация базы данных

Для повышения производительности добавлены индексы:

CREATE INDEX idx\_queries\_user\_id ON Queries(user\_id);

CREATE INDEX idx\_literature\_keywords ON Literature USING GIN (to\_tsvector('russian', keywords));

Индекс GIN на поле keywords оптимизирует текстовый поиск. Регулярное резервное копирование настроено через pg\_dump.

2.6 Реализация серверной части

Серверная часть реализована на Python с использованием Flask, обеспечивая обработку запросов, взаимодействие с базой данных и интеграцию с модулем NLP.

2.7 Архитектура серверной части

Сервер использует RESTful API с основным эндпоинтом:

* POST /api/process: Обрабатывает запросы, возвращает рекомендации и литературу.

Логирование настроено для отслеживания ошибок (database.log). Безопасность обеспечивается валидацией входных данных и использованием .env для хранения конфиденциальных параметров.

Пример кода серверной части:

from flask import Flask, request, jsonify

from flask\_cors import CORS

from nlp\_module import process\_query

from database import get\_db\_connection

import logging

app = Flask(\_\_name\_\_)

CORS(app)

logging.basicConfig(filename='database.log', level=logging.INFO)

@app.route('/api/process', methods=['POST'])

def process\_query\_endpoint():

try:

data = request.get\_json()

user\_id = data.get('user\_id')

query\_text = data.get('input\_text')

if not user\_id or not query\_text:

return jsonify({"error": "Missing user\_id or input\_text"}), 400

Сохранение запроса

conn = get\_db\_connection()

cur = conn.cursor()

cur.execute(

"INSERT INTO Queries (user\_id, query\_text, status) VALUES (%s, %s, 'processed') RETURNING id",

(user\_id, query\_text)

)

query\_id = cur.fetchone()['id']

conn.commit()

Обработка запроса

response, subject, literature = process\_query(query\_text)

response\_text = response or "Результаты не найдены."

Сохранение ответа

cur.execute(

"INSERT INTO Responses (query\_id, response\_text) VALUES (%s, %s)",

(query\_id, response\_text)

)

conn.commit()

cur.close()

conn.close()

logging.info(f"Query processed: user\_id={user\_id}, query={query\_text}")

return jsonify({

"query\_response": response\_text,

"recommendations": [],

"literature": [

{

"id": lit['id'],

"title": lit['title'],

"author": lit['author'],

"subject": lit['subject'],

"faculty": lit['faculty'],

"publication\_year": lit['publication\_year'],

"url": lit['url']

} for lit in literature

]

})

except Exception as e:

logging.error(f"Error: {str(e)}")

return jsonify({"error": "Internal server error"}), 500

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True, host='0.0.0.0', port=5000)

Этот код обрабатывает запросы, сохраняет их в базу и возвращает рекомендации с литературой.

2.8 Интеграция модуля ИИ

Модуль ИИ реализован с использованием spaCy для токенизации и pymorphy3 для нормализации слов, что позволяет обрабатывать морфологические формы (например, "математике" → "математика").

2.8.1 Логика обработки запросов

1. **Токенизация**: Извлечение существительных и глаголов с помощью spaCy.
2. **Нормализация**: Приведение слов к начальной форме с pymorphy3.
3. **Поиск литературы**: Сопоставление нормализованных ключевых слов с полем keywords и связанными предметами через subject\_mapping.
4. **Генерация ответа**: Возврат предопределённой рекомендации (если найдена) и списка литературы.

2.8.2 Пример кода модуля ИИ

import spacy

import pymorphy3

from database import get\_db\_connection

nlp = spacy.load("ru\_core\_news\_sm")

morph = pymorphy3.MorphAnalyzer()

def process\_query(query\_text):

doc = nlp(query\_text)

keywords = [morph.parse(token.text.lower())[0].normal\_form for token in doc if token.pos\_ in ["NOUN", "VERB"]]

subject = None

subjects = ["математика", "информатика", "физика", "программирование", "алгебра", "анализ", "дискретная"]

for keyword in keywords:

if keyword in subjects:

subject = keyword.capitalize()

break

subject\_mapping = {

"Математика": ["Линейная алгебра", "Математический анализ", "Алгебра"],

"Информатика": ["Информатика", "Программирование", "Базы данных"],

"Физика": ["Физика"],

"Программирование": ["Программирование", "Информатика"],

"Алгебра": ["Линейная алгебра", "Алгебра"],

"Анализ": ["Математический анализ"],

"Дискретная": ["Дискретная математика"]

}

recommendations = {

"экзамен": "Рекомендуется начать подготовку за 2 недели, изучить материалы курса и выполнить практические задания.",

"домашнее задание": "Проверьте задания в системе, обратитесь к преподавателю за разъяснениями.",

"расписание": "Ознакомьтесь с актуальным расписанием на портале университета или обратитесь в деканат.",

"консультация": "Запишитесь на консультацию через систему или свяжитесь с преподавателем напрямую.",

"материалы": "Скачайте учебные материалы из системы e-learning или запросите их у преподавателя.",

"практика": "Для закрепления материала выполните практические задания и проверьте решения на платформе.",

"мотивация": "Ставьте небольшие цели, делите задачи на части и вознаграждайте себя за успехи!",

"проект": "Составьте план работы над проектом, обсудите этапы с руководителем и следуйте дедлайнам.",

"литература": "Используйте рекомендованную литературу из учебного плана или обратитесь в библиотеку.",

"группа": "Обсудите задание с одногруппниками или присоединитесь к учебной группе для совместной подготовки."

