

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)

Институт информационных технологий и управляющих систем
Кафедра программного обеспечения вычислительной техники и
автоматизированных систем
Направление подготовки 09.03.04 – Программная инженерия
Направленность (профиль) образовательной программы *Разработка*
программно-информационных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

**«Разработка front-end Web – приложения – учебной среды с
чатами и AI-анализом кода лабораторных работ»**

Студент: Бондаренко Сергей Владимирович
Зав. кафедрой: канд. техн. наук, доц. Поляков В.М.
Руководитель: Мельников А.Б.

К защите допустить:

Зав. кафедрой _____ **/Поляков В.М./**

«_____» _____ 2024 г.

Белгород 2025 г.

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)

Институт информационных технологий и управляющих систем
Кафедра программного обеспечения вычислительной техники и
автоматизированных систем
Направление подготовки *09.03.04 – Программная инженерия*
Направленность (профиль) образовательной программы *Разработка*
программно-информационных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

**«Разработка front-end Web – приложения – учебной среды с
чатами и AI-анализом кода лабораторных работ»**

Студент: Бондаренко Сергей Владимирович
Зав. кафедрой: канд. техн. наук, доц. Поляков В.М.
Руководитель: Мельников А.Б.

К защите допустить:

Зав. кафедрой _____ **/Поляков В.М./**

«___» _____ 2024 г.

Белгород 2025 г.

ОПРЕДЕЛЕНИЯ, СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ

ИИ - искусственный интеллект.

Изм.	Лист	№ докум.	Подп.	Дата				
Разраб.	Иванов А. Б.				Название документа Название дипломной работы Название дипломной работы	Лит.	Лист	Листов
Руковод.	Мельников А.Б.						1	17
Консул.								
Н. контр.	Н.Кнтр. И.О.							
Зав. Каф.	Поляков В.М.							

Содержание

Введение	3
ОБЗОР И ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	6
Введение в предметную область	6
Анализ существующих образовательных платформ	7
Проблемы существующих решений	8
Потребности образовательной среды	8
Технологии разработки клиентской части приложения	10
Требования к функциональности приложения	13
Выводы по результатам анализа предметной области	14
Список литературы	16
Приложения	17

Изм.	Лист	№ докум.	Подп.	Дата				
Разраб.		Иванов А. Б.			Название документа Название дипломной работы Название дипломной работы	Лит.	Лист	Листов
Руковод.		Мельников А.Б.					2	17
Консул.								
Н. контр.		Н.Кнтр. И.О.						
Зав. Каф.		Поляков В.М.						

Введение

Развитие цифровых технологий в сфере образования значительно меняет способы взаимодействия между преподавателями и студентами, предоставляя новые возможности для обучения и обмена информацией. В условиях дистанционного и смешанного обучения особенно важной становится необходимость создания платформ, которые бы объединяли образовательные инструменты в едином пространстве. Веб-приложения, которые решают задачи взаимодействия, позволяют сократить барьеры между преподавателями и студентами, улучшить коммуникацию и повысить качество образования. Цифровая среда должна обеспечивать не только размещение учебных материалов и заданий, но и средства для общения, автоматической оценки и анализа решений с использованием современных технологий, включая искусственный интеллект.

Актуальность темы заключается в потребности создания интегрированной образовательной платформы, которая объединяет функции чатов, проведения занятий и автоматического анализа решений, используя возможности ИИ. На данный момент отсутствует единая система, которая бы эффективно сочетала в себе эти ключевые аспекты: возможность общения через чаты, создание заданий и автоматизированную проверку решений с помощью ИИ. Современные платформы, как правило, фрагментированы — отдельные системы для чатов, другие для размещения заданий, третьи для автоматической проверки кода, что значительно усложняет организацию учебного процесса и снижает его эффективность. Разработка интегрированного решения, которое объединило бы эти элементы, позволяет улучшить качество образовательного процесса, повысив продуктивность студентов и преподавателей, а также упростив взаимодействие и автоматизировав многие рутинные задачи.

Целью данной работы является разработка клиентской части образовательной платформы, которая будет включать функции взаимодействия между преподавателями и студентами, автоматизированную проверку кода, а также возможности общения в рамках чатов. Особое внимание уделяется созданию такого интерфейса,

Изм.	Лист	№ докум.	Подп.	Дата				
Разраб.	Иванов А. Б.				Название документа Название дипломной работы Название дипломной работы	Лит.	Лист	Листов
Руковод.	Мельников А.Б.						3	17
Консул.								
Н. контр.	Н.Кнтр. И.О.							
Зав. Каф.	Поляков В.М.							

