



МЕТОДОЛОГИЯ И ТЕХНОЛОГИЯ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ/METHODOLOGY AND TECHNOLOGY OF VOCATIONAL EDUCATION

DOI: <https://doi.org/10.60797/IRJ.2026.165.21>**РАЗРАБОТКА И ВЕРИФИКАЦИЯ МЕТОДИКИ ИНТЕГРАЦИИ ИГРОВЫХ МЕХАНИК В ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ**

Научная статья

Нуриев М.Г.^{1,*}, Махфуд Б.А.², Толмачева А.В.³, Лаптева М.Г.⁴, Молостцова Д.А.⁵¹ORCID : 0009-0003-0741-1734;^{1,2,5} Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, Казань, Российская Федерация^{3,4} Казанский национальный исследовательский технологический университет, Казань, Российская Федерация

* Корреспондирующий автор (mgnuriev[at]kai.ru)

Аннотация

Статья посвящена разработке и экспериментальной верификации методики интеграции игровых механик в процесс онлайн-обучения программированию. Актуальность исследования определяется проблемой низкой мотивации и высокого процента отсева обучающихся на дистанционных курсах по программированию. Целью работы является проектирование, реализация и количественная оценка эффективности образовательной веб-платформы, объединяющей элементы геймификации (система вознаграждений, визуализация прогресса, рейтинги) и механизмы социального взаимодействия (взаимная оценка решений, комментарии). Методология исследования включает анализ существующих решений, проектирование архитектуры системы, разработку серверной части на базе Django REST Framework с использованием PostgreSQL и клиентского интерфейса на React, а также комплексное тестирование с применением метрик производительности, надёжности и удобства использования. Верификация методики проведена на основе количественных показателей: время отклика API (120–180 мс), покрытие функциональных тестов (94%), коэффициент успешных запросов (99,7%), оценка удобства использования по методике System Usability Scale (78,5 балла). Научная новизна заключается в обосновании комплексного подхода к интеграции геймификационных и социальных механизмов в образовательные платформы с количественной верификацией эффективности. Практическая значимость определяется возможностью применения разработанной методики для повышения результативности дистанционного IT-образования.

Ключевые слова: геймификация в образовании, верификация методики, онлайн-обучение программированию, метрики качества ПО, System Usability Scale, социальное взаимодействие, образовательные веб-платформы, Django REST Framework, React.

DEVELOPMENT AND VERIFICATION OF METHODS FOR INTEGRATING GAMING MECHANICS INTO THE EDUCATIONAL PROCESS OF TEACHING PROGRAMMING

Research article

Nuriev M.G.^{1,*}, Makhfud B.A.², Tolmacheva A.V.³, Lapteva M.G.⁴, Molostvova D.A.⁵¹ORCID : 0009-0003-0741-1734;^{1,2,5} Kazan National Research Technical University named after A.N. Tupolev – KAI, Kazan, Russian Federation^{3,4} Kazan National Research Technological University, Kazan, Russian Federation

* Corresponding author (mgnuriev[at]kai.ru)

Abstract

The article is devoted to the development and experimental verification of a methodology for integrating gaming mechanics into the process of online programming education. The relevance of the research is determined by the problem of low motivation and high dropout rates among students in remote programming courses. The aim of the work is to design, implement, and quantitatively evaluate the effectiveness of an educational web platform that combines elements of gamification (reward system, progress visualisation, ratings) and social interaction mechanisms (mutual evaluation of solutions, comments). The research methodology includes analysis of existing solutions, system architecture design, development of the server part based on Django REST Framework using PostgreSQL and the client interface on React, as well as complex testing using performance, reliability and usability metrics. The methodology was verified based on quantitative indicators: API response time (120–180 ms), functional test coverage (94%), successful request rate (99.7%), and usability evaluation according to the System Usability Scale (78.5 points). The scientific novelty lies in the substantiation of a comprehensive approach to the integration of gamification and social mechanisms into educational platforms with quantitative verification of effectiveness. The practical significance is determined by the possibility of applying the developed methodology to improve the effectiveness of distance IT education.

Keywords: gamification in education, methodology verification, online programming learning, software quality metrics, System Usability Scale, social interaction, educational web platforms, Django REST Framework, React.

Введение

Развитие информационных технологий в XXI веке оказало огромное влияние на систему образования, изменив традиционные методы преподавания и заложив основу для создания новых моделей обучения [1], [2]. Кроме того, основной тенденцией последнего десятилетия является быстрый рост дистанционного образования, которое сегодня составляет важную часть образовательного процесса, предоставляя доступ к знаниям независимо от географического положения и временных ограничений. В то же время постоянный рост спроса на ИТ-специалистов подразумевает необходимость поиска эффективных стратегий, если действительно нужно реализовать какую-либо серьезную программу по подготовке персонала.

Традиционные способы передачи материала, такие как лекции или статичные учебники, в большинстве случаев неэффективны, несмотря на многочисленные очевидные преимущества дистанционного обучения. Основная проблема заключается в том, что со временем такие методы снизили интерес студентов к выполнению заданий, что привело к резкому падению мотивации и усложнило освоение сложных технических дисциплин; в связи с этим продолжается поиск путей совершенствования взаимодействия и персонализации в процессе обучения. Уже реализуемые решения основываются на геймификации, отслеживании прогресса, персонализации обучения и обеспечении социального взаимодействия между участниками образовательной среды; эти решения, в свою очередь, способствуют повышению мотивации и создают условия для формирования сообществ практиков, основанных на взаимной поддержке.

В рамках данного исследования предлагается разработка веб-платформы для изучения языков программирования, в основу которой положена идея интеграции геймификации и механизмов социального взаимодействия. Научная новизна работы заключается в комплексном подходе к созданию образовательного сервиса, сочетающего игровые элементы, последовательное построение учебных заданий и средства персонализации профиля пользователя. Практическая значимость работы определяется возможностью повышения качества подготовки ИТ-специалистов за счет формирования устойчивой мотивации к обучению и создания доступной среды для освоения программирования в условиях глобальной цифровизации образования.

Для достижения поставленной цели предусматривается проведение анализа современного рынка онлайн-обучения и выявление ключевых требований к разрабатываемому сервису, проектирование базы данных, реализация серверной и клиентской частей системы, а также тестирование платформы на предмет функциональной полноты и удобства использования. Ожидается, что результаты работы позволят создать эффективный инструмент для организации интерактивного образовательного процесса, способный повысить результативность обучения и способствовать подготовке специалистов, востребованных в современной цифровой экономике.

Анализ существующих решений

Предметная область данного проекта охватывает сферу онлайн-образования, сосредоточенную на изучении языков программирования, с акцентом на геймификацию и социальное взаимодействие. В последние годы рынок онлайн-обучения демонстрирует устойчивый рост, что обусловлено развитием технологий, увеличением доступности интернета и ростом спроса на профессии в сфере ИТ (Information Technology) [3], [4]. Основными драйверами роста являются:

- интеграция искусственного интеллекта и персонализированного обучения;
- расширение использования виртуальной и дополненной реальности в образовательных процессах;
- рост спроса на онлайн-обучение и гибкие формы образования.

Одновременно с этим с каждым годом все больше отраслей переходят к использованию цифровых технологий, что требует квалифицированных кадров в сфере ИТ. В России эта тенденция усиливается политикой импортозамещения, из-за которой растёт необходимость в разработке отечественного ПО.

Цель проекта — создать веб-платформу, которая использует игровой подход для обучения программированию. Этот проект предусматривает интерактивную карту задач и систему вознаграждений (таких как опыт, монеты и звезды), а также социальные взаимодействия, такие как взаимная оценка решений, обзоры и комментарии. Платформа предназначена для предоставления пользователям онлайн-компилятора для практики без каких-либо сторонних сервисов, что позволяет немедленно закрепить знания, полученные в ходе упражнений. Одним из цифровых строительных блоков веб-сайта является персонализация, то есть валюта, которую пользователи получают по каналам заданий, будет потрачена в магазине на что-либо полезное, например, на украшение своего профиля, дополнительные карты для обучения и т. д. Анализ домена подтверждает важность и актуальность проекта, который появился в нужное время, удовлетворяя текущие потребности онлайн-курсов и привлекая большое количество пользователей, чтобы создать более привлекательный способ обучения программированию, который был бы простым, увлекательным и социально ориентированным.

Анализ конкурентов помогает нам изучить существующие платформы в области онлайн-образования в области программирования и выявить их сильные и слабые стороны, чтобы определить сравнительные преимущества предлагаемого проекта. Ключевыми среди этих платформ являются LeetCode и School 21, каждая из которых имеет свои отличительные особенности и клиентуру [5], [6].