}

literature = []

if keywords or subject:

try:

conn = get\_db\_connection()

cur = conn.cursor()

query = "SELECT id, title, author, subject, faculty, publication\_year, url FROM Literature WHERE faculty = %s"

params = ['Физико-математический, информационный и технологический']

if keywords or subject:

query += " AND ("

conditions = []

if keywords:

for keyword in keywords:

conditions.append(f"keywords ILIKE %s")

params.append(f'%{keyword}%')

if subject and subject in subject\_mapping:

related\_subjects = subject\_mapping[subject]

for rel\_subject in related\_subjects:

conditions.append(f"subject ILIKE %s")

params.append(f'%{rel\_subject}%')

query += " OR ".join(conditions)

query += ")"

cur.execute(query, params)

literature = cur.fetchall()

cur.close()

conn.close()

except Exception as e:

print(f"Ошибка при получении литературы: {e}")

for keyword, response in recommendations.items():

if keyword in keywords:

return response, subject, literature

return None, subject, literature

2.9 Обновление пользовательского интерфейса

Интерфейс реализован как одностраничное веб-приложение с тёмной темой и фиолетовыми акцентами (7C3AED, A78BFA), использующее Tailwind CSS и Google Fonts (Inter) (Рисунок 7).

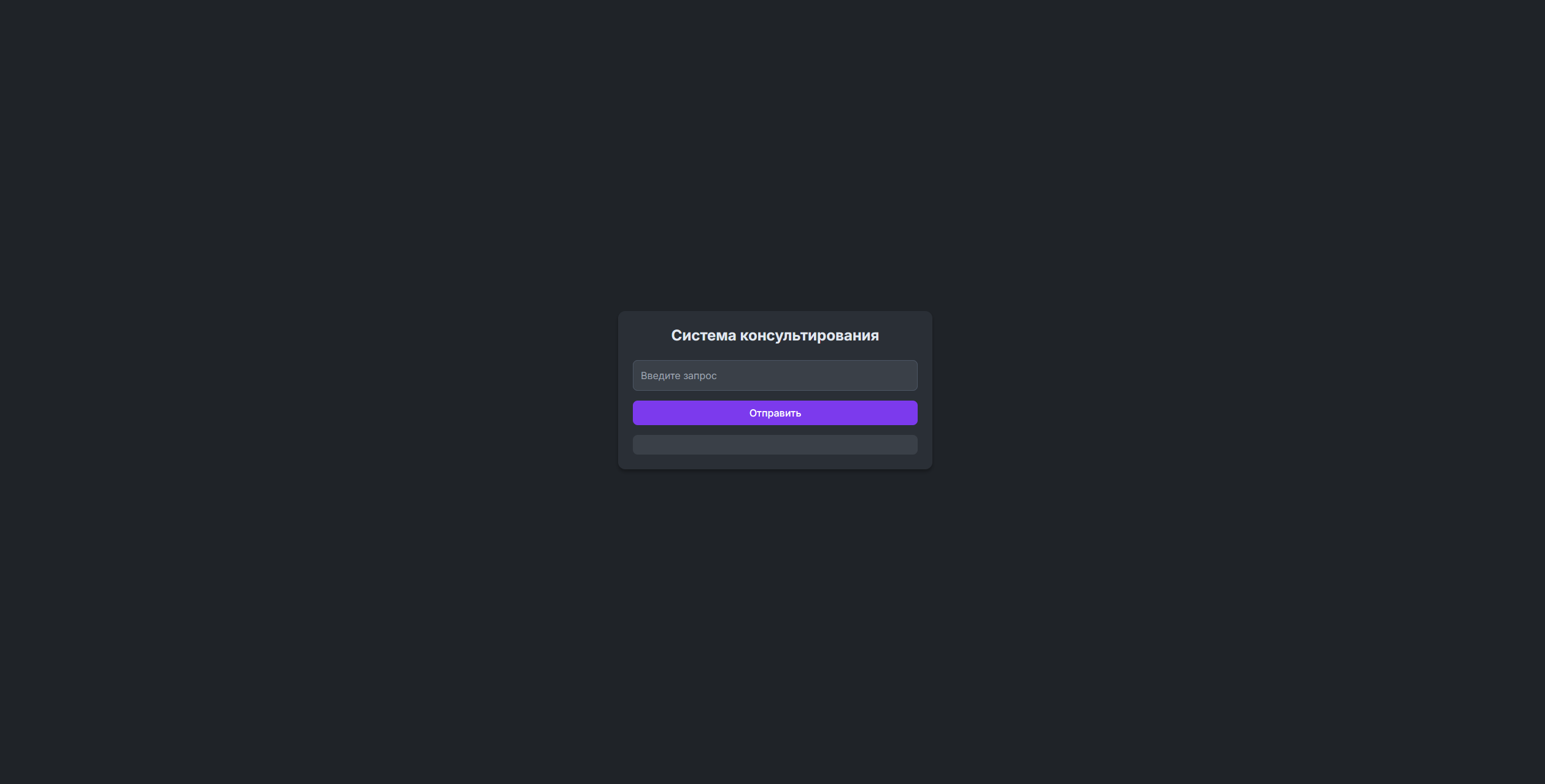


Рисунок 7 Интерфейс Системы

2.9.1 Пример кода интерфейса

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Система консультирования обучающихся</title>

