

Р. В. Енчу¹✉, А. А. Басаргин¹

Разработка веб-сервиса для мониторинга успеваемости с применением технологий искусственного интеллекта

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: yenchuroman@gmail.com

Аннотация. В статье представлена разработка веб-сервиса «КампусПлюс» для мониторинга успеваемости с применением технологий искусственного интеллекта. Актуальность работы обусловлена необходимостью автоматизации контроля знаний в образовательном процессе и устранения разрозненности учебных инструментов. Описан сравнительный анализ существующих платформ – Moodle, Google Forms и Stepik, выявивший отсутствие единого прикладного маршрута преподавателя и встроенных средств AI-генерации тестов. Представлены функциональная и информационно-логическая модели системы, архитектура прототипа на базе FastAPI, Jinja2 и OpenAI API, а также механизм нормализации AI-сгенерированных вопросов. Описаны результаты ручного, автоматизированного и нагрузочного тестирования сервиса. Практическая значимость работы заключается в формировании единого цифрового контура «дисциплина – лекция – тест – попытка – аналитика» и снижении трудоемкости сопровождения дисциплины для преподавателя.

Ключевые слова: мониторинг успеваемости, веб-сервис, искусственный интеллект, генерация тестов, образовательная аналитика, FastAPI, OpenAI API, контроль знаний

R. V. Enchu¹✉, A. A. Basargin¹

Development of a web service for academic performance monitoring using artificial intelligence technologies

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: yenchuroman@gmail.com

Abstract. The article presents the development of the CampusPlus web service for academic performance monitoring using artificial intelligence technologies. The relevance of the work stems from the need to automate knowledge assessment in the educational process and overcome the fragmentation of teaching tools. A comparative analysis of existing platforms – Moodle, Google Forms, and Stepik – is conducted, revealing the lack of an integrated instructor workflow and built-in AI-driven test generation. The paper presents the functional and information-logical models of the system, the prototype architecture based on FastAPI, Jinja2 and the OpenAI API, as well as the normalization mechanism for AI-generated questions. The results of manual, automated and load testing of the service are described. The practical significance of the work lies in establishing a unified digital pipeline «discipline – lecture – test – attempt – analytics» and reducing the workload of teachers in supporting a course.

Keywords: academic performance monitoring, web service, artificial intelligence, test generation, educational analytics, FastAPI, OpenAI API, knowledge assessment

Введение

Современный образовательный процесс в высших учебных заведениях по-прежнему требует от преподавателя значительных временных затрат на подготовку контрольно-оценочных материалов, проверку корректности формулировок тестовых заданий и интерпретацию накопленных результатов. При этом учебный материал, тестирование и аналитика, как правило, разнесены по разнородным инструментам – текстовому редактору, Google Forms или модулям LMS, – вследствие чего утрачивается единый предметный контекст сопровождения дисциплины [1].

Появление больших языковых моделей (LLM) открывает возможность автоматической генерации тестовых вопросов по тексту лекции, однако прямое использование результатов генерации без последующей нормализации и содержательной валидации приводит к появлению дублирующих, шаблонных или семантически некорректных формулировок [2]. Это определяет необходимость разработки системы, в которой AI-модуль выступает помощником преподавателя, а не подменяет его в принятии методических решений.

Целью настоящей работы является разработка специализированного веб-сервиса «КампусПлюс», формирующего единый прикладной маршрут преподавателя: от загрузки лекционного материала до интерпретации результатов тестирования по конкретной учебной группе. В состав задач исследования входят анализ предметной области и существующих решений, проектирование функциональной и информационно-логической моделей системы, разработка прототипа с интеграцией AI-модуля генерации тестов и его апробация на реальных данных.

Исследование предметной области

Контроль знаний обучающихся в высшем учебном заведении представляет собой систематический процесс проверки степени усвоения учебного материала, включающий формирование контрольно-оценочных заданий, проведение тестирования, обработку результатов и их интерпретацию. В практике кафедр преподаватель параллельно ведет несколько дисциплин, закрепленных за различными учебными группами, и регулярно повторяет полный цикл операций – подготовку лекционного материала, составление тестовых вопросов, публикацию доступа и формирование сводных ведомостей. Совокупная трудоемкость этих операций растет линейно с числом дисциплин и групп, тогда как методически значимым этапом, не поддающимся полной автоматизации, остается содержательная интерпретация результатов.

Современные информационные системы поддержки образовательного процесса делятся на три класса. Универсальные платформы управления обучением (LMS), такие как Moodle, предоставляют широкий набор инструментов – курсы, тесты, форумы, журналы оценок, – однако требуют значительной предварительной настройки и не содержат встроенного модуля AI-генерации тестовых заданий по тексту лекции. Упрощенные инструменты создания онлайн-форм (Google Forms, Яндекс.Формы) отличаются простотой интерфейса, но не поддерживают связь тестов с учебным материалом и не предоставляют дисциплинарной

аналитики. Специализированные образовательные платформы (Stepik и аналоги) ориентированы на формат массовых открытых курсов и не учитывают специфику текущего контроля знаний на занятии, в том числе оперативный QR-доступ к тесту [3].

