

Д.С. Найманова^{1*} , А.З. Даутова¹ , Ж.С. Алимова¹ , Н.Н. Оспанова¹ 

¹ Торайгыров университет, г.Павлодар, Қазақстан

*e-mail: dinarasakenkyzy83@gmail.com

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ АНАЛИТИЧЕСКОЙ СИСТЕМЫ ДЛЯ АНАЛИЗА ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ НА БАЗЕ LLM

Аннотация

В статье рассматривается разработка интеллектуальной аналитической системы для анализа образовательных программ, предназначенной для автоматизации оценки содержания образовательных документов с использованием технологий искусственного интеллекта. В отличие от существующих решений, ориентированных преимущественно на локальную обработку отдельных документов, предложенная система реализована в виде веб-приложения с клиент-серверной архитектурой, включающей frontend, backend, реляционную базу данных, модуль обработки PDF-документов и модуль искусственного интеллекта. Представлена логика функционирования системы, описывающая взаимодействие пользователей различных ролей (преподавателя, студента и администратора), организацию хранения данных и маршрут их обработки внутри системы. Для интеллектуального анализа содержания образовательных программ используются методы обработки естественного языка (NLP) и большие языковые модели (LLM), обеспечивающие автоматическое выявление универсальных компетенций (soft skills) и формирование рекомендаций по совершенствованию образовательных программ. Результаты анализа сохраняются в базе данных и предоставляются пользователям через веб-интерфейс. Разработанная интеллектуальная аналитическая система обеспечивает автоматизацию анализа образовательных программ, сокращает трудоемкость экспертной оценки, повышает объективность анализа и может быть использована образовательными организациями при мониторинге и совершенствовании образовательных программ в соответствии с современными требованиями рынка труда.

Ключевые слова: анализ образовательных программ; искусственный интеллект; обработка естественного языка (NLP); большие языковые модели (LLM); автоматизация анализа документов; клиент-серверная архитектура; реляционная база данных; образовательные технологии; цифровизация образования.

Д.С. Найманова¹, А.З. Даутова¹, Ж.С. Алимова¹, Н.Н. Оспанова¹

¹ Торайгыров университеті, Павлодар қ., Қазақстан.

БІЛІМ БЕРУ БАҒДАРЛАМАЛАРЫН ТАЛДАУДЫҢ ВЕБ-ЖҮЙЕСІНІҢ ЖҰМЫС ЛОГИКАСЫ

Аңдатпа

Мақалада жасанды интеллект технологияларын пайдалана отырып, білім беру құжаттарының мазмұнын бағалауды автоматтандыруға арналған білім беру бағдарламаларын талдаудың веб-жүйесін әзірлеу қарастырылады. Жекелеген құжаттарды жергілікті өңдеуге бағытталған қолданыстағы шешімдерден айырмашылығы, ұсынылған жүйе frontend, backend, реляциялық деректер базасы, PDF құжаттарын өңдеу модулі және жасанды интеллект модулінен тұратын клиент-серверлік архитектура негізінде веб-қосымша ретінде жүзеге асырылған. Өртүрлі рөлдегі пайдаланушылардың (оқытушы, студент және әкімші) өзара әрекеттесуін, деректерді сақтау құрылымын және жүйе ішіндегі өңдеу маршрутын сипаттайтын жүйенің жұмыс логикасы ұсынылған. Білім беру бағдарламаларының мазмұнын интеллектуалды талдау үшін табиғи тілді өңдеу (Natural Language Processing, NLP) әдістері мен үлкен тілдік модельдер (Large Language Models, LLM) қолданылады, бұл әмбебап құзыреттерді (soft skills) автоматты түрде анықтауға және білім беру бағдарламаларын жетілдіру бойынша ұсынымдар қалыптастыруға мүмкіндік береді. Талдау нәтижелері деректер базасында сақталып, веб-интерфейс арқылы пайдаланушыларға қолжетімді болады. Әзірленген веб-жүйе білім беру бағдарламаларын талдау үдерісін автоматтандыруды қамтамасыз етеді, сараптамалық бағалаудың

еңбек сыйымдылығын төмендетеді, талдаудың объективтілігін арттырады және білім беру ұйымдарында еңбек нарығының заманауи талаптарына сәйкес білім беру бағдарламаларын мониторингтеу және жетілдіру мақсатында қолданылуы мүмкін.

Түйін сөздер: білім беру бағдарламаларын талдау; жасанды интеллект; табиғи тілді өңдеу (NLP); үлкен тілдік модельдер (LLM); құжаттарды автоматтандырылған талдау; клиент–серверлік архитектура; реляциялық деректер базасы; білім беру технологиялары; білім беруді цифрландыру

D. Naimanova^{1*}, A. Dautova¹, Zh. Alimova¹, N. Ospanova¹

¹ Toraighyrov University, Pavlodar, Kazakhstan.

OPERATIONAL LOGIC OF THE EDUCATIONAL PROGRAM ANALYSIS WEB SYSTEM

Abstract

This paper presents the development of a web-based system for educational program analysis designed to automate the assessment of educational document content using artificial intelligence technologies. Unlike existing solutions that primarily focus on local processing of individual documents, the proposed system is implemented as a web application based on a client–server architecture comprising a frontend, backend, relational database, PDF document processing module, and an artificial intelligence module. The paper describes the system's operational logic, including interactions among users with different roles (instructors, students, and administrators), data storage organization, and the internal data processing workflow. Natural Language Processing (NLP) techniques and Large Language Models (LLMs) are employed to perform intelligent analysis of educational program content, enabling the automatic identification of soft skills and the generation of recommendations for improving educational programs. The analysis results are stored in a relational database and made available to users through a web interface. The developed web-based system automates the analysis of educational programs, reduces the workload associated with expert evaluation, improves the objectivity of the analysis, and can be applied by higher education institutions to monitor and enhance educational programs in accordance with current labor market requirements.

Keywords: educational program analysis; artificial intelligence; natural language processing (NLP); large language models (LLMs); automated document analysis; client–server architecture; relational database; educational technologies; digitalization of education.

Введение

В условиях цифровой трансформации высшего образования особую актуальность приобретает обеспечение соответствия образовательных программ современным требованиям рынка труда. Быстрое обновление профессиональных компетенций, появление новых технологий и изменение запросов работодателей требуют регулярного анализа содержания образовательных программ и своевременного внесения изменений в их структуру. Наряду с профессиональными знаниями все большее значение приобретают универсальные компетенции (soft skills), формирование которых рассматривается как один из ключевых факторов успешной профессиональной деятельности выпускников.

