

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего  
образования «Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики  
Кафедра программных систем

**ОТЧЕТ ПО ПРАКТИКЕ**

Вид практики: учебная

Тип практики: Научно-исследовательская работа  
(получение первичных навыков научно-исследовательской работы)

Сроки прохождения практики: с 20.06.2025 г. по 03.07.2025 г.  
(в соответствии с календарным учебным графиком)

по направлению подготовки 02.03.02 «Фундаментальная информатика  
и информационные технологии»(уровень бакалавриата)  
направленность (профиль) «Информационные технологии»

Обучающийся группы № 6201-020302D \_\_\_\_\_ B.A. Бренева

Руководитель практики  
от университета, доцент кафедры  
программных систем, к.т.н., доцент \_\_\_\_\_ Д.А. Попова-Коварцева

Дата сдачи 03.07.2025 г.

Дата защиты 03.07.2025 г.

Оценка \_\_\_\_\_

Самара 2025

## СОДЕРЖАНИЕ

Задание (я) по практике для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью (сбор и анализ данных и материалов, проведение исследований)	3
Введение	6
1 Описание и анализ предметной области	7
1.1 Основные понятия и определения	7
1.2 Обзор существующих систем-аналогов	8
1.2.1 Система «Google Classroom»	8
1.2.2 Система «Stepik»	9
1.2.3 Система «iSpring Learn»	10
1.2.4 Сравнительный анализ систем-аналогов	10
1.2.5 Постановка задачи	11
2 Проектирование системы	14
2.1 Выбор и обоснование архитектуры системы	14
2.2 Выбор комплекса программных средств	17
Заключение	19
Список использованных источников	20
Приложение А Слайды презентационного материала	21

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики  
Кафедра программных систем

Задание (я) по практике для выполнения определенных видов работ,  
связанных с будущей профессиональной деятельностью  
(сбор и анализ данных и материалов, проведение исследований)

Обучающейся Бреневой Веронике Андреевне группы 6201-020302D

Направлен на практику приказом по университету от 11.06.2025 г. № 309-ПР в Самарский университет на кафедру программных систем.

Планируемые результаты освоения образовательной программы (компетенции)/индикаторы	Выполнение определенных видов работ, связанных с будущей профессиональной деятельностью (сбор и анализ данных и материалов, проведение исследований)	Результаты практики (планируемые результаты обучения при прохождении практики)
<p><b>ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности</b></p> <p><b>ОПК- 1.1</b> Использует основные положения и концепции в области математических и естественных наук, базовые теории коммуникации; знает основную терминологию</p> <p><b>ОПК-1.2</b> Осуществляет первичный сбор и анализ материала, интерпретирует различные математические объекты</p> <p><b>ОПК-1.3</b> Осуществляет первичный сбор и анализ материала, интерпретирует</p>	<p>1) Ознакомиться с концепциями и терминологией по теме научно-исследовательской работы.</p> <p>2) Осуществить поиск материала по теме научно-исследовательской работы в сети Интернет, электронных библиотечных системах и базах данных.</p> <p>3) Ознакомиться со стандартом оформления текстовых учебных документов Самарского университета.</p> <p>4) Закрепить знания материала по стандарту оформления текстовых документов с помощью прохождения теста в системе Moodle.</p> <p>5) Оформить отчет по результатам прохождения практики в строгом</p>	<p>1) Ознакомилась с основными концепциями и терминологией, касающихся разработки мобильного приложения.</p> <p>2) Осуществила поиск материала по теме разработки мобильных приложений с дополненной реальностью в сети Интернет, электронных библиотечных системах и базах данных.</p> <p>3) Ознакомилась со стандартом оформления текстовых учебных документов Самарского университета.</p> <p>4) Закрепила знания материала по стандарту оформления текстовых</p>