который позволит преподавателям и студентам взаимодействовать в едином пространстве, где будут доступны все образовательные инструменты и ресурсы.

Для достижения поставленной цели необходимо решить следующие задачи:

- а) Проанализировать предметную область и существующие системы, выявив их сильные и слабые стороны.
- б) Определить архитектурные и технологические решения, подходящие для реализации клиентской части платформы.
- в) Спроектировать пользовательский интерфейс, обеспечивающий интуитивное и удобное взаимодействие для преподавателей и студентов.
- г) Разработать компоненты для управления учебными структурами (институт, кафедра, группа), заданиями и чатами.
- д) Интегрировать средства для автоматизированной проверки решений студентов с применением ИИ.
- е) Реализовать тестирование бизнес-логики приложения для обеспечения её корректности и эффективности.

Структура пояснительной записки включает следующие разделы:

- В первом разделе рассматриваются особенности предметной области, проводится анализ существующих решений и обоснование выбора технологий и методов проектирования. Приводится обзор существующих образовательных платформ и их недостатков, а также объясняется необходимость разработки интегрированного решения.
- Во втором разделе описывается архитектура клиентской части приложения, структура пользовательского интерфейса, проектирование компонентов и их взаимодействие. Рассматриваются решения для реализации системы чатов, создания и проверки заданий, а также интеграции ИИ-анализа.
- В третьем разделе приводится описание реализации: структура кода, используемые технологии (Next.js, React, TypeScript, Redux, Auth.js), описание экрана и взаимодействий, примеры реализации различных компонентов системы.

						Лист
						4
Изм.	Лист	№ докум.	Подп.	Дата		

- В заключении приводятся выводы по выполненной работе, оценивается эффективность разработанного интерфейса и функционала, а также определяются направления для дальнейшего развития и улучшения системы. Указываются перспективы внедрения ИИ в образовательные платформы для улучшения процессов оценки и взаимодействия.

						Лист
						5
Изм.	Лист	№ докум.	Подп.	Дата		

ОБЗОР И ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Введение в предметную область

Современные образовательные процессы переживают значительные изменения под воздействием цифровых технологий. В условиях быстрого роста объемов информации и перехода на дистанционное и смешанное обучение возникает потребность в создании платформ, которые объединяют различные образовательные инструменты в единую систему. Проблемы, с которыми сталкиваются преподаватели и студенты, включают фрагментацию существующих решений: чаты для общения, отдельные системы для размещения и проверки заданий, а также инструменты для анализа решений студентов.

Существующие платформы не всегда обеспечивают интеграцию всех этих функций в одном приложении, что приводит к необходимости использования множества разных сервисов для выполнения учебных задач. В рамках образовательных процессов это усложняет взаимодействие между преподавателями и студентами, увеличивает время на организацию обучения и снижает его эффективность.

Одной из важнейших задач является создание платформы, которая объединяет все эти компоненты в одном месте, обеспечивая удобный интерфейс для студентов и преподавателей. Такая система должна включать:

- возможность создания и размещения учебных заданий;
- автоматическое тестирование решений студентов с использованием ИИ для проверки правильности кода;
- чат-функциональность для общения студентов с преподавателями и внутри групп;
- централизованный доступ к учебным материалам.

Интеграция всех этих функций в одну платформу позволит значительно упростить организацию учебного процесса, улучшить взаимодействие между пре-

Изм.	Лист	№ докум.	Подп.	Дата				
Разраб.	Иванов А. Б.				Название документа Название дипломной работы Название дипломной работы	Лит.	Лист	Листов
Руковод.	Мельников А.Б.						6	17
Консул.								
Н. контр.	Н.Кнтр. И.О.							
Зав. Каф.	Поляков В.М.							

подавателями и студентами, а также повысить качество обучения за счет автоматизации рутинных задач.

Анализ существующих образовательных платформ

Современные образовательные платформы, такие как Moodle, Google Classroom, Microsoft Teams для образования, а также специализированные решения, предназначенные для работы с программированием, предлагают различные функциональные возможности для взаимодействия преподавателей и студентов. Однако каждая из этих платформ имеет свои ограничения и не всегда покрывает все потребности в рамках единой системы.

Moodle

Moodle является одной из самых популярных образовательных платформ, используемых во многих учебных заведениях. Она предоставляет инструменты для размещения учебных материалов, организации тестов и заданий, а также ведения онлайн-курсов. Однако, несмотря на свои возможности, Moodle не предоставляет встроенных решений для автоматической проверки кода студентов, а также не включает в себя продвинутые механизмы общения в реальном времени, что делает её менее эффективной для динамичного взаимодействия в процессе обучения.