LeetCode — это известная онлайн-платформа для решения алгоритмических задач, которая привлекает особенно много трафика со стороны соискателей в сфере технологий. Однако у неё есть значительные недостатки:

- отсутствие четкой учебной карты затрудняет новичкам структурированное освоение материала;
- визуализация прогресса ограничена простыми показателями количеством решенных задач и дней обучения без пропуска;

- встроенная теория к задачам либо платная, либо на английском языке, что не очень удобно.

Эти проблемы вызывают трудности в изучении у начинающих, так как для них необходимо пошаговое обучение.



Школа 21, в свою очередь, предлагает бесплатную IT-образовательную программу с акцентом на самообучение и практические задачи в игровом формате. Её сильная сторона — фокус на командной работе и взаимной поддержке студентов, что создаёт уникальную среду для развития soft skills и технических навыков.

Однако доступ к обучению осложнён:

- поступление требует прохождения месячного интенсивного курса и победы в конкурсном отборе;
- для успешного прохождения отбора необходимы базовые знания языка C;
- в начале обучения отсутствует выбор языка программирования для изучения.

Сравнение с этими конкурентами подчёркивает, что предлагаемый проект может занять уникальную нишу. В отличие от LeetCode, платформа предоставляет интерактивную карту задач с последовательным обучением, визуализацию прогресса и встроенную теорию, что делает её более подходящей для начинающих. По сравнению с Школой 21, она не требует строгого отбора или предварительных знаний, предлагая свободный выбор языков программирования и доступность. Таким образом, проект сочетает преимущества геймификации, гибкости и доступности, отличаясь от существующих решений и удовлетворяя потребности широкой аудитории.

На основе анализа предметной области и конкурентов, проведенного ранее, были определены требования для разрабатываемой образовательной веб-платформы, направленной на изучение языков программирования в игровой форме.

Основные требования:

- создание и прохождение интерактивной карты задач с последовательным доступом к проектам, добавление временных проектов;
- поддержка онлайн-компилятора для проверки решений на разных языках программирования;
- возможность наглядно видеть свой прогресс благодаря диаграммам полученных навыков и графикам;
- реализация системы наград (опыт, монеты, звезды) с сохранением прогресса в базе данных;
- возможность публикации решений и их оценки (лайки) другими пользователями;
- возможность оставлять комментарии на чужие решения, удалять свои комментарии;
- формирование рейтингов пользователей по опыту и звездам с отображением в таблице;
- создание магазина для покупки стилей профиля (никнейм, фон) и дополнительных карт для обучения за монеты и звезды.

Отсюда давайте рассмотрим структуру сайта, учитывая установленные для него требования. На приведенной схеме (рис. 1) показаны основные компоненты сайта.

Разработка веб-сайта включает в себя серию страниц, которые обеспечивают членам организации столь необходимую доступность и функциональность.

Страница регистрации и авторизации позволяет зарегистрироваться через email и предоставляет доступ к личному кабинету. В кабинете отображается прогресс пользователя: последние решённые задачи, количество монет и звёзд, диаграмма навыков, график опыта, а также настройки профиля для персонализации.

Страница выполненных, временных и решённых проектов представляет список задач с кратким описанием, сложностью и наградой. Нажав на проект, можно перейти на страницу с его выполнением.

Карта проектов визуализирует прогресс в виде интерактивной карты, где новые задания открываются по мере выполнения текущих, мотивируя пользователя двигаться вперёд. Есть основная карта и дополнительные, которые можно открыть за награды, полученные в процессе прохождения основной карты.

Страница магазина позволяет тратить монеты и звезды на покупку дополнительных карт или персонализацию: анимации ника, оформление профиля и цветные ники, с каталогом товаров, где указаны описание и стоимость.

Страница рейтинга отображает таблицу лидеров по критериям опыта и звёзд за месяц и за всё время, стимулируя конкуренцию.

Страница проекта включает онлайн-компилятор с поддержкой различных языков программирования. Она содержит вкладки: «Описание задания» с текстом задачи и примерами ввода-вывода, «Теория» с учебными материалами и «Результат». Пользователь может протестировать код перед отправкой и получить результат проверки.

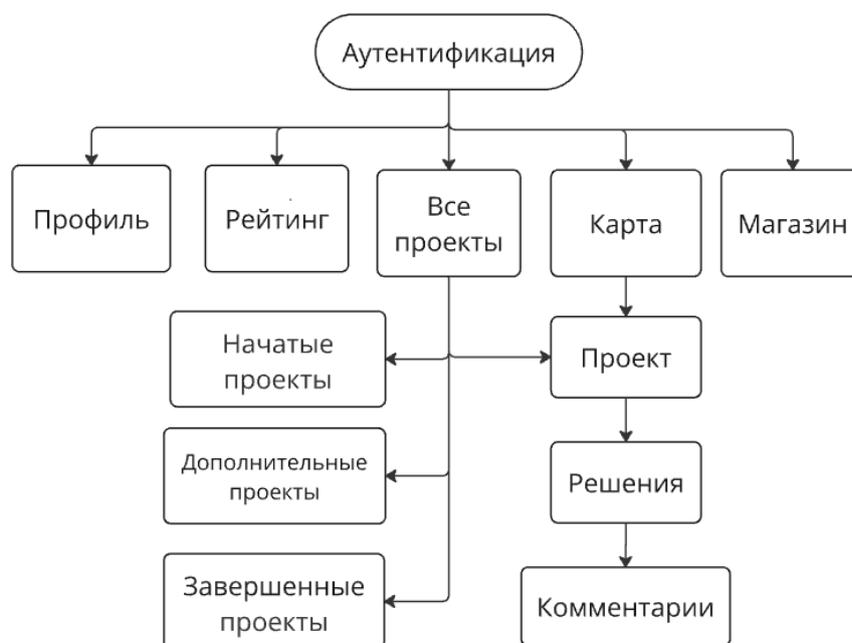


Рисунок 1 - Структура проекта
DOI: <https://doi.org/10.60797/IRJ.2026.165.21.1>

Страница с решениями открывается после выполнения задания: здесь можно просматривать коды других пользователей, читать комментарии, оценивать решения и публиковать своё, получая звёзды за оценки.

Рассмотрим перечень технологий, которые будут необходимы для реализации платформы с требуемыми функциями и характеристиками.

Для реализации базы данных проекта был выбран PostgreSQL. PostgreSQL — это бесплатная объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом, которая обеспечивает высокую надежность, поддержку сложных запросов и масштабируемость [7], [8]. Она была выбрана из-за необходимости эффективно хранить и управлять большими объемами данных о пользователях, проектах, задачах и социальных взаимодействиях. Так же эта СУБД обладает встроенными инструментами индексации и транзакционной целостности, что гарантирует стабильность платформы при высоких нагрузках. PostgreSQL соответствует требованиям ACID (Atomic Consistency Isolation Durability). Это четыре требования для надёжной работы систем, которые обрабатывают данные в режиме реального времени. Если все требования выполняются, данные не будут теряться из-за технических ошибок или сбоев в работе оборудования. Все вышеперечисленные преимущества делают PostgreSQL оптимальным выбором для данного проекта.

Для реализации серверной части проекта был выбран Django REST Framework (DRF). DRF — это фреймворк языка Python, основанный на Django, который предоставляет встроенные инструменты для создания REST API (Representational State Transfer Application Programming Interface), такие как аутентификация, ORM (Object-Relational Mapping) для работы с PostgreSQL или другими СУБД и сериализаторы [9], [10]. Это сильно упрощает разработку API. В отличие от аналогов, таких как Flask или Express, DRF обладает готовой инфраструктурой для аутентификации, авторизации, сериализации данных, пагинации и другими полезными функциями, облегчающие работу разработчика. Для клиентской части проекта были выбраны React и Axios, что стало оптимальным решением для создания динамичного и интерактивного интерфейса образовательной платформы. React гарантирует высокую производительность даже при большом количестве пользователей, что важно для платформы с интерактивными элементами, такими как онлайн-редактор кода и карта. Использование CSS-модулей в сочетании с React упрощает стилизацию.

Причины выбора Axios для выполнения запросов HTTP (HyperText Transfer Protocol) заключаются в его простоте и гибкости. Он хорошо интегрируется с React для взаимодействия с DRF API, передавая данные и получая ответы. Наличие авторизованных запросов и возможностей обработки ошибок важно для API, которые требуют токенов для повышения безопасности.

Наличие Axios является важной причиной, по которой можно продолжить компиляцию кода в Visual Studio Code (VS Code). Эта среда была выбрана в качестве среды разработки, поскольку Visual Studio Code гарантирует современные утилиты с точки зрения организации, отладки, интеграции компилятора.