В Республике Казахстан развитие цифровых технологий в сфере образования является одним из приоритетных направлений государственной политики. Концепция развития высшего образования и науки Республики Казахстан на 2023–2029 годы предусматривает цифровую трансформацию образовательного процесса, внедрение технологий искусственного интеллекта, развитие цифровой инфраструктуры вузов и использование интеллектуальных информационных систем для повышения качества подготовки кадров. Дополнительным фактором является постоянное обновление требований рынка труда, отраженных в профессиональных стандартах и аналитических материалах, включая «Атлас новых профессий и компетенций Казахстана», которые подчеркивают возрастающую значимость цифровых и универсальных компетенций выпускников.

Традиционный анализ образовательных программ осуществляется экспертами вручную, что требует значительных временных затрат и во многом зависит от субъективной оценки специалистов. При большом количестве образовательных программ и постоянном обновлении профессиональных стандартов такой подход становится недостаточно эффективным. В связи

с этим возрастает необходимость создания интеллектуальных информационных систем, способных автоматически анализировать содержание образовательных документов, выявлять соответствие компетенций требованиям работодателей и предоставлять преподавателям рекомендации по совершенствованию образовательных программ.

Развитие технологий обработки естественного языка (Natural Language Processing, NLP) и больших языковых моделей (Large Language Models, LLM) открывает новые возможности для автоматизированного анализа текстовых документов. Использование подобных технологий позволяет анализировать не только наличие отдельных терминов, но и смысловое содержание образовательных программ, оценивать полноту представления компетенций и формировать рекомендации по их совершенствованию.

Научная новизна исследования заключается в разработке архитектуры интеллектуальной аналитической системы для анализа образовательных программ, обеспечивающей полный цикл обработки данных в многопользовательской среде. В отличие от существующих решений, предложенная система реализует интеграцию клиент-серверной архитектуры, реляционной базы данных, модулей обработки документов и больших языковых моделей с механизмом динамического управления промптами, автоматически адаптирующего запросы к LLM в зависимости от роли пользователя, типа документа и этапа анализа. Это обеспечивает более гибкое, масштабируемое и объективное выполнение интеллектуального анализа образовательных программ.

Цифровизация высшего образования обусловила активное развитие исследований, посвященных интеллектуальному анализу образовательных данных и применению технологий искусственного интеллекта для поддержки принятия решений в образовательной среде. Одними из наиболее интенсивно развивающихся направлений являются *Educational Data Mining (EDM)* и *Learning Analytics (LA)*. Согласно обзору Romero и Ventura [1], Educational Data Mining ориентирован на извлечение знаний из образовательных данных с использованием методов интеллектуального анализа, машинного обучения и статистики, тогда как Learning Analytics направлен на сбор, анализ и интерпретацию данных о процессе обучения для повышения его эффективности и качества образовательного процесса. Современные исследования демонстрируют постепенный переход от использования традиционных статистических методов к интеллектуальным прогностическим моделям, основанным на технологиях искусственного интеллекта, обеспечивающим персонализацию обучения и поддержку принятия управленческих решений в образовательных организациях [2].

Развитие методов обработки естественного языка (Natural Language Processing, NLP) и больших языковых моделей (Large Language Models, LLM) значительно расширило возможности автоматизированного анализа текстовых образовательных документов. Современные модели способны выполнять не только поиск ключевых слов, но и анализировать смысловое содержание текстов, выявлять компетенции, определять семантическое соответствие образовательных программ требованиям профессиональных стандартов и рынка труда, а также формировать рекомендации по их совершенствованию. В работе Xu и соавт. [3] показано, что применение архитектуры Retrieval-Augmented Generation (RAG) при анализе учебных программ обеспечивает более высокую точность извлечения профессиональных компетенций по сравнению с традиционными подходами, основанными на стандартных промптах (zero-shot prompting). Аналитические исследования компании Marademics [4] также подтверждают необходимость автоматизации анализа образовательных программ, указывая, что к 2030 году около 39 % востребованных профессиональных навыков изменятся, что требует регулярного обновления содержания образовательных программ.

Особое значение применение больших языковых моделей приобретает в условиях многоязычной образовательной среды Республики Казахстан. В исследовании Togmanov и соавт. [5] представлен первый специализированный бенчмарк *KazMMLU*, предназначенный для оценки качества работы языковых моделей на казахском, русском и английском языках.

Полученные результаты подтверждают перспективность использования LLM при решении прикладных задач анализа образовательных документов и обработки текстовой информации в национальном образовательном пространстве.

Актуальность разработки интеллектуальных образовательных систем подтверждается и государственными инициативами Республики Казахстан в области цифровой трансформации. В соответствии с государственной программой *Digital Qazaqstan* цифровизация образования рассматривается как одно из приоритетных направлений развития страны, предусматривающее внедрение технологий искусственного интеллекта, развитие интеллектуальных образовательных сервисов и использование аналитических систем для повышения качества подготовки кадров [6]. Практическое применение методов Learning Analytics в казахстанских университетах рассмотрено в исследовании Zhumazhanova и соавт. [7], результаты которого демонстрируют положительное влияние аналитики образовательных данных на повышение вовлеченности студентов и эффективность образовательного процесса. Вместе с тем Klevtsov [8] отмечает, что большинство вузов Казахстана пока не готовы к полномасштабному внедрению технологий искусственного интеллекта вследствие отсутствия современных цифровых регламентов и недостаточной интеграции интеллектуальных сервисов в существующую образовательную инфраструктуру.

Анализ существующих исследований показывает, что большинство разработанных решений ориентировано либо на анализ образовательных данных студентов, либо на локальную обработку отдельных документов. При этом недостаточное внимание уделяется построению комплексных программ, обеспечивающих полный цикл обработки данных, включая управление пользователями, централизованное хранение результатов анализа, маршрутизацию данных между компонентами системы и интеграцию модулей искусственного интеллекта в единую информационную среду. Кроме того, существующие решения, как правило, используют статические сценарии взаимодействия с языковыми моделями и не учитывают различия между ролями пользователей и этапами обработки данных.