Google Classroom

Google Classroom предлагает простоту в использовании и позволяет интегрировать различные Google сервисы. Платформа позволяет преподавателям создавать задания, прикреплять материалы и отслеживать выполнение студентами. Однако Google Classroom не предоставляет функциональности для автоматического анализа решений, особенно в контексте программирования. Это требует интеграции с внешними инструментами, что усложняет использование системы в образовательных учреждениях.

Microsoft Teams for Education

Microsoft Teams, в отличие от Moodle и Google Classroom, активно используется для организации видеоконференций и групповых чатов. Он позволяет преподавателям и студентам взаимодействовать в реальном времени, а также интегрирует различные сервисы Microsoft 365. Однако, как и в случае с другими платформами,

Microsoft Teams не предоставляет функционала для интегрированного анализа кода студентов с использованием искусственного интеллекта, что ограничивает его возможности в обучении программированию.

Платформы для анализа кода

Существуют специализированные платформы, такие как CodeSignal, Codility, LeetCode, которые позволяют преподавателям и работодателям тестировать навыки программирования студентов. Эти системы используют алгоритмы для автоматической проверки решений, однако они ограничены в функционале взаимодействия с преподавателями и студентами, а также не обеспечивают централизованный доступ к учебным материалам и заданиям.

Проблемы существующих решений

Основной проблемой существующих образовательных платформ является фрагментация функционала. На данный момент нет единой платформы, которая бы эффективно объединяла создание и проверку заданий, общение преподавателей и студентов, а также использовала бы технологии ИИ для автоматизированного анализа решений студентов. Это затрудняет образовательный процесс и снижает его эффективность, особенно в условиях быстро меняющихся требований дистанционного обучения.

Таким образом, для улучшения образовательного процесса существует необходимость в разработке единой интегрированной платформы, которая бы сочетала в себе все эти компоненты и обеспечивала бы максимально удобное взаимодействие для всех участников учебного процесса.

Потребности образовательной среды

Современные образовательные процессы предъявляют высокие требования к функциональности учебных платформ. Для эффективного взаимодействия между преподавателями и студентами необходимо создавать приложения, которые обеспечивают организационную и техническую поддержку всех ключевых элементов образовательного процесса.

						Лист
						8
Изм.	Лист	№ докум.	Подп.	Дата		

Доступность материалов и заданий

Материалы и задания должны быть доступны студентам в любое время. Приложение должно обеспечивать размещение учебных ресурсов в различных форматах (текст, видео, презентации) и упрощать их поиск и использование. Это позволяет студентам готовиться к занятиям и выполнять задания без привязки ко времени, а преподавателям — быстро обновлять и дополнять учебные модули.

Автоматизация проверок и оценки

Автоматизированная проверка заданий существенно ускоряет процесс получения обратной связи. Использование искусственного интеллекта для анализа кода позволяет выявлять ошибки, давать подсказки и оценивать работы без участия преподавателя. Это освобождает ресурсы преподавателя для индивидуальной поддержки студентов и более сложной экспертной оценки.

Удобная система заданий и общения

Приложение должно включать удобную систему создания и отслеживания заданий. Важно, чтобы преподаватели могли формулировать задания, прикреплять к ним материалы и получать результаты выполнения. Неотъемлемой частью также является возможность общения между участниками процесса — как в групповых, так и личных чатах, для обмена мнениями и получения поддержки.

Интеграция всех процессов в одну систему

Отдельные решения для чатов, размещения заданий и анализа кода создают фрагментированную среду. Необходима единая платформа, объединяющая все эти компоненты. Это упрощает взаимодействие, повышает удобство и эффективность обучения, а также снижает затраты на сопровождение и обучение работе с системой.

Вывод

Таким образом, при проектировании образовательной платформы следует учитывать потребности в постоянном доступе к материалам, автоматической проверке решений, поддержке взаимодействия и целостности функционала в рамках одного интерфейса.

						Лист
						9
Изм.	Лист	№ докум.	Подп.	Дата		

Технологии разработки клиентской части приложения

Для реализации клиентской части платформы выбраны современные инструменты, обеспечивающие модульность, производительность, типизацию и масштабируемость интерфейса.