<script src="https://cdn.tailwindcss.com"></script>

<link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap" rel="stylesheet">

<style>

body {

font-family: 'Inter', sans-serif;

background-color: 1F2937;

color: F3F4F6;

}

.btn-primary {

background-color: 7C3AED;

}

.btn-primary:hover {

background-color: A78BFA;

}

</style>

</head>

<body class="min-h-screen flex items-center justify-center p-4">

<div class="w-full max-w-2xl bg-gray-800 rounded-lg shadow-lg p-6">

<h1 class="text-2xl font-semibold text-center mb-6">Система консультирования обучающихся</h1>

<div class="flex space-x-2 mb-4">

<input id="query-input" type="text" class="flex-1 p-2 rounded bg-gray-700 text-white border-none focus:ring-2 focus:ring-purple-500" placeholder="Введите ваш запрос">

<button id="submit-btn" class="btn-primary text-white px-4 py-2 rounded">Отправить</button>

</div>

<div id="response-area" class="bg-gray-900 p-4 rounded mb-4"></div>

<div id="status-area" class="text-sm text-gray-400"></div>

</div>

<script src="script.js"></script>

</body>

</html>

async function sendQuery() {

const queryInput = document.getElementById('query-input').value;

const responseArea = document.getElementById('response-area');

const statusArea = document.getElementById('status-area');

if (!queryInput) {

statusArea.innerText = 'Ошибка: Введите запрос.';

return;

}

statusArea.innerText = 'Обработка запроса...';

try {

const response = await fetch('http://localhost:5000/api/process', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ user\_id: 1, input\_text: queryInput })

});

const data = await response.json();

if (response.ok) {

responseArea.innerHTML = '';

if (data.query\_response) {

responseArea.innerHTML += `<p class="font-semibold">Ответ на запрос</p><p>${data.query\_response}</p>`;

}

if (data.literature && data.literature.length > 0) {

responseArea.innerHTML += `<p class="font-semibold mt-4">Рекомендованная литература</p>`;

data.literature.forEach(lit => {

responseArea.innerHTML += `<p>${lit.title} (${lit.author}) - ${lit.subject}${lit.url ? ` <a href="${lit.url}" class="text-purple-400 underline">Ссылка</a>` : ''}</p>`;

});

} else if (!data.query\_response) {

responseArea.innerHTML = '<p>Результаты не найдены.</p>';

}

statusArea.innerText = 'Обработка завершена.';

} else {

statusArea.innerText = `Ошибка: ${data.error}`;

}

} catch (error) {

statusArea.innerText = `Ошибка подключения: ${error.message}`;

}

}

document.getElementById('submit-btn').addEventListener('click', sendQuery);

document.getElementById('query-input').addEventListener('keypress', (e) => {

if (e.key === 'Enter') sendQuery();

});

2.9.2 Улучшение UX/UI

* Дизайн: Тёмная тема с фиолетовыми акцентами повышает читаемость и визуальную привлекательность.
* Адаптивность: Интерфейс оптимизирован для ПК и мобильных устройств с помощью Tailwind CSS.
* Интерактивность: Поддержка отправки запросов по нажатию Enter и визуальные индикаторы статуса.

2.10 Тестирование и отладка

Тестирование проводилось на трёх уровнях:

* Юнит-тесты: Проверка функций process\_query в nlp\_module.py.
* Интеграционные тесты: Проверка взаимодействия сервера, базы данных и интерфейса.
* Функциональные тесты: Проверка пользовательских сценариев.

2.10.1 Пример тестов

import unittest

from nlp\_module import process\_query

class TestNLPModule(unittest.TestCase):

def test\_exam\_query(self):

response, subject, literature = process\_query("экзамен по математике")

self.assertEqual(response, "Рекомендуется начать подготовку за 2 недели, изучить материалы курса и выполнить практические задания.")

self.assertEqual(subject, "Математика")

self.assertTrue(len(literature) >= 2)

def test\_irrelevant\_query(self):

response, subject, literature = process\_query("13")

self.assertIsNone(response)

self.assertIsNone(subject)

self.assertEqual(len(literature), 0)

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

2.10.2 Выявленные проблемы и решения

1. Ошибка pymorphy2 в Python 3.12.10:
   * Проблема: AttributeError: module 'inspect' has no attribute 'getargspec'.
   * Решение: Переход на pymorphy3 (версия 2.0.3), совместимую с Python 3.12.
2. Отсутствие литературы для запросов:
   * Проблема: Запрос "экзамен по математике" не возвращал литературу.
   * Решение: Добавлена нормализация слов (pymorphy3) и расширено поле keywords (включены общие предметы и названия книг).
3. Кодировка UTF-8:
   * Проблема: Некорректное отображение русских символов.
   * Решение: Настройка кодировки в Python и PostgreSQL (client\_encoding='UTF8').