В отличие от существующих подходов, в настоящем исследовании рассматривается архитектура и разработка интеллектуальной аналитической системы для анализа образовательных программ, реализующая полный цикл интеллектуальной обработки данных в многопользовательской среде. Система объединяет клиент-серверную архитектуру, централизованную реляционную базу данных, модуль обработки PDF-документов, динамическое управление промптами для больших языковых моделей и механизмы ролевого доступа пользователей, что обеспечивает масштабируемость, воспроизводимость результатов анализа и возможность практического внедрения в образовательных организациях.

Анализ существующих исследований показывает, что большинство разработанных решений представляет собой отдельные программные модули или исследовательские прототипы, ориентированные исключительно на выполнение анализа текста образовательной программы. При этом недостаточно рассматриваются вопросы построения полноценной веб-системы, обеспечивающей совместную работу различных категорий пользователей, централизованное хранение данных, разграничение прав доступа и автоматизированное взаимодействие между компонентами системы.

Отсутствие комплексного подхода к организации серверной логики, базы данных, механизмов обработки документов и интеграции модулей искусственного интеллекта затрудняет практическое внедрение подобных решений в образовательных организациях. В связи с этим возникает необходимость разработки интеллектуальной системы, объединяющей процессы загрузки образовательных программ, интеллектуального анализа документов и предоставления результатов различным категориям пользователей в рамках единой архитектуры.

Целью исследования является разработка интеллектуальной аналитической системы для анализа образовательных программ, обеспечивающей автоматизированную обработку

документов, интеллектуальный анализ содержания образовательных программ с использованием технологий искусственного интеллекта и организацию взаимодействия пользователей в рамках единой клиент-серверной архитектуры.

Для достижения поставленной цели были решены следующие задачи:

- разработана интеллектуальная аналитическая система для анализа образовательных программ;
- описана логика работы интеллектуальной аналитической системы: логика работы преподавателей, студента, администратора; база данных и связи между сущностями; маршрут данных внутри системы.

Методология исследования

Методология исследования основана на применении методов системного анализа, объектно-ориентированного проектирования, разработки веб-приложений, технологий обработки естественного языка (Natural Language Processing, NLP) и больших языковых моделей (Large Language Models, LLM). При проектировании системы использовались принципы клиент-серверной архитектуры, обеспечивающие разделение пользовательского интерфейса, серверной логики, хранения данных и интеллектуальной обработки документов.

В процессе исследования разработана интеллектуальная аналитическая система для анализа образовательных программ, реализованная на основе многослойной клиент-серверной архитектуры. Система включает frontend-интерфейс, backend для обработки пользовательских запросов, реляционную базу данных, модуль обработки PDF-документов и модуль искусственного интеллекта, выполняющий интеллектуальный анализ содержания образовательных программ. Организация обработки данных обеспечивает централизованное хранение информации, поддержку многопользовательского режима работы, ролевое разграничение доступа и сохранение состояния объектов системы на всех этапах анализа.

Предлагаемый метод исследования основан на многоэтапной обработке образовательных программ с использованием технологий NLP и LLM. Исходные документы в формате PDF проходят этап извлечения и подготовки текста с применением специализированных библиотек. Подготовленный текст используется в архитектуре RAG, а запросы к языковой модели динамически формируются с учетом типа документа, роли пользователя и задачи анализа. Далее LLM выполняет семантический анализ образовательной программы, выявляет универсальные компетенции (soft skills), оценивает полноту их представления и соответствие установленным требованиям. На основе результатов анализа система автоматически формирует рекомендации по устранению выявленных пробелов и совершенствованию образовательной программы. Метод позволяет автоматизировать анализ документов, повысить объективность оценки и сократить время обработки.

Для проведения интеллектуального анализа в системе применяется подход, основанный на принципах архитектуры Retrieval-Augmented Generation (RAG), при котором извлеченный из документа текст используется в качестве контекстной информации при формировании запроса к языковой модели. Выбор данного подхода обусловлен результатами исследования Ху и соавт. [3], согласно которым применение контекстно-ориентированной архитектуры RAG позволяет повысить точность извлечения компетенций из образовательных программ по сравнению с традиционными подходами к взаимодействию с большими языковыми моделями.

Таким образом, использование архитектуры Retrieval-Augmented Generation позволяет объединить возможности современных больших языковых моделей с информацией, содержащейся непосредственно в анализируемом документе. Это обеспечивает высокую точность извлечения профессиональных компетенций, уменьшает вероятность возникновения ошибок генерации, повышает надежность результатов анализа и делает разработанную систему пригодной для обработки широкого спектра образовательных документов без дополнительного обучения модели.

Одной из ключевых особенностей разработанной системы является механизм динамического управления промптами (Dynamic Prompt Management), обеспечивающий адаптивное взаимодействие с большой языковой моделью. В отличие от традиционного подхода, при котором используется единый статический запрос для всех сценариев обработки документов, разработанная система автоматически формирует содержимое промпта с учетом текущего контекста выполнения задачи. Это позволяет максимально эффективно использовать возможности языковой модели при анализе документов различного типа и назначения.

Необходимость применения механизма динамического управления промптами обусловлена тем, что эффективность работы больших языковых моделей в значительной степени зависит не только от качества входных данных, но и от структуры, содержания и формулировки самого запроса. Даже при использовании одной и той же языковой модели изменение инструкции может существенно повлиять на полноту, точность и формат получаемого результата. По этой причине современные системы, основанные на технологиях искусственного интеллекта, все чаще используют специализированные механизмы генерации промптов, позволяющие адаптировать запрос к конкретной предметной области и решаемой задаче.

В разработанной системе промпт не является заранее фиксированным текстом. Он формируется автоматически на основании нескольких параметров, характеризующих текущий сценарий работы. Основными факторами, влияющими на содержание запроса, являются роль пользователя в системе, тип анализируемого документа, цель анализа и требуемый формат представления результатов. Такой подход позволяет использовать единую языковую модель для решения широкого круга задач без необходимости изменения ее внутренней архитектуры или повторного обучения.

Первым фактором, учитываемым при формировании промпта, является роль пользователя. В зависимости от категории пользователя система может изменять глубину анализа, степень детализации ответа и характер извлекаемой информации. Если системой пользуется администратор, запрос может быть ориентирован на получение технической информации о документе или диагностике процесса обработки. Таким образом, одна и та же языковая модель используется для различных категорий пользователей, однако содержание запроса автоматически адаптируется к их функциональным обязанностям.