TypeScript

JavaScript является одним из самых популярных языков программирования для веб-разработки. Он широко используется для создания динамичных веб-страниц и приложений, поскольку позволяет работать с элементами DOM, асинхронно загружать данные и обеспечивать интерактивность пользовательских интерфейсов. Однако JavaScript имеет важный недостаток — отсутствие статической типизации. Это означает, что переменные и функции не привязываются к определённым типам данных, что может привести к ошибкам на этапе выполнения, которые трудно обнаружить в процессе разработки. Особенно это может быть проблемой в крупных приложениях, где сложно отслеживать все возможные типы данных и их изменения.

Для устранения этих проблем был разработан язык TypeScript, являющийся надмножеством JavaScript. TypeScript добавляет в JavaScript статическую типизацию, что позволяет разработчикам явно указывать типы данных для переменных и функций. Это значительно снижает вероятность ошибок и улучшает поддержку кода в будущем. Благодаря строгой типизации TypeScript помогает предотвращать баги, связанные с динамическими типами в JavaScript, и улучшает автозаполнение в редакторах кода. TypeScript распространяется как библиотека, которую можно интегрировать в проекты на JavaScript, обеспечивая совместимость с существующим кодом и позволяя постепенно внедрять типизацию без необходимости переписывать весь проект. Это особенно важно в крупных и масштабируемых приложениях, где несколько разработчиков работают с общими компонентами, и типизация помогает поддерживать консистентность кода на протяжении всего проекта.

React

React используется как библиотека для построения пользовательских интерфейсов. Она позволяет эффективно обновлять и рендерить компоненты при изменении данных, что делает интерфейс приложения быстрым и отзывчивым. React обеспечивает декларативный подход к построению UI, где разработчик описывает,

						Лист
						10
Изм.	Лист	№ докум.	Подп.	Дата		

как должен выглядеть интерфейс при разных состояниях приложения, а React самостоятельно управляет его обновлением. Это позволяет избежать многих ошибок, связанных с ручным управлением DOM, и значительно ускоряет процесс разработки.

Next.js

Next.js — это популярный фреймворк для React, который значительно расширяет его возможности, предоставляя разработчикам мощные инструменты для создания высокопроизводительных веб-приложений. Одной из ключевых особенностей Next.js является поддержка рендеринга на сервере (SSR, Server-Side Rendering) и статической генерации контента (SSG, Static Site Generation). Эти подходы позволяют улучшить производительность приложений, поскольку они обеспечивают быструю загрузку страниц, оптимизированную для поисковых систем и пользователей.

С серверным рендерингом Next.js позволяет генерировать HTML на сервере для каждой страницы перед её отправкой клиенту, что обеспечивает быстрое отображение контента. Это особенно полезно для SEO, поскольку поисковые системы могут индексировать контент сразу после его загрузки. Такой подход значительно улучшает видимость веб-приложений в поисковых системах и способствует их более высокому ранжированию.

Одним из ключевых преимуществ Next.js является автоматическая разбивка кода (code splitting). Это означает, что Next.js разделяет приложение на небольшие части, которые загружаются только по мере необходимости, что помогает сократить время загрузки страниц и улучшить пользовательский опыт. Таким образом, браузер загружает только тот код, который необходим для отображения текущей страницы, а не весь код приложения.

Кроме того, Next.js поддерживает гибкие методы рендеринга, что дает разработчикам возможность выбирать наиболее подходящий способ для каждой страницы. Статическая генерация (SSG) идеально подходит для страниц, которые не изменяются часто и могут быть сгенерированы заранее, например, блоговые записи или страницы с информацией о компании. В то время как для динамических страниц, которые требуют актуализации данных на сервере при каждом запросе, можно использовать серверный рендеринг.

Next.js также упрощает настройку маршрутизации и управление данными.

						Лист
						11
Изм.	Лист	№ докум.	Подп.	Дата		

Встроенная система маршрутизации автоматически генерирует страницы на основе файловой структуры, что делает создание новых страниц и маршрутов простым и интуитивно понятным. Кроме того, Next.js предоставляет инструменты для работы с API, что позволяет без труда интегрировать серверную логику в приложение.

Еще одной значимой особенностью является поддержка типизации с помощью TypeScript, что делает разработку в Next.js ещё более удобной и безопасной. Комбинация TypeScript и Next.js позволяет создавать стабильные и хорошо структурированные приложения, минимизируя количество ошибок на этапе разработки.