Сравнительный анализ показал, что ни одно из рассмотренных решений не обеспечивает сквозного прикладного маршрута преподавателя, объединяющего подготовку контрольно-оценочных материалов, AI-генерацию вопросов, публикацию доступа на занятии и предметную аналитику в едином цифровом контуре. Данное обстоятельство обосновывает целесообразность разработки специализированного веб-сервиса, ориентированного на типовой сценарий работы преподавателя кафедры прикладной информатики и информационных систем.

Методы и методики

Был проведен сравнительный анализ существующих образовательных платформ контроля знаний – Moodle, Google Forms и Stepik. Moodle представляет собой универсальную систему управления обучением (LMS) с многоуровневой аналитикой, однако ее внедрение требует значительной предварительной настройки и отсутствует встроенный механизм AI-генерации вопросов по тексту лекции. Google Forms ограничена форматом онлайн-формы и не обеспечивает ни связи тестов с лекционным материалом, ни дисциплинарной аналитики. Stepik ориентирован на формат массовых открытых курсов и не учитывает специфику текущего контроля знаний на занятии [3].

Анализ показал, что ни одно из рассмотренных решений не обеспечивает сквозного прикладного маршрута преподавателя, объединяющего подготовку контрольно-оценочных материалов, AI-генерацию вопросов, публикацию доступа к тесту и предметную аналитику в едином цифровом контуре. Данное обстоятельство обосновывает целесообразность разработки специализированного веб-сервиса, ориентированного именно на типовой сценарий работы преподавателя кафедры.

В качестве языка реализации выбран Python. Выбор обоснован прямой интеграцией с библиотеками обработки естественного языка и публичными API больших языковых моделей. Среди веб-фреймворков рассматривались Django, Express, ASP.NET Core и FastAPI; по совокупности критериев – скорость старта, нативная асинхронная обработка запросов, простота интеграции с AI – выбран FastAPI в связке с шаблонизатором Jinja2 [4]. Для хранения данных применяются SQLite на этапе разработки и PostgreSQL в продуктивной среде, что обеспечивает единый прикладной код для обеих СУБД. AI-модуль реализован через OpenAI API с моделью GPT-4.1 [5].

Архитектура решения включает четыре уровня: клиентский уровень (адаптивные HTML-страницы), серверный уровень (маршруты FastAPI, авторизация, бизнес-логика), AI-модуль (генерация тестовых вопросов через OpenAI API) и слой данных (модуль db.py с поддержкой SQLite/PostgreSQL). Для обеспечения отказоустойчивости функции генерации предусмотрено автоматическое переключение на резервный AI-провайдер при недоступности основного, а при

превышении лимитов запросов система предлагает преподавателю воспользоваться ручным конструктором тестов.

Результаты

В рамках проектирования веб-сервиса разработан комплекс UML-моделей, описывающих функциональные, поведенческие и информационно-логические аспекты системы. Диаграмма прецедентов выделяет четырех акторов: преподавателя, студента, администратора и AI-модуль; преподаватель является центральным актором, поскольку именно его прикладной маршрут определяет основную ценность сервиса. Диаграмма классов отражает учебные сущности и связи между ними, в основе модели лежит цепочка «дисциплина → лекция → тест → попытка → аналитика». Диаграмма последовательностей описывает жизненный цикл теста – от создания лекции и AI-генерации вопросов до получения аналитики после прохождения студентом, а диаграмма активности моделирует процесс работы преподавателя с развилкой выбора между автоматической AI-генерацией и ручным конструктором тестов.

Физическая модель базы данных построена по фактической структуре проекта и включает таблицы `users`, `groups`, `disciplines`, `lectures`, `tests`, `questions`, `attempts` и `answers`, а также служебные таблицы `group_teachers`, `teacher_disciplines`, `teaching_assignments` и `teaching_assignment_blocks`. Ключевая таблица `teaching_assignments` реализует тройки «преподаватель + дисциплина + группа» и определяет, к каким тестам имеют доступ обучающиеся и какие результаты попадают в преподавательскую аналитику [6, 7]. Доступ преподавателя к студентам реализован не напрямую, а через указанную тройку, что точнее отражает реальный учебный процесс.

Прототип системы реализован как полнофункциональное веб-приложение с серверным рендерингом на базе FastAPI и Jinja2. Интерфейс охватывает три ролевых контура – преподавателя, студента и администратора – и адаптирован для мобильных устройств, что позволяет студентам проходить тестирование непосредственно на занятии через QR-код. Стартовый экран веб-сервиса (рис. 1) обеспечивает быстрый вход в систему и обзор ключевых разделов платформы: личного кабинета, результатов тестирования и точек роста обучающегося.