Вторым фактором является тип анализируемого документа. Различные виды образовательной документации обладают собственной структурой, содержанием и терминологией. Рабочие программы дисциплин, учебные планы, профессиональные стандарты, образовательные программы и иные документы содержат различные наборы сведений, которые необходимо извлекать в процессе анализа. По этой причине система изменяет инструкцию, передаваемую языковой модели, в зависимости от категории документа. Благодаря такой адаптации языковая модель получает более точные инструкции и концентрируется только на информации, необходимой для конкретного типа документа.

Третьим фактором выступает цель анализа, определяемая выполняемой пользователем задачей. Даже при обработке одного и того же документа система может формировать различные промпты. Например, в одном случае требуется извлечь перечень профессиональных компетенций, в другом – определить результаты обучения, в третьем – проверить соответствие документа установленному шаблону или сформировать краткое аналитическое резюме. Поскольку каждая из перечисленных задач требует различных инструкций, система автоматически изменяет текст запроса, обеспечивая наиболее эффективное использование возможностей языковой модели.

Дополнительно механизм динамического управления промптами учитывает требуемый формат выходных данных. В зависимости от дальнейшего использования результатов языковой модели ответ может формироваться в виде обычного текста, маркированного списка, структурированной таблицы или объекта в формате JSON. Особенно важным является

формирование структурированных данных при их последующей автоматической обработке программными средствами. В этом случае промпт содержит дополнительные инструкции, определяющие строгую структуру ответа, перечень обязательных полей и правила их заполнения. Это значительно упрощает дальнейший программный анализ результатов и их сохранение в базе данных.

Формирование промпта осуществляется программным модулем непосредственно перед обращением к языковой модели. На первом этапе система определяет параметры текущего запроса, включая сведения о пользователе, выбранной операции, типе документа и необходимом формате результата. Затем из набора шаблонов выбираются соответствующие фрагменты инструкций, которые объединяются с контекстом, полученным после извлечения текста из документа в рамках архитектуры Retrieval-Augmented Generation. После этого формируется итоговый запрос, содержащий системную инструкцию, контекст анализируемого документа и описание задачи, который передается языковой модели для выполнения семантического анализа.

Следует отметить, что применение механизма динамического управления промптами позволяет реализовать принцип разделения логики обработки документов и логики взаимодействия с языковой моделью. При необходимости изменения требований к анализу документов разработчику достаточно скорректировать соответствующий шаблон промпта, не изменяя программный код системы и не выполняя повторное обучение модели. Это существенно упрощает сопровождение программного обеспечения и обеспечивает возможность быстрого расширения функциональных возможностей системы.

Использование динамического формирования промптов также способствует повышению качества семантического анализа. Благодаря тому, что языковая модель получает инструкции, максимально соответствующие конкретной задаче, уменьшается вероятность неоднозначной интерпретации запроса, снижается количество нерелевантной информации в ответе и повышается точность извлечения требуемых сведений. Кроме того, адаптация запроса позволяет эффективнее использовать ограниченный контекст языковой модели, включая в него только наиболее значимые инструкции и текстовые данные.

С архитектурной точки зрения механизм динамического управления промптами обеспечивает масштабируемость разработанной системы. По мере появления новых типов документов, новых ролей пользователей или дополнительных аналитических задач отсутствует необходимость переработки существующего модуля искусственного интеллекта. Достаточно добавить новые шаблоны промптов и определить правила их выбора. Такой подход соответствует современным принципам модульного проектирования интеллектуальных информационных систем и обеспечивает возможность дальнейшего развития программного комплекса без существенного изменения его архитектуры.

Таким образом, применение механизма динамического управления промптами является одним из ключевых элементов разработанной системы интеллектуального анализа документов. Его использование позволяет адаптировать взаимодействие с большой языковой моделью к различным сценариям обработки, повысить точность извлечения информации, обеспечить единообразие формируемых результатов, а также значительно увеличить гибкость, расширяемость и сопровождаемость программного обеспечения.

Результаты интеллектуального анализа автоматически сохраняются в реляционной базе данных и становятся доступными пользователям через веб-интерфейс. Дополнительно реализован механизм генерации персонализированной обратной связи, основанный на совместном анализе результатов оценки образовательной программы и ответов студента. Интеграция модулей обработки документов, динамического управления промптами, больших языковых моделей и централизованного хранения данных обеспечивает автоматизацию анализа образовательных программ, снижает трудоемкость экспертной оценки и повышает объективность анализа содержания компетенций.

Результаты исследования

В результате проведенного исследования была разработана интеллектуальная аналитическая система для анализа образовательных программ, обеспечивающая автоматизированную обработку документов и интеллектуальный анализ их содержания с использованием технологий искусственного интеллекта. Реализована логика функционирования системы с учетом сценариев работы преподавателя, студента и администратора. Более подробно остановимся на этом.

1) Разработка интеллектуальной аналитической системы для анализа образовательных программ.

Архитектурной основой системы являются современные веб-технологии и средства искусственного интеллекта. Серверная часть реализована с использованием фреймворка FastAPI, обеспечивающего высокую производительность, поддержку асинхронной обработки запросов и удобное построение REST API для взаимодействия клиентской и серверной частей приложения. Пользовательский интерфейс разработан как веб-приложение, обеспечивающее регистрацию и авторизацию пользователей, работу с личными кабинетами и доступ к функциям системы в соответствии с назначенными ролями.

Для хранения информации используется реляционная база данных, в которой сохраняются сведения о пользователях, образовательных программах, результатах анализа и истории обработки документов. Такой подход обеспечивает целостность данных, поддержку многопользовательского режима работы и возможность последующего анализа накопленной информации.

Обработка образовательных программ осуществляется с помощью специализированных библиотек для работы с PDF-документами, которые выполняют извлечение текстового содержимого и его предварительную обработку. Подготовленный текст передается в модуль интеллектуального анализа, использующий методы обработки естественного языка (Natural Language Processing, NLP) для анализа структуры документа и подготовки контекста.

Интеллектуальный анализ содержания образовательных программ реализован на основе больших языковых моделей (Large Language Models, LLM). Для повышения качества обработки применяется архитектура Retrieval-Augmented Generation (RAG), при которой извлеченный из документа текст используется в качестве контекста запроса к языковой модели. Дополнительно реализован механизм динамического формирования промптов, позволяющий адаптировать инструкции модели в зависимости от типа документа, роли пользователя и выполняемой задачи. Такой подход повышает точность выявления универсальных компетенций (soft skills), снижает вероятность генерации нерелевантной информации и обеспечивает возможность расширения функциональности системы без изменения ее базовой архитектуры (рисунок 1).

ЛОГИКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ДЛЯ АНАЛИЗА ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ

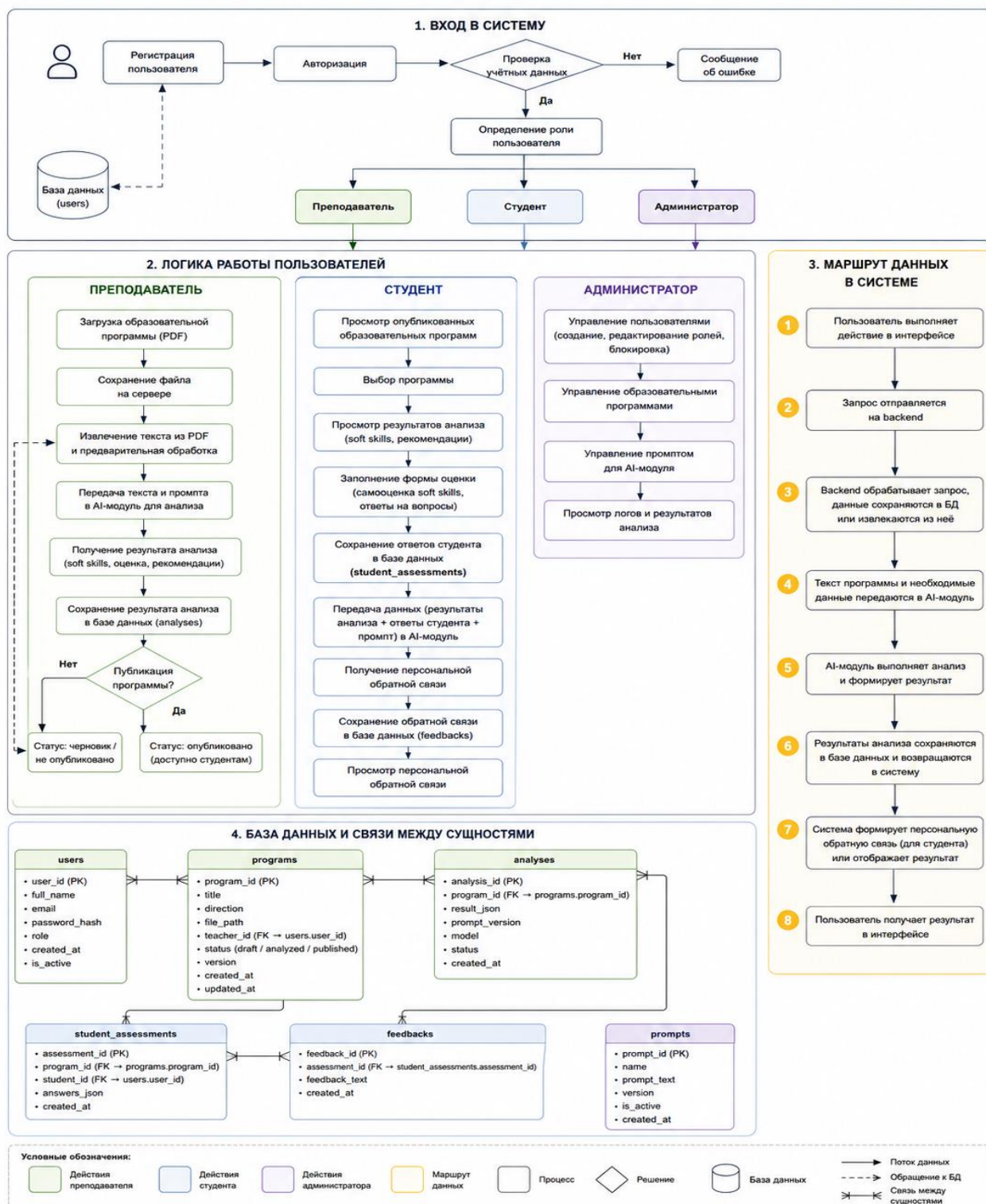


Рисунок 1. Логика работы разработанной интеллектуальной аналитической системы для анализа образовательных программ

2) Сценарии взаимодействия пользователей различных ролей с интеллектуальной системой.

1. Регистрация, авторизация и определение роли

Первый этап работы начинается с регистрации пользователя на сайте. Пользователь заполняет форму, в которую обычно входят имя, электронная почта, пароль и выбранная роль. Frontend передаёт эти данные через HTTP-запрос на backend. Сервер принимает запрос, проверяет корректность формата полей, убеждается, что email ещё не используется, шифрует пароль и создаёт новую запись в таблице пользователей. В базе данных сохраняются идентификатор пользователя, ФИО или имя, email, хэш пароля, роль, дата регистрации и статус активности аккаунта.

После регистрации пользователь проходит авторизацию. Backend сравнивает введённые данные со сведениями из базы, и при успешной проверке создаёт токен доступа или сессионную запись. После этого frontend использует этот токен во всех последующих запросах, чтобы сервер мог определить, кто именно инициирует действие. Это необходимо для разграничения доступа между преподавателем, студентом и администратором.

Данный процесс регистрации, авторизации и разграничения прав доступа пользователей представлен на рисунке 2, который отражает взаимодействие frontend, backend и базы данных при проверке учетных данных и предоставлении доступа к системе.