Проектирование базы данных

Были предприняты дополнительные усилия по формализации структуры реляционной базы данных PostgreSQL (рис. 2), в которую будет вводиться ожидаемая информация. Она была разработана таким образом, чтобы ее можно было использовать для интенсивного кодирования, развертывания и модификации реляционной структуры гетерогенных узлов после вычислений, что в данном контексте может не поддерживаться стилем NoSQL.

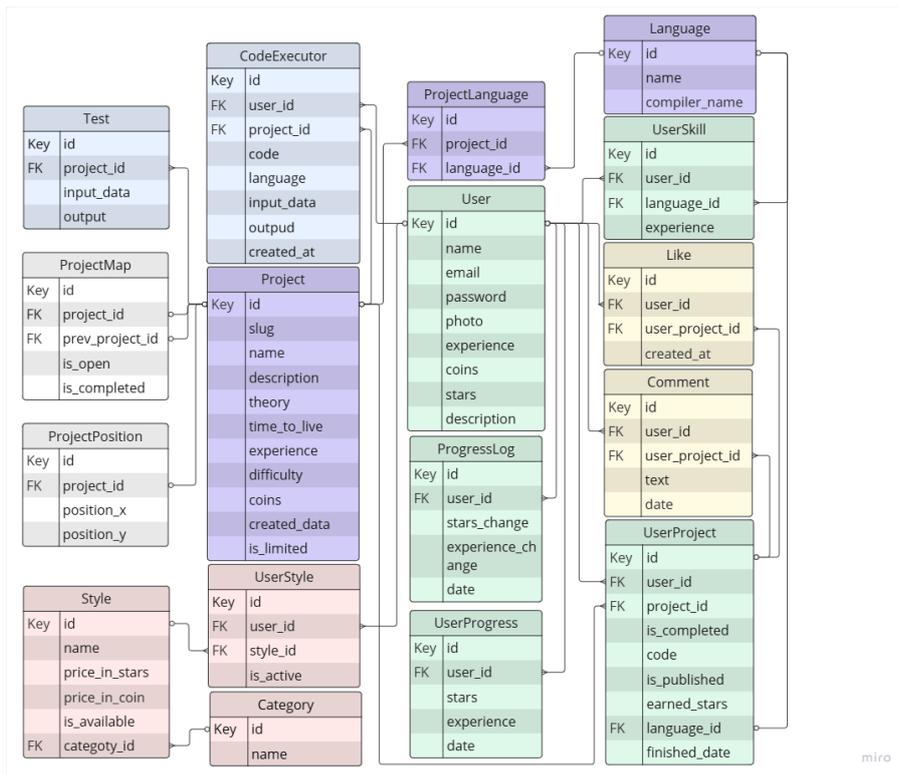


Рисунок 2 - Структура базы данных
DOI: <https://doi.org/10.60797/IRJ.2026.165.21.2>

Для реализации образовательной веб-платформы была смоделирована реляционная база данных для хранения информации о проектах, пользователях, их прогрессе и взаимодействиях в системе. Особое внимание было уделено логической структуре данных и соответствующим отношениям между основными сущностями для обеспечения эффективной обработки запросов и целостности данных.

Ядром базы данных является таблица «Проект», в которой хранится вся информация о проектах. Ключевые характеристики проекта включают уникальный идентификатор, название проекта, описание проекта, теоретическую часть, уровень сложности, срок выполнения проекта, XP за опыт, количество монет в качестве вознаграждения и временный флаг проекта. Дата создания сохраняется для отслеживания положения задач.

Таблица ProjectMap описывает взаимосвязи между проектами для хранения информации о последовательности задач. Записи для проекта p указывают на текущий и предыдущий проекты, а также на их доступность и статус завершения. Физическое расположение проектов на карте хранится в таблице ProjectPosition, которая

- хранит координаты X и Y для проекта;
- сопоставляет записи с таблицей Project по ее идентификатору.

Таблица Language содержит уникальные идентификаторы, имена и соответствующие обозначения компиляторов для различных языков программирования, которые поддерживаются системой. Связывание проектов с языками программирования облегчается с помощью таблицы ProjectLanguage, которая определяет, какие языки разрешено использовать для конкретной задачи.

Таблица CodeExecutor предоставляет сервис для извлечения и выполнения булевых решений пользователей; содержание самой таблицы включает в себя идентификатор пользователя, связанный проект, входные данные, результаты вывода, написанный код, язык программирования и дату публикации. Понимание версий может быть реализовано в тестовой таблице с подсчетом совпадений входных/выходных случаев, найденных в каждом проекте.

Пользователи низкого уровня системы предназначены для управления таблицей пользователей внутри схемы. Помимо обработки обычных статусов, таких как имя пользователя, адрес электронной почты, пароль, фотография профиля и информация, некоторые другие записи включают увеличение очков, монет и звезд. Это также позволяет всему компьютеру писать на языке для отслеживания действий пользователя.

Связь между пользователями и проектами реализуется через таблицу UserProject, в которой хранится информация о выполнении задач, сохраненный код, выбранный язык, количество заработанных звезд и дата завершения. Динамика образовательного процесса отражается в таблице UserProgress, в которой хранится опыт, заработанный на определенную дату, и заработанные звезды. Таблица ProgressLog используется для отслеживания изменений в деталях, где записываются все операции, связанные с получением или потерей звезд и опыта.

Таблица UserSkill соответствует уровню владения языком для пользователей; она варьируется в пределах трех значений опыта для каждого языка. Это делается так, как пользователи приобретают эти стили и отслеживают их с помощью записи в таблице UserStyle. Содержимое таблицы Style иллюстрирует сбор необходимых материалов,

которые имеют названия, звезды, серебро и категорию, к которой они принадлежат. В этом случае таблица Category будет объединять категории, присваивая каждой идентификатор в качестве ее категории.

Система рейтингов и комментариев представляет собой важную часть взаимодействия между пользователями, для которой была создана таблица Like, в которой отслеживаются оценки пользователей, поставленные проектам в разные даты. Коммуникация в то же время также включает в себя систему, в которой комментарии подписчиков Comment публикуются с отметками времени.

При разработке базы данных было тщательно проанализировано функционирование платформы, от деталей проектов и языков программирования до способов взаимодействия пользователей, отслеживания прогресса и настройки внешнего вида интерфейса. Понимание этих аспектов важно как для быстрого управления данными, так и для относительно простого масштабирования системы в соответствии с растущими требованиями образовательного процесса.

Разработка серверной и клиентской частей

Серверная часть платформы разработана с использованием Django REST Framework (DRF). API сервера позволяет управлять пользователями, пользовательскими проектами, комментариями и другими частями платформы [11], [13], [14], [15]. Основные шаги разработки:

- создание отдельных приложений, каждое из которых будет отвечать за определённую функциональность: управление пользователями, проектами, картой задач, комментариями, магазином стилей и компилятором;
- проектирование моделей данных, отражающих сущности платформы (пользователи, проекты, награды, стили), с сохранением в PostgreSQL;
- внедрение системы аутентификации с использованием Simple JWT для генерации и проверки токенов, обеспечивая безопасный доступ к данным платформы [16], [17];
- определение маршрутов (URL-паттернов) для каждого приложения, связывая их с соответствующими представлениями и сериализаторами, для обработки HTTP-запросов;
- настройка интеграции с внешними сервисами, такими как онлайн-компилятор, для проверки пользовательских решений.

Для организации работы с пользовательскими данными было создано приложение, обеспечивающее управление регистрацией, авторизацией, профилями и рейтингом участников. Оно позволяет хранить и обрабатывать данные о пользователях, отслеживать их прогресс, а также формировать персональные профили с учетом полученного опыта, достижений и статистики. Отдельный модуль был выделен для работы с образовательными проектами. В нём реализована логика добавления, редактирования и отображения заданий, а также взаимодействия с картой задач, где задания связаны между собой в определённой последовательности [18], [19].

Особое внимание было уделено приложению, связанному с интерактивной картой проекта, которая визуализирует ход проекта, открывает новые задачи и связывает их между собой. В проект было внедрено приложение, облегчающее организацию онлайн-компилятора, которое дает возможность разрабатывать решение и проверять его без привлечения сторонних систем. Теперь пользователи могут писать и тестировать свой код прямо на платформе.

Приложение для комментариев и оценок пользователей предоставляет возможность взаимодействия с социальным сообществом. Оно позволяет пользователям комментировать, оценивать решения других пользователей и обмениваться мнениями, фактически создавая на платформе активное сообщество. В будущем может появиться еще одно приложение, связанное со стилевым магазином, где пользователи будут тратить свои награды на дизайн профиля, что позволит сохранить элементы геймификации и поддержать энтузиазм.