Особое внимание уделено контролю качества AI-генерации тестовых вопросов. Серверный модуль нормализации выполняет дедупликацию вопросов по порогу текстового сходства ($\geq 82\%$), фильтрацию шаблонных и обрезанных формулировок, проверку соотношения правильного и неправильных ответов (при сходстве $\geq 90\%$ вопрос отбрасывается), а также рандомизацию порядка вариантов ответа. При этом преподаватель остается в контуре принятия решения – он видит все вопросы до публикации и может удалить, отредактировать или дополнить набор [8].

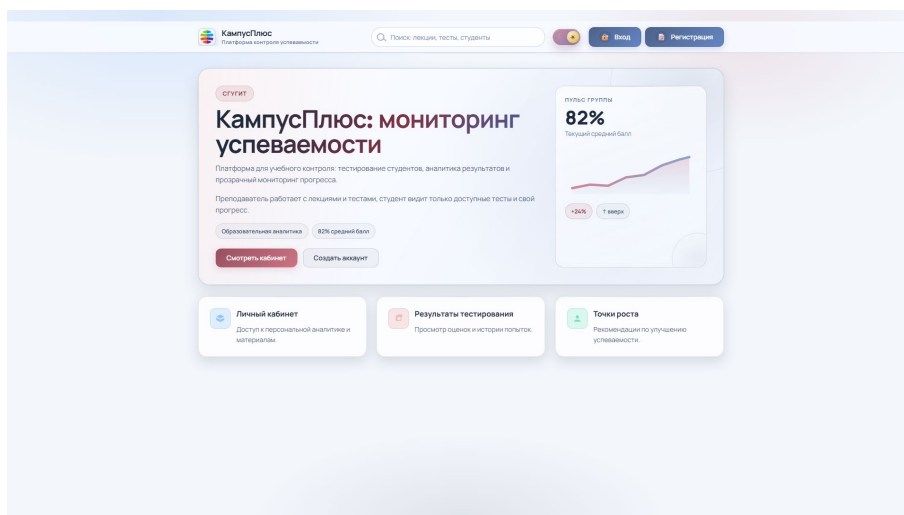


Рис. 1. Стартовый экран веб-сервиса «КампусПлюс»

Тестирование системы выполнено в трех форматах: ручное сценарное тестирование (20 тест-кейсов по пяти направлениям), автоматизированное модульное и интеграционное тестирование на базе pytest (свыше 140 тестов, покрывающих авторизацию, разграничение ролей, бизнес-логику, SQL-хелперы и smoke-проверки), а также нагрузочное тестирование. Нагрузочное тестирование подтвердило соответствие сервиса заявленным нефункциональным требованиям: среднее время отклика составило 48 мс, пропускная способность – 142 запроса в секунду, доля успешных запросов – 99,8 % при P95-latency 174 мс.

Заключение

В результате выполнения работы разработан и апробирован веб-сервис «КампусПлюс» для мониторинга успеваемости с применением технологий искусственного интеллекта. Сервис формирует единый прикладной маршрут преподавателя «дисциплина – лекция – тест – попытка – аналитика», обеспечивает автоматизированную AI-генерацию тестовых вопросов с последующей нормализацией и валидацией преподавателем, а также предметную аналитику по группам и отдельным студентам. Прототип развернут на облачной платформе Render и доступен по адресу campusplus.onrender.com. Результаты работы могут быть использованы для сопровождения текущего контроля знаний обучающихся на кафедре прикладной информатики и информационных систем СГУГиТ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Басаргин, А. А. Методы и средства проектирования информационных систем / А. А. Басаргин. – Новосибирск: СГУГиТ, 2015. – 236 с.
2. Brown, T. Language Models are Few-Shot Learners / T. Brown [et al.] // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 1877–1901.
3. Coates, H. Student engagement in campus-based and online education / H. Coates. – London: Routledge, 2006. – 200 p.
4. Tiangolo, S. FastAPI: modern, fast web framework for building APIs with Python. – URL: <https://fastapi.tiangolo.com> (дата обращения: 10.03.2026).

5. OpenAI. Models – OpenAI API documentation. – URL: <https://platform.openai.com/docs/models> (дата обращения: 15.03.2026).
6. Буч, Г. UML. 2-е издание / Г. Буч, А. Якобсон, Д. Рамбо. – Санкт-Петербург: Питер, 2005. – 736 с.
7. Райордан, Р. Основы реляционных баз данных / Р. Райордан. – Москва: Русская Редакция, 2010. – 384 с.
8. Anil, R. Gemini: A Family of Highly Capable Multimodal Models / R. Anil [et al.] // arXiv preprint arXiv:2312.11805. – 2023.

© *Р. В. Енчу, А. А. Басаргин, 2026*