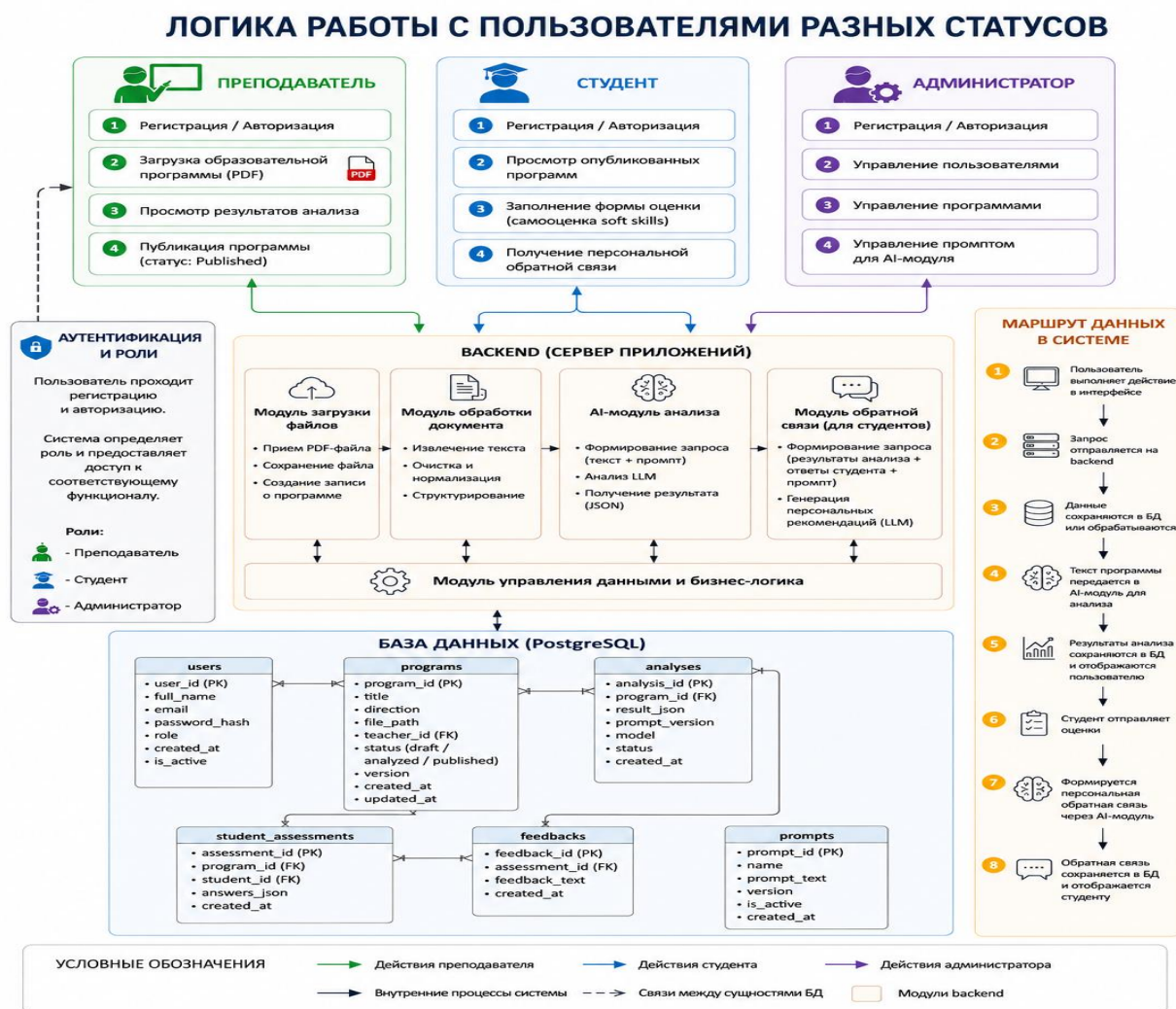


Рисунок 2. Функциональные сценарии интеллектуальной системы анализа образовательных программ

Роль пользователя определяет весь дальнейший сценарий работы. Если в систему вошёл преподаватель, ему открывается интерфейс загрузки образовательных программ и просмотра результатов анализа. Если входит студент, он получает доступ к уже опубликованным

программам, форме оценки и персональной обратной связи. Если входит администратор, он получает полный доступ ко всем сущностям системы, включая изменение промпта, используемого языковой моделью.

2. Логика работы преподавателя

Работа преподавателя начинается с добавления образовательной программы в систему. На сайте он выбирает PDF-файл, при необходимости указывает название программы, направление подготовки, учебный год или комментарий, после чего frontend отправляет файл и сопроводительные данные на backend. Передача файла выполняется как multipart/form-data, потому что в одном запросе нужно передать и двоичные данные файла, и обычные текстовые поля. Сервер получает этот запрос и сначала проверяет, имеет ли пользователь право на выполнение загрузки. Если роль подтверждена, backend создаёт запись о новой образовательной программе в таблице programs. В этой записи могут храниться program_id, название ОП, путь к файлу, идентификатор преподавателя, дата загрузки, статус документа, номер версии и другие служебные поля. Сам PDF-файл обычно сохраняется в файловом каталоге сервера или в отдельном хранилище, а в базу данных записывается путь к этому файлу, чтобы не перегружать БД бинарными данными.

После физического сохранения файла запускается этап его обработки. Специальный модуль открывает PDF-документ, извлекает из него текст, очищает его от лишних символов, служебных переносов, пустых блоков и других элементов, которые не нужны для смыслового анализа. Затем текст может быть разбит по разделам, например по описанию компетенций, результатам обучения, дисциплинам или блокам, где потенциально описаны soft skills. Эта промежуточная текстовая версия документа может сохраняться в базе данных в поле raw_text либо в отдельной таблице, чтобы при необходимости не выполнять повторное извлечение текста из PDF.

Следующим этапом extracted text передаётся в модуль анализа. Здесь backend формирует запрос к AI-модулю, который включает сам текст образовательной программы, активный промпт, системные инструкции и формат ожидаемого ответа. За счёт этого модель получает не просто документ, а чёткую задачу: определить наличие soft skills, выделить найденные навыки, оценить полноту их представления и предложить рекомендации по улучшению образовательной программы.

Ответ модели возвращается на backend уже как структурированный результат. Это может быть JSON-объект, содержащий перечень выявленных soft skills, их категории, текстовые комментарии, итоговую оценку и рекомендации. Сервер сохраняет этот результат в таблицу analyses, где каждая запись связана с конкретной образовательной программой. Также здесь могут фиксироваться версия промпта, дата анализа, статус выполнения и используемая модель.

После завершения анализа frontend получает готовый результат и отображает его преподавателю. На странице анализа он может увидеть структуру найденных soft skills, интерпретацию модели, замечания по содержанию документа и возможные рекомендации. На этом этапе программа ещё не обязательно видна студентам, потому что окончательное решение о её публикации принимает преподаватель.

Если преподавателя устраивает результат, он нажимает кнопку сохранения или публикации. Тогда frontend отправляет на backend команду изменить статус программы. Сервер обновляет запись в таблице programs: например, меняет статус с draft или analyzed на published. С этого момента программа попадает в список доступных для студентов. Если преподаватель не согласен с результатом, он может оставить программу в черновике, удалить её или загрузить исправленную версию для повторного анализа.