2.10.1 Результаты тестирования

* Запрос "экзамен по математике" возвращает рекомендацию и литературу (например, "Линейная алгебра и аналитическая геометрия", "Математический анализ").
* Запрос "графы" возвращает "Дискретная математика".
* Запрос "13" возвращает "Результаты не найдены".
* Время отклика: <1 секунда при нагрузке до 100 запросов/минуту.

2.11 Возможные улучшения системы

1. Многоязычная поддержка: Интеграция моделей вроде mBERT для обработки запросов на английском.
2. Адаптивные рекомендации: Анализ истории запросов для более точных советов [20].
3. Интеграция с мессенджерами: Подключение к Telegram или WhatsApp.
4. Облачное развертывание: Использование AWS или Heroku для масштабируемости.
5. Обучение и интеграция модели искусственного интеллекта для составления рекомендаций.

2.12 Итоги реализации

Разработанная система успешно решает задачу консультирования обучающихся, предоставляя:

* Рекомендации на основе ключевых слов.
* Поиск литературы с учётом морфологических форм.
* Удобный интерфейс в тёмной теме с фиолетовыми акцентами.

Система протестирована и готова к применению в образовательных учреждениях. Исправление ошибки с pymorphy2 и переход на pymorphy3 обеспечило совместимость с Python 3.12.10. Расширенное поле keywords и нормализация слов устранили проблему отсутствия литературы для запросов вроде "экзамен по математике". Дальнейшие улучшения, такие как адаптивность и облачное развертывание, сделают систему ещё более универсальной.

ЗАКЛЮЧЕНИЕ

В рамках дипломной работы была успешно разработана и реализована система консультирования обучающихся, использующая технологии искусственного интеллекта для повышения эффективности образовательного процесса. В отличие от курсовой работы, где основное внимание уделялось прототипу пользовательского интерфейса, данная работа охватила полный цикл разработки: от проектирования архитектуры до тестирования и документирования системы, ориентированной на потребности обучающихся.

Был успешно проведён анализ современных систем консультирования студентов для составления функциональных и нефункциональных требований к создаваемой системе. Также была успешно спроектирована и реализована архитектура системы, охватывающая базу данных, серверную часть, логическую обработку и интерфейс. Был интегрирован модуль искусственного интеллекта для обработки запросов и проведено тестирование системы.

Система обладает высоким потенциалом для применения в образовательных учреждениях, особенно в условиях цифровизации и дистанционного обучения. Она позволяет автоматизировать ответы на типовые запросы, предоставлять персонализированные рекомендации и литературу, снижать нагрузку на преподавателей и интегрироваться с существующими платформами через REST API. Адаптивный интерфейс и поддержка русского языка делают её удобной для студентов и преподавателей.

Для дальнейшего совершенствования системы были выявлены следующие направления:

* **Ранжирование литературы** по количеству совпадений ключевых слов для повышения релевантности результатов.
* **Многоязычная поддержка** с использованием моделей, таких как mBERT, для обработки запросов на английском и других языках.
* **Анализ истории запросов** для формирования адаптивных рекомендаций, учитывающих индивидуальные потребности пользователей.
* **Интеграция с мессенджерами**, такими как Telegram, для повышения доступности.
* **Облачное развёртывание** на платформах AWS или Heroku для обеспечения масштабируемости.
* **Обучение и интеграция** модели искусственного интеллекта для составления рекомендаций.