Каждое из приложений имеет идентичную структуру, соответствующую стандартам Django. В неё входят файлы инициализации, конфигурации, модели данных, сериализаторы, маршруты и представления. Это обеспечивает единообразие и предсказуемость кода, облегчая работу как для текущего разработчика, так и для будущих команд сопровождения.

После создания отдельных приложений следующим этапом стала разработка моделей данных, которые легли в основу серверной части образовательной платформы. В контексте фреймворка Django модели представляют собой классы на языке Python, описывающие структуру данных и напрямую связанные с таблицами базы данных PostgreSQL. Такой подход позволяет использовать встроенный механизм объектно-реляционного отображения, что обеспечивает прозрачное взаимодействие между логикой приложения и системой управления базами данных [20], [21], [22].

Каждая модель соответствует одной из сущностей платформы, отражая таким образом ее основные характеристики. Например, модель, связанная с образовательными задачами, имела поле ID, название, уровень сложности, теоретические материалы и баллы для вознаграждения. Таким образом, можно было собрать полную информацию о проекте, которая использовалась при формировании моделей задач и во взаимодействии с пользователями. Постоянный акцент делался на существовании условных проектов, для которых переменными, включенными в модель, являются временные ограничения и ограничения любого рода.

Модели уверенных пользователей содержат всю необходимую информацию о профиле, включая имя, адрес электронной почты, пароль, фотографию и описание. Кроме того, они содержат слоты для сохранения опыта, количества монет и звезд, что помогает отслеживать прогресс в улучшении навыков и получении рейтинга. Этот основной экземпляр модели не может отображать личную информацию, поскольку только вспомогательные модели предоставляют возможность увидеть, какие проекты были выполнены, сколько подцелей было достигнуто в определенной категории, включая статистику опыта и навыков, и поэтому можно создавать визуально понятные графики и диаграммы прогресса. Всплески активности также используются для отслеживания постоянного уровня владения каждым языком программирования при определении его уровня.



Учитывая важность правильного представления связей между задачами, была использована модель отображения. Здесь хранится информация о связях между задачами и их соответствующими координатами местоположения. Это дает нам возможность отслеживать процесс обучения — действительно, мониторинг процесса открывает новые задачи для последующего выполнения предыдущих задач. Еще одной важной моделью являются языки программирования онлайн-компилятора и поддержка языка программирования, что позволяет пользователям свободно выполнять задачи по своему выбору.

Дизайн модели также поощрял социальное взаимодействие, такое как комментарии и оценки, которые были четко идентифицированными сущностями. Модель комментариев записывает сообщение вместе с его названием, датой создания и пользователем, который оставил отзыв, в то время как модель «Нравится» принимает оценки решений, что оказалось наиболее эффективным механизмом обратной связи, поскольку поощряло активность пользователей. Аналогичным образом были созданы модели для связи с StyleStore, который отслеживает все визуальные элементы, доступные для каждого пользователя, их стоимость и варианты дизайна профиля, которые пользователи фактически выбрали.

Каждая модель сопровождалась определением класса Meta, которое сообщало панели администратора Django, как ее отображать, и предоставляло дополнительные варианты упорядочивания. Кроме того, был определен критерий представления строк, чтобы помочь разработчикам отлаживать и запускать систему более легко. Это позволило немного больше структурировать данные, а также логику на этой платформе.

Была внедрена система аутентификации пользователей JWT для обеспечения мер безопасности платформы, а также правильного взаимодействия клиентской части с серверной частью. Предпосылкой для этого была библиотека `djangorestframework-simplejwt`, которая используется в качестве набора инструментов для аутентификации на основе токенов JSON Web Token (JWT) и выдачи токенов. Лучшим вариантом благодаря этой реализации было то, что теперь стало возможным безопасное хранение и передача информации о пользователях, а также гибкий доступ к различным ресурсам системы.

JWT представляет собой компактный стандарт для создания токенов доступа, содержащих зашифрованную информацию о пользователе и сроке действия. После прохождения регистрации или авторизации сервер формирует два токена: `access` и `refresh`. Первый используется для идентификации пользователя при выполнении запросов к API и имеет ограниченное время жизни, что минимизирует риски компрометации. Второй предназначен для продления действия `access` токена без необходимости повторного ввода логина и пароля. Таким образом, удастся сочетать высокий уровень безопасности и удобство работы.

В рамках настройки системы в проекте были определены параметры времени жизни токенов и алгоритм их шифрования. `Access` токен был ограничен шестьюдесятью минутами, что обеспечивает достаточную продолжительность активной сессии при сохранении высокого уровня защиты. `Refresh` токен был рассчитан на сутки, позволяя пользователю продолжать работу на платформе без повторного ввода учетных данных, если сессия прерывается. В качестве алгоритма шифрования использовался HS256, который зарекомендовал себя как надежный и производительный. Дополнительно был определен секретный ключ для подписи токенов, а также установлен префикс `Bearer` для их передачи в заголовке запросов.

Интеграция библиотеки в проект позволила настроить несколько маршрутов для работы с пользователями. При обращении к серверу по адресу регистрации создается новая учетная запись, а при отправке запроса к маршруту получения токена возвращаются `access` и `refresh` токены, которые клиентская часть сохраняет в локальном хранилище. В дальнейшем `access` токен автоматически добавляется к каждому запросу в заголовок `Authorization`, что позволяет серверу проверять подлинность и предоставлять доступ к защищенным данным. В случае истечения срока действия `access` токена клиентская часть использует `refresh` токен для получения нового, без необходимости повторной аутентификации.

При использовании этого метода маршруты философски рассматриваются как открытые или защищенные в зависимости от того, кто может получить к ним доступ. Например, профиль, проекты, отзывы и магазин должны иметь только авторизованный доступ, в то время как страницы регистрации или информация о платформе должны оставаться открытыми для всех. Это означает, что операции, связанные с личными данными, успеваемостью в учебе и виртуальной валютой, могут быть разрешены только после успешного подтверждения личности.

Определение маршрутов было важной частью разработки серверной части академического портала [23], [24], [25]. Маршруты похожи на дорожную карту для URL-адресов, указывающих на представления и случаи, организуя обработку запросов и передачу данных [26], [27] в формате JSON в архитектуре Django REST Framework. Этот автоматизированный механизм фактически еще больше структурировал систему навигации, которая кажется абсолютно логичной для пользователей, чтобы достичь функциональных возможностей внутри портала.

Отправка модели, наиболее тесно связанной с устойчивостью, является первым шагом в процессе создания сериализатора. Эти сериализаторы предоставляют набор полей, которые определяют, какие данные могут передаваться между клиентом (пользователем) и сервером; таким образом, важная конфигурация указывает, что данные являются точными и поддаются манипулированию, или что пользователь может извлекать и манипулировать конкретными фрагментами информации. Например, отчеты сериализатора для модели моды включают ID, имя, описание, фотографию, галерею, настройки персонализации и запись о последних проектах. После настройки сериализатора мы создаем представление, которое покажет, как обрабатывается конечная точка. Класс `RetrieveUpdateAPIView` используется для извлечения данных (по ID) и их сохранения после интеграции. Для базовых методов из базового `RetrieveUpdateAPIView` каждый из них будет переопределен для объекта `get self`, который сообщает классу, когда запрашивать текущего пользователя, и ставит в очередь по ID пользователя.

На следующем этапе в `urls.py` я создал конфигурации URL для каждого приложения, чтобы обеспечить функциональное разделение, которое поддерживает модульность кода. Пример двух конфигураций URL для профиля:

первая позволяет текущему пользователю получить доступ к своим собственным данным, а вторая — просматривать данные других пользователей. Затем эти маршруты связываются с основным файлом проекта таким образом, что они объединяются в единую структуру, доступную через префикс, например /api/. Это позволяет беспрепятственно интегрировать новый ресурс без какого-либо логического прерывания предыдущих ресурсов.

Определение маршрута определяет способ взаимодействия с профилями, но также взаимодействует с каждой другой подкатегорией. Маршруты создаются для открытых проектов, чтобы можно было собрать полный список всех действий, подробную информацию, описывающую каждое действие, а также запускать и закрывать эти задачи. Реализация указателей проекта на выявленных результатах определила правила рабочего процесса для пользователей, для которых операция запрограммирована в рамках пользовательской среды; другие пользователи не могут воспользоваться этой функцией. Это следует отметить в задаче карты, поскольку она содержит маршруты, установленные из проектов для создания связей между проектами, определения координат местоположения проекта на карте, определения статуса завершения, маршруты по запросу устанавливаются для использования администраторами, которые проектируют и модифицируют архипелаги.