3. Логика работы студента

Студент не работает с загрузкой документов и не может изменять образовательные программы. Его сценарий начинается с входа в систему и просмотра опубликованных ОП. Frontend отправляет запрос на backend, сервер выбирает из таблицы programs только те записи,

у которых статус равен `published`, и возвращает их список. Таким образом студент видит только проверенные и подтверждённые программы, а промежуточные версии и черновики ему недоступны.

После выбора конкретной программы студент открывает её страницу. В этот момент `frontend` может запросить сразу несколько наборов данных: общую информацию о программе, сохранённый результат анализа из таблицы `analyses` и форму для ввода собственных оценок или ответов. То есть страница студента строится не только на статических данных программы, но и на результатах уже проведённого анализа.

Далее студент заполняет форму. В зависимости от реализации это может быть самооценка `soft skills`, выставление баллов по отдельным навыкам, ответы на вопросы или текстовые комментарии. После отправки формы все введённые данные уходят на `backend` и валидируются: сервер проверяет, что выбранная программа действительно опубликована, что студент имеет право её проходить, и что структура данных соответствует ожидаемому формату. После проверки запись сохраняется в таблицу `student_assessments`.

В этой таблице могут храниться `assessment_id`, `student_id`, `program_id`, дата отправки, значения оценок, текстовые ответы и статус обработки. Таким образом в базе фиксируется не просто факт взаимодействия студента с программой, а полный набор его ответов, который затем можно использовать для генерации обратной связи, повторного анализа или статистики.

4. Генерация обратной связи студенту

После того как ответы студента сохранены, система переходит к следующему этапу – формированию персональной обратной связи. Для этого `backend` извлекает из базы несколько связанных наборов данных: саму образовательную программу, ранее полученный результат её анализа, ответы студента и активный промпт администратора. Затем всё это объединяется в один входной пакет для AI-модуля.

На этом этапе особенно важно, что система сравнивает разные источники информации. С одной стороны, у неё уже есть анализ того, какие `soft skills` отражены в образовательной программе. С другой стороны, есть реальные оценки или ответы студента по этим же навыкам. Модель получает оба источника и может сформировать персонализированную обратную связь: определить сильные стороны студента, выявить пробелы, показать, насколько его самооценка соотносится с содержанием программы, и предложить рекомендации по развитию.

Сгенерированный результат сохраняется в таблицу `feedback`. В ней могут находиться `feedback_id`, `student_id`, `program_id`, `assessment_id`, текст обратной связи, дата генерации, версия промпта и, при необходимости, техническая информация о модели. После записи в БД `backend` возвращает результат на `frontend`, и студент видит его в своём кабинете или на странице выбранной образовательной программы.

Таким образом, в логике студента происходит важное преобразование данных: сначала он отправляет сырой набор своих ответов, затем эти ответы связываются с анализом образовательной программы, после чего AI-модуль превращает объединённые данные в содержательную обратную связь. Это значит, что обратная связь не существует сама по себе, а возникает как результат объединения нескольких сущностей системы.

5. Логика работы администратора

Администратор является самым привилегированным пользователем системы. Он может выполнять действия преподавателя и студента, а также управлять внутренними настройками сайта. Главная дополнительная функция администратора – изменение промпта, по которому работает языковая модель.

Когда администратор открывает страницу настроек, `frontend` отправляет запрос на `backend` для получения текущего активного промпта. Сервер извлекает его из таблицы `settings` или `prompts` и возвращает в интерфейс. После редактирования администратор сохраняет изменения, и `frontend` отправляет новый текст промпта обратно на сервер.

Backend проверяет права пользователя, создаёт новую запись в таблице промптов или обновляет текущую активную запись, после чего помечает её как действующую. Обычно в такой таблице хранятся `prompt_id`, текст промпта, дата изменения, автор изменения, версия и признак активности. Это позволяет не просто заменить инструкцию, но и хранить историю изменений, чтобы можно было понять, по какой версии промпта выполнялся тот или иной анализ.

После активации нового промпта все последующие обращения к AI-модулю используют уже обновлённую инструкцию. Это касается как анализа вновь загруженных образовательных программ, так и генерации обратной связи студентам. Благодаря этому администратор может изменять критерии, стиль ответа, глубину анализа и формат выводов без изменения исходного кода backend-части.

6. База данных и связи между сущностями

Вся система логически объединяется через единую реляционную базу данных. Основные таблицы можно описать следующим образом: `users` – данные пользователей; `programs` – сведения о загруженных образовательных программах; `analyses` – результаты анализа ОП; `student_assessments` – ответы студентов; `feedback` – сформированная обратная связь; `settings` или `prompts` – системные настройки и активный промпт.

Связи между таблицами определяют логику работы системы. Один преподаватель из `users` может быть связан со многими программами из `programs` через поле `uploaded_by`. Каждая программа из `programs` может иметь одну или несколько записей в `analyses`, если анализ выполнялся повторно или для разных версий документа. Одна опубликованная программа может быть связана со многими студентами через таблицу `student_assessments`. Каждая запись из `student_assessments` может иметь соответствующую запись в `feedback`, в которой хранится итоговая обратная связь. Таблица `prompts` влияет на `analyses` и `feedback`, потому что именно её содержимое используется при запросах к модели.

Такая схема позволяет проследить полный жизненный цикл данных. Например, можно определить, какой преподаватель загрузил конкретную программу, какой текст из неё был извлечён, какой анализ был выполнен, по какому промпту он формировался, какой студент выбрал эту программу, какие оценки он выставил и какую обратную связь в итоге получил. Это особенно важно для прозрачности системы, повторного анализа и аудита результатов.

7. Полный маршрут данных внутри системы

Если описывать систему как непрерывный поток данных, то она работает следующим образом. Пользователь создаёт аккаунт, и его данные сохраняются в `users`. Преподаватель загружает PDF-файл, который физически сохраняется в файловом хранилище, а информация о нём – в `programs`. Затем из PDF извлекается текст, который очищается и передаётся в AI-модуль. Модель формирует результат анализа, а backend записывает его в `analyses`. После одобрения преподавателем программа переводится в статус `published`.

Далее студент выбирает опубликованную программу и отправляет свои оценки. Эти данные записываются в `student_assessments`. После этого backend извлекает из базы анализ программы, ответы студента и активный промпт, объединяет их и отправляет в AI-модуль. На выходе получается персональная обратная связь, которая сохраняется в `feedback` и возвращается студенту через frontend.