Важно отметить, что реализацию приложения можно было бы построить и на чистом React, однако в этом случае значительная часть функциональности, такой как маршрутизация, SSR, SSG и работа с API, потребовала бы ручной настройки и подключения дополнительных библиотек. Использование Next.js избавляет от необходимости собирать всё вручную и предоставляет готовую, хорошо спроектированную архитектуру. Таким образом, Next.js становится де-факто стандартом разработки современных React-приложений. Это не просто библиотека, а фреймворк — а значит, он предлагает определённую «протоптанную дорожку», следование которой позволяет создавать более надёжные и поддерживаемые решения.

Auth.js

Auth.js — это библиотека для реализации аутентификации и авторизации в веб-приложениях. Она является официальным решением, рекомендуемым и поддерживаемым фреймворком Next.js, что гарантирует хорошую интеграцию и поддержку всех необходимых функций. Библиотека позволяет легко подключать сторонние провайдеры аутентификации, такие как Google, Facebook и другие, а также реализовывать собственную аутентификацию с использованием базы данных. Auth.js обеспечивает надёжную защиту пользовательских данных, управление сессиями, работу с токенами и предоставляет удобные API для быстрой настройки. Это решение упрощает реализацию всех ключевых механизмов безопасности, освобождая разработчиков от необходимости погружаться в тонкости реализации.

Redux

Redux — это библиотека для управления состоянием в приложениях, основанных на React. Она используется для централизованного хранения состояния приложения, что облегчает обмен данными между компонентами и упрощает их вза-

						Лист
						12
Изм.	Лист	№ докум.	Подп.	Дата		

имодействие. Redux помогает избежать ”проблемы пропс-дерева” в больших приложениях, когда передача данных через множество вложенных компонентов становится сложной. Хотя Redux часто используется в более сложных приложениях, в данном проекте его роль заключается в том, чтобы сделать взаимодействие между компонентами более организованным и улучшить предсказуемость состояния приложения.

Требования к функциональности приложения

Разрабатываемое приложение представляет собой образовательную платформу, ориентированную на университетскую среду. Основная цель — предоставить единое пространство для организации учебного процесса, взаимодействия между преподавателями и студентами, а также управления учебными структурами.

Регистрация и структура университетов

Каждый университет имеет возможность зарегистрироваться на платформе и получить доступ к собственной административной панели. Через неё администраторы могут создавать внутреннюю структуру: институты, кафедры и учебные группы. Эти сущности используются как основа для управления доступом, назначения преподавателей и приглашения студентов.

Панель преподавателя

Преподаватели, закреплённые за кафедрами, получают доступ ко всем учебным группам, относящимся к соответствующей кафедре. Через панель преподавателя доступен следующий функционал:

- создание групповых чатов для любой группы своей кафедры;
- размещение учебных материалов — как в групповых чатах, так и в личных сообщениях;
- формирование и отправка заданий для студентов;
- просмотр и анализ результатов выполнения заданий;
- предоставление обратной связи студентам.

Панель студента

Студенты, присоединённые к определенным группам, имеют доступ к:

						Лист
						13
Изм.	Лист	№ докум.	Подп.	Дата		

- групповым чатам своей учебной группы;
- личной переписке с преподавателями;
- материалам, отправленным преподавателями;
- заданиям, опубликованным в рамках их группы;
- форме отправки решений и получению обратной связи.

Предлагаемое приложение должно закрыть этот разрыв, обеспечив единую среду, в которой объединены функции управления учебным процессом, общения, публикации и проверки заданий. Такой подход позволит повысить качество образовательного взаимодействия, сократить технические барьеры и обеспечить более

высокую степень вовлечённости пользователей.

Таким образом, на основании проведённого анализа подтверждается необходимость разработки новой системы, в которой ключевые элементы образовательной среды будут интегрированы в одно приложение, удовлетворяющее современным требованиям пользователей.

						Лист
						15
Изм.	Лист	№ докум.	Подп.	Дата		

Список литературы

Изм.	Лист	№ докум.	Подп.	Дата								
Разраб.		Иванов А. Б.			Название документа Название дипломной работы Название дипломной работы				Лит.	Лист	Листов	
Руковод.		Мельников А.Б.									16	17
Консул.												
Н. контр.		Н.Кнтр. И.О.										
Зав. Каф.		Поляков В.М.										

Приложения

Изм.	Лист	№ докум.	Подп.	Дата								
Разраб.		Иванов А. Б.			Название документа Название дипломной работы Название дипломной работы				Лит.	Лист	Листов	
Руковод.		Мельников А.Б.									17	17
Консул.												
Н. контр.		Н.Кнтр. И.О.										
Зав. Каф.		Поляков В.М.										