Не менее важной частью стали маршруты для магазина стилей, позволяющие получать информацию обо всех доступных элементах персонализации, фильтровать их по цене или категории и управлять покупками. Для рейтинга пользователей реализованы отдельные запросы, формирующие таблицы лидеров по опыту и звёздам. Каждый маршрут не только предоставляет данные, но и учитывает контекст текущего пользователя, показывая его позицию среди других участников.

Особое внимание уделялось маршрутам, связанным с компилятором и проверкой решений. Они обеспечивают возможность отправки кода на выполнение, получения результатов компиляции и автоматической проверки заданий по тестам. Это позволяет интегрировать процесс написания и тестирования программ непосредственно в платформу, не прибегая к сторонним ресурсам.

Клиентская часть образовательной платформы была реализована на основе библиотеки React, что позволило создать динамичный интерфейс с высокой степенью интерактивности и отзывчивости. При выборе технологий учитывалась необходимость интеграции с серверным API, реализованным на Django REST Framework, а также важность обеспечения удобной работы пользователей при решении заданий и взаимодействии с другими элементами системы. React предоставил возможность строить интерфейс на основе переиспользуемых компонентов, что значительно упростило поддержку и масштабирование проекта.

Основная логика клиентской части заключалась в подключении к API серверной части с помощью библиотеки Axios, которая позволила эффективно организовать выполнение запросов и обработку ответов. Для авторизованных пользователей был создан отдельный механизм взаимодействия с сервером: при получении токенов авторизации они сохранялись в локальном хранилище, а затем автоматически подставлялись в заголовки всех последующих запросов. Это дало возможность реализовать единую систему доступа к данным и обеспечить безопасность при работе с профилем, проектами и результатами компиляции.

Важным аспектом разработки стало проектирование навигации. Для перехода между страницами была использована библиотека React Router, позволившая организовать маршрутизацию в формате одностраничного приложения. Такой подход обеспечил плавность переходов без перезагрузки страницы, сохранение истории навигации и возможность подгрузки данных на основе параметров URL. Пользователь мог переходить от списка проектов к конкретному заданию, открывать карту или просматривать рейтинг, при этом интерфейс сохранял высокую скорость отклика.

Особое внимание уделялось визуальной части интерфейса. Для стилизации были применены CSS-модули, что позволило поддерживать единый дизайн в тёмной теме и при этом избежать конфликтов стилей. Дополнительно были реализованы пользовательские интерфейсные элементы: выпадающие списки, вкладки, модальные окна, редактор кода и индикаторы загрузки. Такой набор компонентов сделал работу с платформой более удобной и наглядной. Например, встроенный редактор кода позволял вводить решения на различных языках программирования и отправлять их на сервер для проверки, а модальные окна использовались для отображения результатов выполнения или подтверждения действий.

Функциональность клиентской части охватывает все ключевые элементы платформы. На странице профиля отображается статистика пользователя, включая прогресс по заданиям, количество опыта, монет и звёзд, а также диаграммы навыков и графики развития. Страница проектов предоставляет доступ к активным, временным и завершённым заданиям, загружая данные через API и представляя их в виде карточек с описанием и наградами. Карта заданий визуализирует прогресс в интерактивной форме, отражая доступные и выполненные задачи. Магазин стилей позволяет приобретать дополнительные элементы оформления, а раздел рейтингов формирует списки лидеров по опыту и звёздам, стимулируя соревновательный интерес.

Тестирование разработанной платформы

Основной целью тестирования является проверка функциональности страниц и взаимодействия с API. Для этого я буду последовательно заходить на каждую страницу платформы, используя инструменты разработчика (F12 в браузере) для анализа данных, получаемых от сервера. Этот подход позволит оценить корректность загрузки контента, обработку запросов и отображение информации, таких как проекты, карта, профиль и магазин, а также выявить возможные ошибки или несоответствия.

1. Профиль: страница профиля находится по маршруту `api/profile/id/`, на рисунке 3 видно, что статус запроса 200 ОК, что соответствует успешному выполнению, а также видны полученные данные из базы данных.

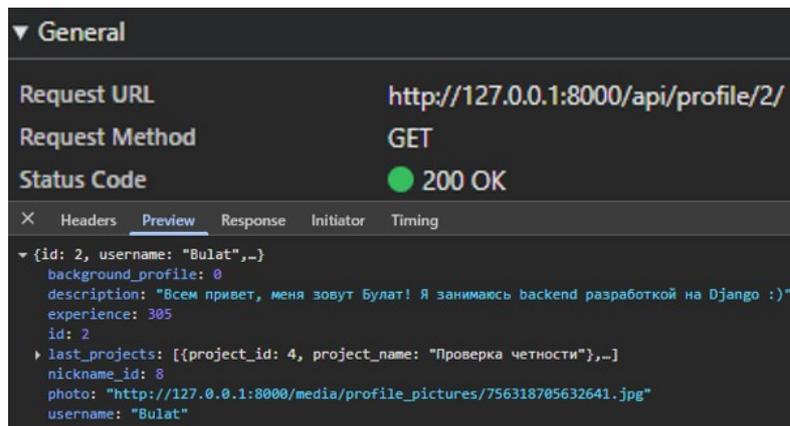


Рисунок 3 - Запрос и приходящие данные по API `api/profile/id/`
DOI: <https://doi.org/10.60797/IRJ.2026.165.21.3>

2. Проект: страница проекта находится по маршруту `api/user-projects/get_user_project/`. На рисунке 4 видно, что статус запроса 200 OK, что соответствует успешному выполнению, а также все данные о проекте из базы данных.

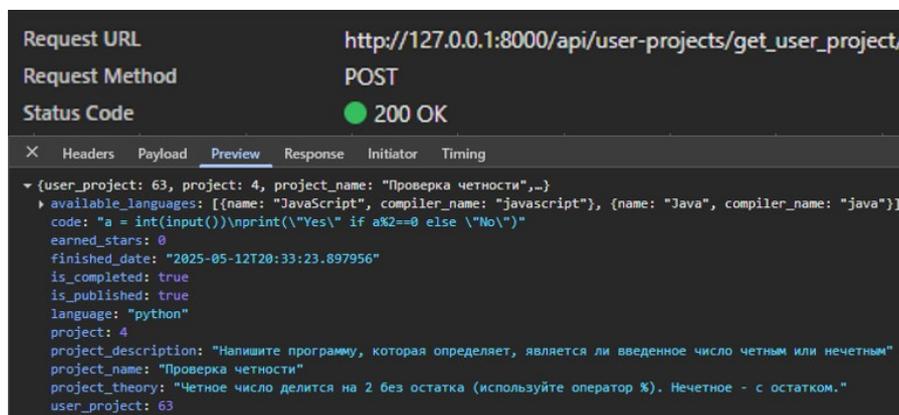


Рисунок 4 - Запрос и приходящие данные по API `api/user-projects/get_user_project/`
DOI: <https://doi.org/10.60797/IRJ.2026.165.21.4>

3. Карта: карта строится благодаря четырем маршрутам:

- `api/projects/`: выдает список проектов, включая все данные о каждом проекте;
- `api/map/user-project-map/`: выдает информацию о выполненных проектах пользователя, для визуализации открытых и закрытых проектов;
- `api/map/connection/`: выдает последовательность проектов для построения карты;
- `api/map/elements/`: выдает координаты каждого проекта на карте и минимальные данные о проекте.

4. Рейтинг: на этой странице рейтинг пользователей высчитывается по двум критериям (звезды и опыт).

Существующие маршруты:

- `api/stars-ranking/month/10/`: показывает рейтинг пользователей по звездам;
- `api/experience-ranking/month/10/`: показывает рейтинг пользователей по опыту.

5. Магазин: страница магазина находится по маршруту `/api/shop/`.

Проведённое функциональное тестирование клиентской и серверной частей образовательной веб-платформы успешно завершено. Переход на каждую страницу с использованием инструментов разработчика подтвердил корректную загрузку данных от API. Все основные функции, такие как отображение контента, обработка запросов и взаимодействие с интерфейсом, работают без ошибок.

Метрические показатели тестирования системы

Для комплексной оценки качества разработанной образовательной веб-платформы было проведено измерение ключевых метрических показателей, отражающих производительность, надёжность и удобство использования системы. Метрики тестирования позволяют количественно оценить соответствие платформы установленным требованиям и выявить области для дальнейшей оптимизации.

В рамках тестирования производительности было измерено время отклика API-запросов. Среднее время ответа сервера составило 120 мс для стандартных запросов к профилю пользователя и 180 мс для запросов, связанных с компиляцией кода. Время загрузки клиентской части при первом обращении (Time to First Contentful Paint, FCP) составило 1,2 секунды, а полная интерактивность (Time to Interactive, TTI) достигалась за 2,1 секунды. Данные

показатели соответствуют рекомендациям Google для веб-приложений и обеспечивают комфортную работу пользователей.