различные математические объекты	соответствии со стандартом оформления текстовых учебных документов.	документов с помощью прохождения теста в системе Moodle.
<p><b>ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности</b></p> <p><b>ОПК-2.1</b> Использует основные положения и концепции в области программирования, архитектуру языков программирования, теорию коммуникации, основную терминологию, знаком с содержанием единого реестра российских программ</p> <p><b>ОПК-2.2</b> Анализирует код на типовых языках программирования, может составлять программы</p> <p><b>ОПК-2.3</b> Применяет опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций</p>	<p>1) Ознакомиться с рекомендациями по подготовке презентационного материала.</p> <p>2) Подготовить презентацию по теме научно-исследовательской работы в соответствии с общими правилами оформления презентаций.</p> <p>3) Закрепить знания материала по подготовке презентационного материала с помощью прохождения теста в системе Moodle.</p>	<p>1) Ознакомилась с рекомендациями по подготовке презентационного материала.</p> <p>2) Подготовила презентацию по теме научно-исследовательской работы в соответствии с общими правилами оформления презентаций.</p> <p>3) Закрепила знания материала по подготовке презентационного материала с помощью прохождения теста в системе Moodle.</p>
<p><b>ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз</b></p>	<p>1) Ознакомиться с материалом по построению схем алгоритмов.</p> <p>2) Закрепить знания материала по построению схем алгоритмов с помощью прохождения теста в системе Moodle.</p>	<p>1) Ознакомилась с материалом по построению схем алгоритмов.</p> <p>2) Закрепила знания материала по построению схем алгоритмов с помощью прохождения теста в системе Moodle.</p>

<p><b>данных, тестов и средств тестиирования систем и средств на соответствие стандартам и исходным требованиям</b></p> <p><b>ОПК-3.1</b> Понимает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей</p> <p><b>ОПК-3.2</b> Соотносит знания в области программирования, интерпретирует прочитанное, определяет и создает информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем</p> <p><b>ОПК-3.3</b> Имеет практический опыт применения разработки программного обеспечения</p>		
--	--	--

Срок представления на кафедру отчета о практике 03.07.2025 г.

Руководитель практики от университета,  
доцент кафедры программных  
систем, к.т.н., доцент

\_\_\_\_\_ Д.А. Попова-Коварцева  
(подпись)

Задание принял к исполнению  
обучающийся группы № 6201-020302D

\_\_\_\_\_ В.А. Бренева  
(подпись)

## ВВЕДЕНИЕ

Современные цифровые технологии кардинально преобразуют сферу образования, создавая новые возможности для оптимизации учебного процесса. Особенно актуальна эта тенденция для учреждений дополнительного образования - школ программирования, языковых центров и профессиональных курсов, где требования к гибкости и персонализации учебных программ особенно высоки. В настоящее время такие организации сталкиваются с острой необходимостью автоматизации ключевых процессов: управления курсами, контроля успеваемости, организации занятий и проверки заданий [1].

Актуальность разработки специализированного веб-приложения подтверждается стремительным ростом рынка дополнительного образования (8-10% ежегодно по данным Research and Markets) и явным дефицитом комплексных решений, адаптированных под специфику малых и средних образовательных центров. При этом преподаватели тратят до 40% рабочего времени на рутинные административные задачи, что негативно сказывается на качестве образовательного процесса.

Разрабатываемая система предназначена для широкого круга учреждений дополнительного образования - от частных школ программирования до корпоративных учебных центров. Ее пользователями станут администраторы, преподаватели, учащиеся и родители, для которых платформа предоставит единое пространство организации учебного процесса. Система будет поддерживать все форматы обучения - офлайн, онлайн и гибридные, что особенно важно в современных условиях.

Целью настоящей работы является создание веб-приложения, способного комплексно решить проблемы автоматизации учебного процесса в учреждениях дополнительного образования. Научная новизна проекта заключается в разработке адаптивных алгоритмов проверки заданий, создании гибкой системы шаблонов курсов и внедрении инструментов прогнозной аналитики. Практическая значимость работы определяется потенциальной возможностью сократить временные затраты преподавателей на 30-40%, повысить эффективность контроля успеваемости и оптимизировать нагрузку педагогического состава.

Реализация проекта включает несколько этапов: анализ существующих решений, проектирование архитектуры системы, разработку ключевых модулей (управление курсами, видеоконференции, проверка заданий), создание MVP и его тестирование. В результате будет предложено технологическое решение, способное существенно повысить эффективность организации учебного процесса в учреждениях дополнительного образования.

## 1        Описание и анализ предметной области

Предметная область – это часть реального мира, которая подлежит изучению с целью автоматизации организации управления [2].

### 1.1     Основные понятия и определения

Дополнительное образование – форма обучения, направленная на расширение знаний, умений и навыков за рамками основных образовательных программ. Включает курсы, школы и программы по различным направлениям (программирование, языки, дизайн и др.), ориентированные на практическое применение знаний.

Автоматизация учебного процесса – применение программных решений для оптимизации административных и педагогических задач: планирования занятий, проверки работ, учета успеваемости и организации коммуникации между участниками образовательного процесса.

Веб-приложение для образования – программное обеспечение, доступное через браузер, которое предоставляет инструменты для управления обучением, проведения занятий и контроля результатов. В отличие от мобильных приложений, не требует установки и поддерживает кроссплатформенность.