Таким образом, данные в системе постоянно меняют состояние. PDF-файл превращается в текст, текст – в анализ, анализ – в опубликованный объект сайта, ответы студента – в основу для персональной обратной связи, а промпт администратора – в управляющий элемент, влияющий на все будущие результаты. Именно эта последовательная трансформация данных и составляет основную логику работы всей программы.

Дискуссия

Полученные результаты согласуются с современными тенденциями развития интеллектуальных образовательных систем, основанных на применении методов Educational

Data Mining, Learning Analytics и технологий искусственного интеллекта. Вместе с тем предлагаемая веб-система отличается от большинства существующих исследований тем, что ориентирована не только на интеллектуальный анализ образовательных документов, но и на организацию полного цикла обработки данных в единой многопользовательской информационной среде.

Результаты исследования Zhumazhanova и соавт. (2026) демонстрируют, что использование методов Learning Analytics в университетах Казахстана, в частности в Казахском национальном университете имени аль-Фараби и Satbayev University, способствует повышению вовлеченности обучающихся и эффективности образовательного процесса [7]. Однако основное внимание в данной работе уделяется анализу образовательных данных студентов и поддержке процесса обучения. В отличие от этого подхода, разработанная интеллектуальная аналитическая система ориентирована на анализ содержания образовательных программ, автоматизированную оценку сформированности универсальных компетенций и поддержку принятия решений при совершенствовании образовательных программ. Таким образом, предлагаемое решение расширяет область применения образовательной аналитики, распространяя ее на этап проектирования и актуализации образовательных программ.

Вместе с тем результаты настоящего исследования подтверждают выводы Klevtsov (2026) о существующих барьерах внедрения технологий искусственного интеллекта в высших учебных заведениях Казахстана [8]. К числу основных ограничений относятся недостаточная готовность цифровой инфраструктуры университетов, отсутствие единых нормативных подходов к использованию генеративного искусственного интеллекта, а также ограниченная интеграция интеллектуальных сервисов с существующими информационными системами вузов. Предложенная интеллектуальная система учитывает указанные ограничения за счет использования клиент-серверной архитектуры, централизованного хранения данных, ролевой модели доступа и модульного построения программного обеспечения, что обеспечивает возможность интеграции с существующими образовательными платформами и дальнейшего расширения функциональности.

Следует отметить, что настоящее исследование имеет ряд ограничений. Предлагаемая система ориентирована на анализ образовательных программ, представленных в формате PDF, а качество интеллектуального анализа зависит от полноты и структуры исходных документов, а также возможностей используемой большой языковой модели. Кроме того, эффективность работы системы требует дальнейшей экспериментальной проверки на более широком наборе образовательных программ различных университетов и направлений подготовки.

Перспективы дальнейших исследований связаны с интеграцией веб-системы с LMS-платформами университетов, использованием технологий Retrieval-Augmented Generation (RAG) с подключением внешних баз знаний, расширением механизмов динамического управления промптами и разработкой инструментов автоматического мониторинга соответствия образовательных программ требованиям профессиональных стандартов и рынка труда.

Заключение

В результате применения предложенных методов была разработана интеллектуальная аналитическая система для анализа образовательных программ, обеспечивающая автоматизированную обработку документов и интеллектуальный анализ их содержания с использованием технологий искусственного интеллекта. Полученные результаты включают описание архитектуры разработанной системы и логики её функционирования, охватывающей взаимодействие пользователей различных ролей, организацию хранения данных и маршрут их обработки внутри системы.

Разработанная система реализована в виде многопользовательского веб-приложения с клиент-серверной архитектурой и включает frontend-интерфейс, backend, реляционную базу

данных, модуль обработки PDF-документов и модуль искусственного интеллекта. Особенностью системы является организация полного цикла обработки данных - от загрузки образовательной программы и извлечения текста до интеллектуального анализа содержания, сохранения результатов и формирования рекомендаций для пользователей различных ролей. В отличие от локальных программных решений, система обеспечивает централизованное хранение данных, сохранение состояния объектов (stateful), разграничение прав доступа и возможность масштабирования для работы нескольких пользователей одновременно.

Интеллектуальный анализ образовательных программ выполняется с использованием методов обработки естественного языка и больших языковых моделей. Реализованный механизм динамического управления промптами позволяет адаптировать запросы к языковой модели в зависимости от роли пользователя, типа анализируемого документа и выполняемой задачи. Это обеспечивает более гибкий и объективный анализ содержания образовательных программ и позволяет автоматически выявлять универсальные компетенции (soft skills), а также формировать рекомендации по их совершенствованию.

Благодарность

Данная статья выполнена в рамках проекта грантового финансирования Министерства образования и науки Республики Казахстан, AP23489805 «Методика интеллектуального анализа данных для выявления факторов, влияющих на профессиональное развитие студентов в технических специальностях».

Список использованных источников

- [1]Romero C., Ventura S. *Educational Data Mining and Learning Analytics: An Updated Survey*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2020. DOI: 10.1002/widm.1355.
- [2]López-Meneses E., et al. *Educational Data Mining and Predictive Modeling in the Age of AI*. Computers. 2025. DOI: 10.3390/computers14020068.
- [3]Xu Z., Yu R., et al. *From Course to Skill: Evaluating LLM Performance in Curricular Analytics*. 2025. DOI: 10.1007/978-3-031-98465-5_26
- [4]Mapademics. *AI-Powered Curriculum Mapping for Academic Excellence*. White Paper. 2025. URL: <https://www.mapademics.com/resources/whitepapers/ai-powered-curriculum-mapping-for-academic-excellence>.
- [5]Togmanov M., et al. *KazMMLU: Evaluating Language Models on Kazakh, Russian, and Regional Knowledge*. arXiv:2502.12829. 2025. DOI: 10.18653/v1/2025.acl-long.701
- [6]Digital Qazaqstan. *Государственная стратегия цифрового развития Республики Казахстан до 2029 года*. URL: <https://adilet.zan.kz/rus/docs/U2600001311>
- [7]Zhumazhanova K. I., et al. *Learning Analytics and Ergonomic Educational Spaces for Active Learning: A Case Study from Kazakhstan*. Frontiers in Education. 2026. DOI: 10.3389/feduc.2026.1801026.
- [8]Klevtsov I. *AI in Kazakh Universities: Institutions Are Not Ready for the ChatGPT Era*. Times of Central Asia. 2026. URL: <https://timesca.com/ai-in-kazakh-universities/>