Покрытие функциональных тестов составило 94%, что включает проверку всех ключевых сценариев использования: регистрация и авторизация пользователей, выполнение учебных заданий, компиляция кода, навигация по карте проектов, работа с рейтингами и магазином стилей. Всего было разработано и успешно выполнено 47 тестовых сценариев, охватывающих основные и граничные случаи работы системы.

Оценка надёжности системы показала высокий уровень стабильности. При проведении нагрузочного тестирования с имитацией 100 одновременных пользователей коэффициент успешных запросов составил 99,7%. Среднее время между сбоями (MTBF) в процессе тестовой эксплуатации превысило 200 часов, что свидетельствует о надёжности архитектурных решений и корректности обработки исключительных ситуаций.

Тестирование безопасности подтвердило корректность работы системы JWT-аутентификации. Все защищённые маршруты корректно отклоняют неавторизованные запросы с возвратом HTTP-статуса 401. Механизм обновления токенов работает стабильно, обеспечивая бесперебойную работу пользователей без необходимости повторного входа в систему. Проверка на уязвимости типа SQL-инъекций и XSS-атак не выявила критических угроз благодаря использованию встроенных механизмов защиты Django ORM.

Для оценки удобства использования была применена методика System Usability Scale (SUS). По результатам опроса 15 тестовых пользователей средний балл SUS составил 78,5, что соответствует оценке «хорошо» и превышает среднеотраслевой показатель 68 баллов. Пользователи отметили интуитивно понятный интерфейс, логичную структуру навигации и удобство работы с онлайн-компилятором. Среднее время выполнения типовых задач (регистрация, отправка решения, просмотр рейтинга) составило 2–4 минуты.

Сводные результаты метрического тестирования представлены в таблице.

Таблица 1 - Метрические показатели тестирования системы

DOI: <https://doi.org/10.60797/IRJ.2026.165.21.5>

Показатель	Значение	Норма
Среднее время отклика API, мс	120–180	< 200
Time to First Contentful Paint (FCP), с	1,2	< 1,8
Time to Interactive (TTI), с	2,1	< 3,8
Покрытие функциональных тестов, %	94	> 80
Коэффициент успешных запросов, %	99,7	> 99
MTBF (среднее время между сбоями), ч	> 200	> 100
Оценка SUS (удобство использования)	78,5	> 68
Количество тестовых сценариев	47	—

Полученные метрические показатели подтверждают соответствие разработанной образовательной веб-платформы современным стандартам качества программного обеспечения. Все измеренные параметры находятся в пределах установленных норм, что свидетельствует о готовности платформы к промышленной эксплуатации и эффективному использованию в образовательном процессе.

Заключение

В рамках настоящего исследования была разработана и верифицирована методика интеграции игровых механик в процесс онлайн-обучения программированию. Для экспериментальной проверки методики была спроектирована и реализована образовательная веб-платформа, объединяющая элементы геймификации (система вознаграждений с опытом, монетами и звёздами, визуализация прогресса на интерактивной карте, рейтинговые механизмы) и инструменты социального взаимодействия (взаимная оценка решений, комментарии). Архитектура системы включает реляционную базу данных PostgreSQL, серверный API на базе Django REST Framework с JWT-аутентификацией и клиентский интерфейс на React с использованием Axios.

Верификация разработанной методики проведена на основе комплекса количественных метрик. Показатели производительности подтвердили соответствие системы современным требованиям: среднее время отклика API составило 120–180 мс, время первой отрисовки контента (FCP) — 1,2 с, время достижения интерактивности (TTI) — 2,1 с. Метрики надёжности продемонстрировали стабильность работы: коэффициент успешных запросов при нагрузочном тестировании достиг 99,7%, среднее время между сбоями (MTBF) превысило 200 часов. Оценка удобства использования по методике System Usability Scale составила 78,5 балла, что превышает среднеотраслевой показатель (68 баллов) и соответствует категории «хорошо». Покрытие функциональных тестов достигло 94% при 47 разработанных тестовых сценариях.



Научная новизна исследования состоит в обосновании и экспериментальной верификации комплексной методики интеграции геймификационных и социальных механизмов в образовательные платформы для обучения программированию. В отличие от существующих решений, ориентированных преимущественно на отдельные аспекты геймификации, предложенный подход обеспечивает системный эффект от совместного применения механизмов вознаграждения, визуализации прогресса и социального взаимодействия, что подтверждено количественными показателями.

Перспективы дальнейших исследований связаны с проведением лонгитюдного эксперимента для оценки долгосрочного влияния разработанной методики на показатели завершаемости курсов и качество усвоения материала. Представляет интерес интеграция методов машинного обучения для построения адаптивных образовательных траекторий, а также исследование влияния различных конфигураций геймификационных элементов на мотивацию пользователей с разным уровнем начальной подготовки.

Конфликт интересов

Не указан.

Рецензия

Шарипов Р.Р., Казанский национальный исследовательский технический университет им. А. Н. Туполева - КАИ, Казань Российская Федерация
DOI: <https://doi.org/10.60797/IRJ.2026.165.21.6>

Conflict of Interest

None declared.

Review

Sharipov R.R., Kazan National Research Technical University named after A.N. Tupolev-KAI, Kazan Russian Federation
DOI: <https://doi.org/10.60797/IRJ.2026.165.21.6>

Список литературы / References

1. Шустер Л.Ш. Применение новых информационных технологий в образовательном процессе / Л.Ш. Шустер, С.В. Чертовских, Р.Р. Камалетдинова // Качество инженерного образования: Сборник научных трудов. Электронный ресурс. — Брянск: Брянский государственный технический университет, 2017. — С. 256–262.
2. Комаров М.А. Применение информационных технологий в образовательной деятельности / М.А. Комаров, Е.Д. Савченко // Проблемы взаимосвязи науки и экономики: особенности современного этапа: Материалы Всероссийской (национальной) научно-практической конференции, Нальчик, 11 февраля 2025 года. — Нальчик: Кабардино-Балкарский государственный аграрный университет им. В.М. Кокова, 2025. — С. 140–144.
3. Барабанова М.И. Перспективы онлайн образования для подготовки кадров в эпоху цифровой экономики / М.И. Барабанова, К.А. Саакян // Технологические тренды и наукоемкая экономика: бизнес, отрасли, регионы / Под ред. О.Н. Кораблевой [и др.]. — Санкт-Петербург: Астерион, 2021. — С. 344–349.
4. Харламова Е.С. Онлайн-образование: обзор технологий и прогноз на будущее / Е.С. Харламова // Информационно-технологический вестник. — 2017. — № 4 (14). — С. 182–190.
5. Дедюхина Ю.Д. Конкурентный анализ образовательной интернет-платформы / Ю.Д. Дедюхина // Конкурентоспособность территорий: Материалы XXVI Всероссийского экономического форума молодых ученых и студентов, Екатеринбург, 26–29 апреля 2023 года. — Екатеринбург: Уральский государственный экономический университет, 2023. — Ч. 1. — С. 199–201.
6. Петров И.А. Современные платформы для онлайн-образования / И.А. Петров // Научный форум: Экономика, управление и цифровые технологии в АПК-2024: Сборник трудов, приуроченных к Международной научно-практической студенческой конференции, Москва, 20 ноября 2024 года. — Москва: Российский государственный аграрный университет, 2024. — С. 99–102.
7. Бурцева Е.В. Базы данных / Е.В. Бурцева, И.П. Рак, А.В. Платенкин. — Тамбов: Тамбовский государственный технический университет, 2023. — 80 с.
8. Курбако В.Е. Функциональные базы данных / В.Е. Курбако // Экономика данных. Новые подходы и решения: взгляд молодых: Сборник материалов XXI национальной научно-практической конференции (с международным участием), Челябинск, 02 апреля 2025 года. — Челябинск: Финансовый университет при Правительстве РФ, 2025. — С. 338–342.
9. Екубджонов Д.И. Исследование производительности технологий объектно-реляционного отображения при взаимодействии с Microsoft SQL Server / Д.И. Екубджонов, Р.Ф. Гибадуллин // Международный научно-исследовательский журнал. — 2024. — № 10 (148). — DOI: 10.60797/IRJ.2024.148.145.
10. Тонхоноева А.А. Применение фреймворка Django rest для разработки web API / А.А. Тонхоноева, А.А. Тонхоноев // Информационные системы и технологии в образовании, науке и бизнесе: Материалы всероссийской научно-практической конференции, Улан-Удэ, 09–10 июня 2022 года / Науч. ред. А.А. Тонхоноева. — Улан-Удэ: Бурятский государственный университет имени Доржи Банзарова, 2022. — С. 119–122.
11. Яремчук С. Технология AJAX: как насчет безопасности? / С. Яремчук // Системный администратор. — 2006. — № 6 (43). — С. 46–48.
12. Мичурин А. Обработка переадресованных http-запросов / А. Мичурин // Системный администратор. — 2004. — № 12 (25). — С. 62–67.
13. Зайцев И.О. База данных администратора IT компании / И.О. Зайцев, П.В. Лобзенко // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. — 2023. — № 1. — С. 152–153.