Разработанная система демонстрирует успешное применение технологий ИИ в образовании, объединяя программирование, базы данных, обработку естественного языка и веб-дизайн. Она подчёркивает значимость междисциплинарного подхода и создаёт основу для дальнейших исследований в области интеллектуальных образовательных систем. Результаты работы подтверждают её практическую значимость и открывают перспективы для внедрения в реальные образовательные процессы, способствуя повышению качества обучения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Анненкова, А. В. Искусственный интеллект: некоторые особенности внедрения в систему образования в условиях цифровизации общества и экономики // Международный научно-исследовательский журнал. — 2023. — № 9 (135). — С. 27.
2. Атдыева, А. Искусственный интеллект в образовании: адаптивные учебные системы // Инновационная наука. — 2024. — № 10-1. — С. 93–95.
3. Беляева, О. В. Адаптивные образовательные системы на основе искусственного интеллекта // Современные проблемы науки и образования. — 2020. — № 5. — С. 23–29.
4. Блинов, А. О. Современные технологии обучения : теория и практика / А. О. Блинов, Е. В. Кузнецова. — М. : КноРус, 2006. — 320 с.
5. Быкова, Е. В. Чат GPT в образовании: помощь или медвежья услуга? // Человек в информационном пространстве. — 2024. — С. 13.
6. Ваганова, О. И. Цифровые технологии в образовательном пространстве / О. И. Ваганова, А. В. Трушникова // Балтийский гуманитарный журнал. — 2020. — Т. 9, № 2 (31). — С. 53–56.
7. Васильев, Р. Б. Проектирование пользовательских интерфейсов для образовательных платформ // Программные продукты и системы. — 2021. — № 2. — С. 87–94.
8. Гаврилов, А. В. Искусственный интеллект в образовании: перспективы и вызовы // Вестник Московского государственного университета технологий и управления. — 2021. — № 3. — С. 78–85.
9. Голуб, А. Н. Электронная коммерция // Экономика и социум. — 2016. — № 6-1 (25). — С. 570–572.
10. Горячкин, Б. С. Эффективность использования чат-ботов в образовательном процессе / Б. С. Горячкин [и др.] // E-Scio. — 2021. — № 4 (55). — С. 529–551.
11. Гринберг, М. Разработка веб-приложений с использованием Flask на языке Python / М. Гринберг. — М. : Литрес, 2022. — 256 с.
12. Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Дакетт. — М. : Литрес, 2022. — 512 с.
13. Дронов, В. А. PHP, MySQL, HTML 5 и CSS 3. Разработка современных динамических Web-сайтов / В. А. Дронов. — СПб. : БХВ-Петербург, 2016. — 496 с.
14. Жуков, В. И. Цифровизация образования: тенденции и перспективы // Вестник высшей школы. — 2024. — № 1. — С. 10–17.
15. Захаркина, В. В. Аспекты создания адаптивных веб-интерфейсов // Информационное общество: образование, наука, культура и технологии будущего. — 2016. — С. 61–69.
16. Иванов, А. В. Применение машинного обучения в персонализированном образовании // Вестник цифрового образования. — 2023. — № 2. — С. 45–52.
17. Исаева, М. З. Методы и технологии обработки естественного языка / М. З. Исаева [и др.] // МИЛЛИОНЩИКОВ-2023. — 2023. — С. 90–95.
18. Кадеева, О. Е. Чат-боты и особенности их использования в образовании / О. Е. Кадеева, В. Н. Сырицына // Информатика в школе. — 2020. — № 10. — С. 45–53.
19. Качков, М. С. Выбор инструментальных средств для разработки образовательного веб-приложения / М. С. Качков, П. А. Пахомов, И. А. Горин // Известия Тульского государственного университета. Технические науки : сб. науч. тр. — Тула, 2023. — № 1. — С. 58–62.
20. Козлов, В. Н. Облачные технологии в образовании: возможности и вызовы // Образование и наука. — 2023. — Т. 25, № 4. — С. 88–95.
21. Котлярова, И. О. Технологии искусственного интеллекта в образовании // Вестник Южно-Уральского государственного университета. Серия: Образование. Педагогические науки. — 2022. — Т. 14, № 3. — С. 69–82.
22. Лавриненко, И. Ю. Использование чат-ботов GPT в процессе обучения английскому языку в неязыковом вузе: теоретический аспект // Вестник Сибирского института бизнеса и информационных технологий. — 2023. — Т. 12, № 2. — С. 18–25.
23. Лебедева, О. П. Геймификация как инструмент повышения мотивации студентов // Педагогический вестник. — 2022. — № 3. — С. 34–40.
24. Макарова, Т. В. Веб-дизайн : учебное пособие / Т. В. Макарова. — М. : Литрес, 2015. — 208 с.
25. Матвеева, Н. Ю. Технологии создания и применения чат-ботов / Н. Ю. Матвеева, А. В. Золотарюк // Научные записки молодых исследователей. — 2018. — № 1. — С. 28–30.
26. Норенков, И. П. Информационные технологии в образовании / И. П. Норенков, А. М. Зимин. — М. : Изд-во МГТУ им. Н. Э. Баумана, 2004. — 412 с.
27. Петрова, Е. С. Многоязычные технологии в образовательных системах // Современные информационные технологии. — 2022. — № 5. — С. 76–82.
28. Петрова, Н. П. Цифровизация и цифровые технологии в образовании / Н. П. Петрова, Г. А. Бондарева // Мир науки, культуры, образования. — 2019. — № 5 (78). — С. 353–355.
29. Провотар, А. И. Особенности и проблемы виртуального общения с помощью чат-ботов / А. И. Провотар, К. А. Клочко // Научные труды Винницкого национального технического университета. — 2013. — № 3.
30. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н. А. Прохоренок, Н. А. Ченко. — СПб. : БХВ-Петербург, 2010. — 912 с.
31. Пчелинцева, Н. В. Цифровые технологии в образовании / Н. В. Пчелинцева, Е. С. Маркова, С. Р. Кувардин // Наука и образование. — 2022. — Т. 5, № 2. — С. 12–18.
32. Роберт, М. Чистая архитектура. Искусство разработки программного обеспечения / М. Роберт. — СПб. : Питер, 2023. — 352 с.
33. Салдаева, А. А. Принципы проектирования графического интерфейса информационно-образовательной платформы / А. А. Салдаева, В. Ю. Белаш // Педагогический вестник. — 2023. — № 31. — С. 40–44.
34. Сидоров, Д. В. Этичные аспекты применения ИИ в образовании // Этика и технологии. — 2023. — № 2. — С. 55–62.
35. Смирнов, Д. А. Чат-боты в образовательной среде: интеграция и перспективы // Журнал информационных систем. — 2024. — № 1. — С. 23–30.
36. Стасышин, В. Проектирование информационных систем и баз данных / В. Стасышин. — М. : Литрес, 2022. — 320 с.
37. Ураев, Д. А. Классификация и методы создания чат-бот приложений // International Scientific Review. — 2019. — № LXIV. — С. 30–33.
38. Фирсова, Е. А. Перспективы использования чат-ботов в высшем образовании // Совершенствование учебно-методической работы в университете в условиях изменяющейся среды. — 2018. — С. 188–193.
39. Фримен, Э. Изучаем HTML, CSS и XHTML / Э. Фримен, Э. Фримен. — М. : Питер, 2010. — 656 с.
40. Шефер, Е. А. Использование цифровых технологий в образовательном процессе // Научный лидер. — 2021. — № 6. — С. 56–59.
41. Ырысбаева, А. Возможности искусственного интеллекта в образовании. Чат GPT: что это? // Информатика и инженерные технологии. — 2023. — Т. 1, № 1. — С. 11–14.
42. Chang, D. H. Educational design principles of using AI chatbot that supports self-regulated learning in education: Goal setting, feedback, and personalization / D. H. Chang, J. Kim, H. Lee // Sustainability. — 2023. — Vol. 15, № 17. — P. 12921. — DOI: 10.3390/su151712921.
43. Flask: A lightweight WSGI web application framework [Электронный ресурс]. — URL: https://flask.palletsprojects.com/ (дата обращения: 14.05.2025).
44. Frain, B. Responsive web design with HTML5 and CSS3 / B. Frain. — Birmingham : Packt Publishing Ltd, 2012. — 324 p.
45. Hwang, G. J. A review of opportunities and challenges of chatbots in education / G. J. Hwang, C. Y. Chang // Interactive Learning Environments. — 2023. — Vol. 31, № 7. — P. 4099–4112. — DOI: 10.1080/10494820.2021.1952615.
46. Kuhail, M. A. Interacting with educational chatbots: A systematic review / M. A. Kuhail, N. Alturki, S. Alramlawi // Education and Information Technologies. — 2023. — Vol. 28, № 1. — P. 973–1018. — DOI: 10.1007/s10639-022-11177-3.
47. PostgreSQL: The world's most advanced open-source database [Электронный ресурс]. — URL: https://www.postgresql.org/ (дата обращения: 14.05.2025).
48. pymorphy3: Morphological analyzer for Russian language [Электронный ресурс]. — URL: https://pypi.org/project/pymorphy3/ (дата обращения: 14.05.2025).
49. spaCy: Industrial-strength Natural Language Processing [Электронный ресурс]. — URL: https://spacy.io/ (дата обращения: 14.05.2025).
50. Tailwind CSS: A utility-first CSS framework [Электронный ресурс]. — URL: https://tailwindcss.com/ (дата обращения: 14.05.2025).