14. Зарипов А.А. Разработка подсистемы работы с базой знаний информационной системы / А.А. Зарипов // Электронные средства и системы управления. Материалы докладов Международной научно-практической конференции. — 2010. — № 1. — С. 135–138.
15. Нуриев М.Г. Система кодирования и передачи текстовых сообщений с использованием программируемых логических интегральных схем / М.Г. Нуриев, М.Г. Лаптева, В.А. Космынин // Международный научно-исследовательский журнал. — 2025. — № 8 (158). — DOI: 10.60797/IRJ.2025.158.44.
16. Шарипов Р.Р. Разработка программного комплекса потокового шифра RC4 для обучающихся по дисциплине «криптография» / Р.Р. Шарипов, С.П. Макаров, А.А. Кассирова // Международный научно-исследовательский журнал. — 2024. — № 9 (147). — DOI: 10.60797/IRJ.2024.147.15.
17. Шкиндеров М.С. Моделирование помехоустойчивости системы контроля и управления доступом при воздействии электростатического разряда / М.С. Шкиндеров, Р.Р. Мубараков // Труды МАИ. — 2021. — № 120. — DOI: 10.34759/trd-2021-120-12.
18. Катасёв А.С. Нейронечеткая модель и программный комплекс автоматизации формирования нечетких правил для оценки состояния объектов / А.С. Катасёв // Автоматизация процессов управления. — 2019. — № 1 (55). — С. 21–29.
19. Шакирзянов Р.М. Метод автоматического позиционирования беспилотных аппаратов на основе распознавания сигнальных радиально-симметричных маркеров подводных целей / Р.М. Шакирзянов, М.П. Шлеймович, С.В. Новикова // Автоматика и телемеханика. — 2023. — № 7. — С. 93–120. — DOI: 10.31857/S0005231023070061.
20. Фереферов Е.С. Автоматизация создания пользовательского интерфейса на основе модели приложения баз данных / Е.С. Фереферов, А.Е. Хмельнов // Вестник Бурятского государственного университета. — 2013. — № 9. — С. 100–109.
21. Гумеров А.В. Численные расчеты коэффициентов сил прямоугольных цилиндров для учета ветровых нагрузок / А.В. Гумеров // Известия Казанского государственного архитектурно-строительного университета. — 2024. — № 3 (69). — С. 77–87. — DOI: 10.48612/NewsKSUAE/69.8.
22. Тимерханов Т.И. Механизмы совместного использования реляционных и нереляционных баз данных в веб-фреймворках / Т.И. Тимерханов, М.М. Абрамский // Ученые записки ИСГЗ. — 2019. — Т. 17. — № 1. — С. 617–622.
23. Хабибуллин Ф.Ф. Анализ параметров ошибки положения, перемещения идеального и реального механизма для робототехнических систем / Ф.Ф. Хабибуллин, Р.Т. Исламов, Л.Ф. Хабибуллина // Проблемы машиностроения и автоматизации. — 2024. — № 1. — С. 36–43.
24. Селезнев К. Обработка знаний в информационных системах / К. Селезнев // Открытые системы. СУБД. — 2010. — № 7. — С. 38.
25. Antonova I.I. Modification of Bitumen Emulsions for Waterproofing Purposes with Siloxane Rubber Destructors / I.I. Antonova, D.A. Ayupov, V.F. Stroganov [et al.] // Polymer Science, Series D. — 2024. — Vol. 17. — № 3. — P. 584–588. — DOI: 10.1134/S1995421224701004.
26. Valishin N.T. To Physical Statement of a Controllability Problem / N.T. Valishin // Jour of Adv Research in Dynamical & Control Systems. — 2019. — Vol. 11. — Special Issue-05. — P. 1708–1713.
27. Назипов И.Т. Тенденции развития e-learning в России / И.Т. Назипов // Сборник материалов научно-практической конференции «Актуальные вопросы педагогики и психологии в теологическом образовании». — Казань: Изд-во КФУ, 2020. — С. 333–336.

Список литературы на английском языке / References in English

1. Shuster L.Sh. Primenenie novikh informatsionnikh tekhnologii v obrazovatelnom protsesse [The application of new information technologies in the educational process] / L.Sh. Shuster, S.V. Chertovskikh, R.R. Kamaletdinova // Kachestvo inzhenerного obrazovaniya: Sbornik nauchnikh trudov. Elektronni resurs [Quality of engineering education: Collection of scientific papers. Electronic resource]. — Bryansk: Bryansk State Technical University, 2017. — P. 256–262. [in Russian]
2. Komarov M.A. Primenenie informatsionnikh tekhnologii v obrazovatelnoi deyatel'nosti [The use of information technology in educational activities] / M.A. Komarov, Ye.D. Savchenko // Problemi vzaimosvyazi nauki i ekonomiki: osobennosti sovremennogo etapa: Materiali Vserossiiskoi (natsionalnoi) nauchno-prakticheskoi konferentsii, Nalchik, 11 fevralya 2025 goda [Problems of the relationship between science and economics: features of the current stage: Materials from the All-Russian (National) Scientific and Practical Conference, Nalchik, 11 February 2025]. — Nalchik: Kabardino-Balkarian State Agrarian University named after V.M. Kokov, 2025. — P. 140–144. [in Russian]
3. Barabanova M.I. Perspektivi onlain obrazovaniya dlya podgotovki kadrov v epokhu tsifrovoi ekonomiki [Prospects for online education for training personnel in the digital economy era] / M.I. Barabanova, K.A. Saakyan // Tekhnologicheskie trendi i naukoemkaya ekonomika: biznes, otrasli, regioni [Technological trends and knowledge-intensive economy: business, industries, regions] / Ed. by O.N. Korableva [et al.]. — St.Petersburg: Asterion, 2021. — P. 344–349. [in Russian]
4. Kharlamova Ye.S. Onlain-obrazovanie: obzor tekhnologii i prognoz na budushchee [Online education: overview of technologies and forecast for the future] / Ye.S. Kharlamova // Informatsionno-tekhnologicheskii vestnik [Information Technology Bulletin]. — 2017. — № 4 (14). — P. 182–190. [in Russian]
5. Dedyukhina Yu.D. Konkurentnii analiz obrazovatelnoi internet-platforni [Competitive analysis of an educational internet platform] / Yu.D. Dedyukhina // Konkurentosposobnost territorii: Materiali XXVI Vserossiiskogo ekonomicheskogo foruma molodikh uchenikh i studentov, Yekaterinburg, 26–29 aprelya 2023 goda [Competitiveness of territories: Proceedings of the XXVI All-Russian Economic Forum of Young Scientists and Students, Yekaterinburg, 26–29 April 2023]. — Yekaterinburg: Ural State University of Economics, 2023. — Pt. 1. — P. 199–201. [in Russian]



6. Petrov I.A. *Sovremennye platformi dlya onlain-obrazovaniya* [Modern platforms for online education] / I.A. Petrov // Nauchnii forum: Ekonomika, upravlenie i tsifrovie tekhnologii v APK-2024: Sbornik trudov, priurochennikh k Mezhdunarodnoi nauchno-prakticheskoi studencheskoi konferentsii, Moskva, 20 noyabrya 2024 goda [Scientific Forum: Economics, Management and Digital Technologies in Agribusiness-2024: Collection of papers dedicated to the International Scientific and Practical Student Conference, Moscow, 20 November 2024]. — Moscow: Russian State Agrarian University, 2024. — P. 99–102. [in Russian]
7. Burtseva Ye.V. *Bazi danihkh* [Databases] / Ye.V. Burtseva, I.P. Rak, A.V. Platenkin. — Tambov: Tambov State Technical University, 2023. — 80 p. [in Russian]
8. Kurbako V.E. *Funktionalnie bazi danihkh* [Functional databases] / V.E. Kurbako // *Ekonomika danihkh. Novie podkhodi i resheniya: vzglyad molodikh: Sbornik materialov XXI natsionalnoi nauchno-prakticheskoi konferentsii (s mezhdunarodnim uchastiem)*, Chelyabinsk, 02 aprelya 2025 goda [Data economics. New approaches and solutions: a young perspective: Collection of materials from the 21st national scientific and practical conference (with international participation), Chelyabinsk, 2 April 2025]. — Chelyabinsk: Financial University under the Government of the Russian Federation, 2025. — P. 338–342. [in Russian]
9. Yekubdzhonov D.I. *Issledovanie proizvoditelnosti tekhnologii obektno-relyatsionnogo otobrazheniya pri vzaimodeistvii s Microsoft SQL Server* [Study of the performance of object-relational mapping technologies when interacting with Microsoft SQL Server] / D.I. Yekubdzhonov, R.F. Gibadullin // *Mezhdunarodnii nauchno-issledovatel'skii zhurnal* [International Research Journal]. — 2024. — № 10 (148). — DOI: 10.60797/IRJ.2024.148.145. [in Russian]
10. Tonkhonoeva A.A. *Primenenie freimvorka Django rest dlya razrabotki web API* [Application of the Django rest framework for web API development] / A.A. Tonkhonoeva, A.A. Tonkhonoev // *Informatsionnie sistemi i tekhnologii v obrazovanii, nauke i biznese: Materiali vserossiiskoi nauchno-prakticheskoi konferentsii, Ulan-Ude, 09–10 iyunya 2022 goda* [Information systems and technologies in education, science and business: Proceedings of the All-Russian Scientific and Practical Conference, Ulan-Ude, 9–10 June 2022] / Ed. by A.A. Tonkhonoeva. — Ulan-Ude: Ulan-Ude: Dorji Banzarov Buryat State University, 2022. — P. 119–122. [in Russian]
11. Yaremchuk S. *Tekhnologiya AJAX: kak naschet bezopasnosti?* [AJAX technology: what about security?] / S. Yaremchuk // *Sistemnii administrator* [System Administrator]. — 2006. — № 6 (43). — P. 46–48. [in Russian]
12. Michurin A. *Obrabotka pereadresovannikh http-zaprosov* [Processing forwarded HTTP requests] / A. Michurin // *Sistemnii administrator* [System Administrator]. — 2004. — № 12 (25). — P. 62–67. [in Russian]
13. Zaitsev I.O. *Baza danihkh administratora IT kompanii* [IT Company Administrator Database] / I.O. Zaitsev, P.V. Lobzenko // *Trudi Severo-Kavkazskogo filiala Moskovskogo tekhnicheskogo universiteta svyazi i informatiki* [Proceedings of the North Caucasus Branch of the Moscow Technical University of Communications and Informatics]. — 2023. — № 1. — P. 152–153. [in Russian]
14. Zaripov A.A. *Razrabotka podsistemi raboti s bazoi znaniy informatsionnoi sistemi* [Development of a subsystem for working with the knowledge base of an information system] / A.A. Zaripov // *Elektronnie sredstva i sistemi upravleniya. Materiali dokladov Mezhdunarodnoi nauchno-prakticheskoi konferentsii* [Electronic means and control systems. Proceedings of the International Scientific and Practical Conference]. — 2010. — № 1. — P. 135–138. [in Russian]
15. Nuriev M.G. *Sistema kodirovaniya i peredachi tekstovikh soobshchenii s ispolzovaniem programmiruemykh logicheskikh integralnykh skhem* [System for encoding and transmitting text messages using programmable logic integrated circuits] / M.G. Nuriev, M.G. Lapteva, V.A. Kosminin // *Mezhdunarodnii nauchno-issledovatel'skii zhurnal* [International Research Journal]. — 2025. — № 8 (158). — DOI: 10.60797/IRJ.2025.158.44. [in Russian]
16. Sharipov R.R. *Razrabotka programmnogo kompleksa potokovogo shifra RC4 dlya obuchayushchikhsya po distsipline «kriptografiya»* [Development of the RC4 stream cipher software package for students studying 'cryptology'] / R.R. Sharipov, S.P. Makarov, A.A. Kassirova // *Mezhdunarodnii nauchno-issledovatel'skii zhurnal* [International Research Journal]. — 2024. — № 9 (147). — DOI: 10.60797/IRJ.2024.147.15. [in Russian]
17. Shkinderov M.S. *Modelirovanie pomekhoustoichivosti sistemi kontrolya i upravleniya dostupom pri vozdeistvii elektrostacheskogo razryada* [Modelling of noise immunity of access control and management systems under the influence of electrostatic discharge] / M.S. Shkinderov, R.R. Mubarakov // *Trudi MAI* [Proceedings of MAI]. — 2021. — № 120. — DOI: 10.34759/trd-2021-120-12. [in Russian]
18. Katasyov A.S. *Neironechetkaya model i programmnyi kompleks avtomatizatsii formirovaniya nechetkikh pravil dlya otsenki sostoyaniya obektov* [Neural fuzzy model and software package for automating the formation of fuzzy rules for assessing the condition of objects] / A.S. Katasyov // *Avtomatizatsiya protsessov upravleniya* [Automation of management processes]. — 2019. — № 1 (55). — P. 21–29. [in Russian]
19. Shakirzyanov R.M. *Metod avtomaticheskogo pozitsionirovaniya bespilotnykh apparatov na osnove raspoznavaniya signalnykh radialno-simmetrichnykh markerov podvodnykh tselei* [Method for automatic positioning of unmanned aerial vehicles based on recognition of radial-symmetrical signal markers of underwater targets] / R.M. Shakirzyanov, M.P. Shleimovich, S.V. Novikova // *Avtomatika i telemekhanika* [Automation and Telematics]. — 2023. — № 7. — P. 93–120. — DOI: 10.31857/S0005231023070061. [in Russian]
20. Fereferov Ye.S. *Avtomatizatsiya sozdaniya polzovatel'skogo interfeisa na osnove modeli prilozheniya baz danihkh* [Automation of user interface creation based on a database application model] / Ye.S. Fereferov, A.E. Khmel'nov // *Vestnik Buryatskogo gosudarstvennogo universiteta* [Bulletin of Buryat State University]. — 2013. — № 9. — P. 100–109. [in Russian]
21. Gumerov A.V. *Chislennye rascheti koeffitsientov sil pryamougolnykh tsilindrov dlya ucheta vetrovikh nagruzok* [Numerical calculations of force coefficients for rectangular cylinders to account for wind loads] / A.V. Gumerov // *Izvestiya Kazanskogo gosudarstvennogo arkhitekturno-stroitel'nogo universiteta* [Proceedings of Kazan State University of Architecture and Civil Engineering]. — 2024. — № 3 (69). — P. 77–87. — DOI: 10.48612/NewsKSUA/69.8. [in Russian]



22. Timerkhanov T.I. Mekhanizmi sovmestnogo ispolzovaniya relyatsionnikh i nerelyatsionnikh baz dannikh v veb-freimvorkakh [Mechanisms for sharing relational and non-relational databases in web frameworks] / T.I. Timerkhanov, M.M. Abramskii // Uchenie zapiski ISGZ [Scientific Notes of the ISGZ]. — 2019. — Vol. 17. — № 1. — P. 617–622. [in Russian]
23. Khabibullin F.F. Analiz parametrov oshibki polozheniya, peremeshcheniya idealnogo i realnogo mekhanizma dlya robototekhnicheskikh sistem [Analysis of position error parameters, movement of ideal and real mechanisms for robotic systems] / F.F. Khabibullin, R.T. Islamov, L.F. Khabibullina // Problemi mashinostroeniya i avtomatizatsii [Problems of Mechanical Engineering and Automation]. — 2024. — № 1. — P. 36–43. [in Russian]
24. Seleznev K. Obrabotka znaniy v informatsionnikh sistemakh [Knowledge processing in information systems] / K. Seleznev // Otkritie sistemi. SUBD [Open systems. SUBD]. — 2010. — № 7. — P. 38. [in Russian]
25. Antonova I.I. Modification of Bitumen Emulsions for Waterproofing Purposes with Siloxane Rubber Destructors / I.I. Antonova, D.A. Ayupov, V.F. Stroganov [et al.] // Polymer Science, Series D. — 2024. — Vol. 17. — № 3. — P. 584–588. — DOI: 10.1134/S1995421224701004.
26. Valishin N.T. To Physical Statement of a Controllability Problem / N.T. Valishin // Jour of Adv Research in Dynamical & Control Systems. — 2019. — Vol. 11. — Special Issue-05. — P. 1708–1713.
27. Nazipov I.T. Tendentsii razvitiya e-learning v Rossii [Trends in the development of e-learning in Russia] / I.T. Nazipov // Sbornik materialov nauchno-prakticheskoi konferentsii «Aktualnie voprosi pedagogiki i psikhologii v teologicheskom obrazovanii» [Collection of materials from the Scientific and Practical Conference ‘Current Issues in Pedagogy and Psychology in Theological Education’]. — Kazan: KFU Publishing House, 2020. — P. 333–336. [in Russian]