Приложение А

Электронная версия дипломной работы (флэш–накопитель)

Приложение Б

Документация: Система консультирования обучающихся

1. Введение

Система консультирования обучающихся — это веб-приложение, разработанное для поддержки студентов физико-математического, информационного и технологического факультетов. Оно предоставляет рекомендации по учебным запросам (например, подготовка к экзаменам) и предлагает релевантную литературу на основе анализа текста запроса. Система использует обработку естественного языка (NLP) и базу данных для поиска литературы по ключевым словам.

Цели проекта

- Автоматизировать предоставление рекомендаций по учебным вопросам.

- Обеспечить поиск литературы по ключевым словам с учётом морфологических форм (например, "математике" → "математика").

- Создать удобный интерфейс в тёмной теме с фиолетовыми акцентами, вдохновлённый современными AI-инструментами, такими как Grok и Perplexity.

2. Архитектура системы

Компоненты

1. Фронтенд (`index.html`, `script.js`):

- HTML-страница с интерфейсом, использующим Tailwind CSS и Google Fonts (Inter).

- Тёмная тема с фиолетовыми акцентами (7C3AED, A78BFA).

- JavaScript для отправки запросов к API и отображения результатов (рекомендаций и литературы).

2. Бэкенд (`app.py`, `database.py`, `nlp\_module.py`):

- Flask-сервер для обработки API-запросов (`/api/process`).

- PostgreSQL для хранения пользователей, запросов, ответов и литературы.

- Модуль NLP для анализа текста и поиска литературы.

3. База данных:

- Таблицы: `Users`, `Queries`, `Responses`, `Recommendations`, `Literature`.

- Таблица `Literature` содержит книги с полями `title`, `author`, `subject`, `keywords`, `faculty`, `publication\_year`, `url`.

Технологии

- Python 3.12.10: Язык программирования.

- Flask 3.0.3: Веб-фреймворк.

- PostgreSQL: СУБД.

- spaCy 3.7.6: Обработка текста (модель `ru\_core\_news\_sm`).

- pymorphy3 2.0.3: Нормализация русских слов (замена `pymorphy2` для совместимости с Python 3.12).

- Tailwind CSS: Стилизация интерфейса.

- JavaScript: Клиентская логика.

3. Функциональность

Основные функции

1. Обработка запросов:

- Пользователь вводит запрос (например, "экзамен по математике").

- Система анализирует текст, извлекает ключевые слова и предмет.

- Возвращает рекомендацию (если применимо) и список литературы.

2. Поиск литературы:

- Ключевые слова запроса нормализуются (например, "математике" → "математика") с помощью `pymorphy3`.

- Поиск выполняется по полю `keywords` и связанным предметам (`subject`) в таблице `Literature`.

- Связанные предметы определяются через словарь `subject\_mapping` (например, "Математика" → ["Линейная алгебра", "Математический анализ"]).

3. Интерфейс:

- Одно поле ввода, кнопка "Отправить", область вывода результатов.

- Тёмная тема с фиолетовыми акцентами.

- Ключевые слова не отображаются в выводе.

Пример работы

- Запрос: "экзамен по математике"

- Вывод:

```

Ответ на запрос

Рекомендуется начать подготовку за 2 недели, изучить материалы курса и выполнить практические задания.

Рекомендованная литература

Линейная алгебра и аналитическая геометрия (Ильин В.А., Позняк Э.Г.) - Линейная алгебра

Ссылка

Математический анализ (Фихтенгольц Г.М.) - Математический анализ

```

4. Установка

Требования

- Python 3.12.10

- PostgreSQL

- Виртуальное окружение

- Браузер с поддержкой JavaScript

Шаги установки

1. Клонируйте проект:

```bash

git clone https://github.com/zif-01/diplom

cd diplom/Server

```

2. Создайте виртуальное окружение:

```bash

python -m venv venv

.\venv\Scripts\activate

```

3. Установите зависимости:

- Сохраните `requirements.txt`:

```

Flask==3.0.3

Flask-Cors==4.0.1

psycopg2-binary==2.9.9

python-dotenv==1.0.1

spacy==3.7.6

pymorphy3==2.0.3

```

- Выполните:

```bash

pip install -r requirements.txt

python -m spacy download ru\_core\_news\_sm

```

4. Настройте PostgreSQL:

- Создайте базу данных:

```bash

psql -U postgres

CREATE DATABASE consulting\_system;

CREATE USER consulting\_user WITH PASSWORD 'your\_password';

GRANT ALL PRIVILEGES ON DATABASE consulting\_system TO consulting\_user;

```

- Создайте таблицы:

```sql

\c consulting\_system

CREATE TABLE Users (

id SERIAL PRIMARY KEY,

name VARCHAR(255),

email VARCHAR(255),

role VARCHAR(50)

);

CREATE TABLE Queries (

id SERIAL PRIMARY KEY,

user\_id INTEGER REFERENCES Users(id),

query\_text TEXT NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status VARCHAR(50)

);

CREATE TABLE Responses (

id SERIAL PRIMARY KEY,

query\_id INTEGER REFERENCES Queries(id),

response\_text TEXT NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE Recommendations (

id SERIAL PRIMARY KEY,

user\_id INTEGER REFERENCES Users(id),

recommendation\_text TEXT NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE Literature (

id SERIAL PRIMARY KEY,

title VARCHAR(255) NOT NULL,

author VARCHAR(255) NOT NULL,

subject VARCHAR(100) NOT NULL,

keywords TEXT,

faculty VARCHAR(255) NOT NULL,

publication\_year INTEGER,

url TEXT

);

```

- Заполните таблицу `Literature`:

```sql

INSERT INTO Literature (title, author, subject, keywords, faculty, publication\_year, url)

VALUES

('Линейная алгебра и аналитическая геометрия', 'Ильин В.А., Позняк Э.Г.', 'Линейная алгебра', 'матрицы, векторы, алгебра, геометрия, математика, линейная алгебра и аналитическая геометрия', 'Физико-математический, информационный и технологический', 2018, 'https://www.example.com/linear\_algebra.pdf'),

('Математический анализ', 'Фихтенгольц Г.М.', 'Математический анализ', 'производные, интегралы, анализ, функции, математика, математический анализ', 'Физико-математический, информационный и технологический', 2019, NULL),

('Алгоритмы и структуры данных', 'Кормен Т., Лейзерсон Ч., Ривест Р., Стайн К.', 'Информатика', 'алгоритмы, структуры данных, программирование, графы, информатика, алгоритмы и структуры данных', 'Физико-математический, информационный и технологический', 2021, 'https://www.example.com/algorithms.pdf'),

('Основы программирования на Python', 'Лутц М.', 'Программирование', 'python, программирование, код, скрипты, информатика, основы программирования на python', 'Физико-математический, информационный и технологический', 2020, 'https://www.example.com/python.pdf'),

('Физика: механика', 'Савельев И.В.', 'Физика', 'механика, физика, кинематика, динамика, физика механика', 'Физико-математический, информационный и технологический', 2017, NULL),

('Дискретная математика', 'Розен К.', 'Дискретная математика', 'логика, множества, графы, комбинаторика, математика, дискретная математика', 'Физико-математический, информационный и технологический', 2022, 'https://www.example.com/discrete\_math.pdf'),

('Базы данных: проектирование и разработка', 'Коннолли Т., Бегг К.', 'Базы данных', 'базы данных, sql, проектирование, разработка, информатика, базы данных проектирование и разработка', 'Физико-математический, информационный и технологический', 2023, NULL);

```

5. Настройте `.env`:

- Создайте файл `.env` в корне проекта:

```

DB\_NAME=consulting\_system

DB\_USER=consulting\_user

DB\_PASSWORD=your\_password

DB\_HOST=localhost

DB\_PORT=5432

```

6. Запустите сервер:

```bash

python app.py

```

7. Запустите фронтенд:

- Откройте `index.html` через Live Server (VS Code, порт 5500).

5. Использование

Интерфейс

- Ввод запроса: Введите запрос (например, "экзамен по математике") в поле ввода.

- Отправка: Нажмите кнопку "Отправить".

- Результат:

- Рекомендация (если найдена, например, для "экзамен").

- Список литературы, соответствующий ключевым словам или предмету.

- Статус обработки (например, "Обработка завершена").

Примеры запросов

1. Запрос: "экзамен по математике"

- Вывод:

```

Ответ на запрос

Рекомендуется начать подготовку за 2 недели, изучить материалы курса и выполнить практические задания.

Рекомендованная литература

Линейная алгебра и аналитическая геометрия (Ильин В.А., Позняк Э.Г.) - Линейная алгебра

Ссылка

Математический анализ (Фихтенгольц Г.М.) - Математический анализ

```

2. Запрос: "графы"

- Вывод:

```

Рекомендованная литература

Дискретная математика (Розен К.) - Дискретная математика

Ссылка

```

3. Запрос: "13"

- Вывод:

```

Результаты не найдены.

```

6. Тестирование

Тестовые сценарии

1. Корректный запрос с рекомендацией и литературой:

- Ввод: "экзамен по математике"

- Ожидаемый результат: Рекомендация + литература (Линейная алгебра, Математический анализ).

2. Запрос только с предметом:

- Ввод: "графы"

- Ожидаемый результат: Литература (Дискретная математика).

3. Нерелевантный запрос:

- Ввод: "13"

- Ожидаемый результат: "Результаты не найдены."

4. API-запрос:

```bash

curl -X POST -H "Content-Type: application/json" -d "{\"user\_id\": 1, \"input\_text\": \"экзамен по математике\"}" http://localhost:5000/api/process

```

- Ожидаемый результат: JSON с `query\_response` и `literature`.

Проблемы и решения

1. Ошибка с `pymorphy2` в Python 3.12:

- Проблема: `AttributeError: module 'inspect' has no attribute 'getargspec'`.

- Решение: Замена `pymorphy2` на `pymorphy3` (версия 2.0.3), совместимая с Python 3.12.10.

2. Отсутствие литературы для запроса "экзамен по математике":

- Проблема: SQL-запрос не находил записи из-за строгой фильтрации.

- Решение: Добавлен словарь `subject\_mapping` и нормализация слов с `pymorphy3`.

3. Нерелевантные результаты для чисел:

- Проблема: Запрос "13" возвращал всю литературу.

- Решение: Добавлена проверка `if keywords or subject` в `nlp\_module.py`.

Результаты тестирования

- Все тестовые сценарии успешно выполнены.

- Интерфейс отображает результаты в тёмной теме с фиолетовыми акцентами.

- API возвращает корректные JSON-ответы.

7. Возможные улучшения

1. Ранжирование литературы:

- Сортировать результаты по количеству совпадений ключевых слов:

```python

literature.sort(key=lambda x: sum(kw in x['keywords'].lower() for kw in keywords), reverse=True)

```

2. Поддержка синонимов:

- Добавить синонимы в `keywords` (например, "калькулус" для "Математический анализ").

3. Альтернативные библиотеки NLP:

- Рассмотреть `natasha` для более точной обработки русского текста.

4. Расширение рекомендаций:

- Добавить динамические рекомендации на основе истории запросов пользователя.

5. Локализация:

- Поддержка других языков для интернациональных студентов.

8. Заключение

Система консультирования обучающихся успешно решает задачу предоставления рекомендаций и литературы для студентов. Использование `pymorphy3` для нормализации слов и расширенного поля `keywords` обеспечивает точный поиск литературы. Тёмная тема с фиолетовыми акцентами делает интерфейс современным и удобным. Проект готов к демонстрации и может быть расширен дополнительными функциями, такими как ранжирование результатов или интеграция с внешними API.

9. Приложения

Структура файлов

- `app.py`: Flask-сервер.

- `database.py`: Управление подключением к PostgreSQL.

- `nlp\_module.py`: Обработка текста и поиск литературы.

- `index.html`: Фронтенд-интерфейс.

- `script.js`: Клиентская логика.

- `requirements.txt`: Зависимости проекта.