Онлайн-обучение – формат образовательного процесса, при котором взаимодействие между преподавателем и учащимися происходит дистанционно с использованием цифровых технологий (видеоконференции, чаты, интерактивные задания).

Гибридное обучение – комбинация оффлайн и онлайн-форматов, позволяющая сочетать преимущества живого общения и гибкости дистанционных технологий, это помогает лучше усваивать информацию.

Система управления обучением (LMS) – платформа для создания, проведения и администрирования учебных курсов. Включает функционал для размещения материалов, назначения заданий и отслеживания прогресса [3].

Автоматическая проверка заданий – технология оценки работ учащихся без непосредственного участия преподавателя. Для программирования включает запуск кода в изолированной среде (песочнице) и сравнение результатов с эталонными.

Геймификация в образовании – внедрение игровых элементов (баллы, уровни, рейтинги) для повышения мотивации учащихся. Примеры: бейджи за выполнение заданий, «прокачка» навыков в виде дерева талантов.

Персонализированное обучение – подход, при котором учебные материалы и темп их освоения адаптируются под индивидуальные потребности каждого учащегося на основе анализа его прогресса. Современные образовательные платформы реализуют этот подход через алгоритмы рекомендательных систем и адаптивное тестирование [1]. Как показывают

исследования, персонализация позволяет повысить эффективность усвоения материала на 25-40% по сравнению с традиционными методами [4].

API интеграции – технология подключения сторонних сервисов (например, видеоконференций или проверки кода) к образовательной платформе без разработки собственных аналогов.

Минимально жизнеспособный продукт (MVP) – базовая версия системы с ключевыми функциями, достаточными для тестирования гипотез и сбора обратной связи.

## 1.2 Обзор существующих систем-аналогов

### 1.2.1 Система «Google Classroom»

Google Classroom – облачная платформа для организации учебного процесса, разработанная компанией Google. Сервис предоставляет инструменты для создания курсов, распределения заданий, проверки работ и коммуникации между преподавателями и учащимися.

На рисунке 1 показан интерфейс системы «Google Classroom».

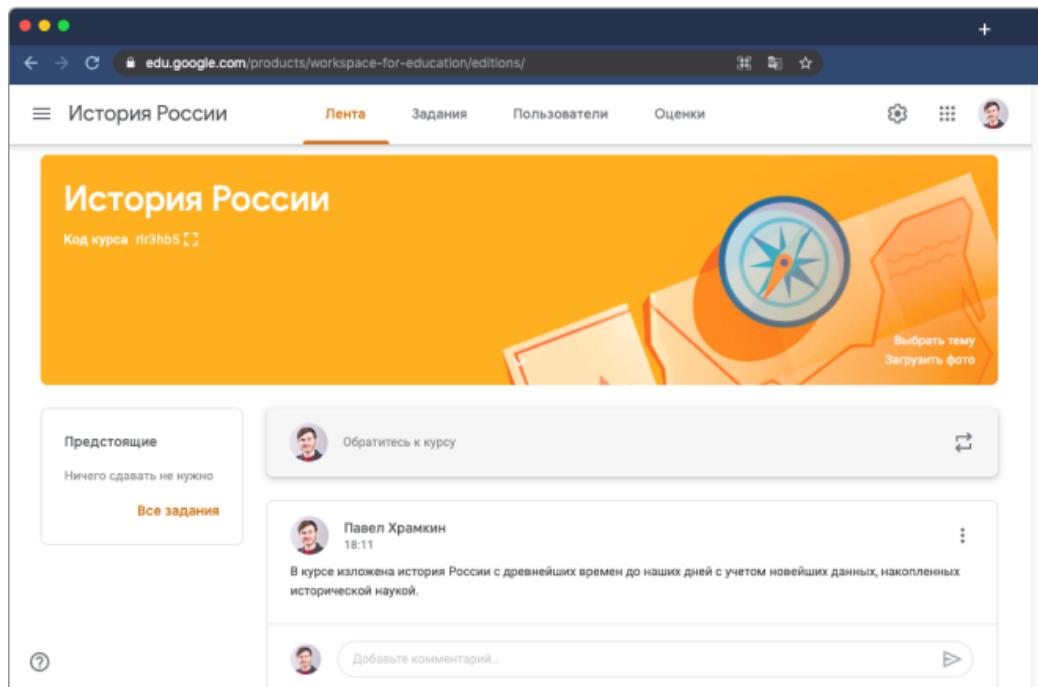


Рисунок 1 – Интерфейс работы программы Google Classroom

Ключевые особенности:

- интеграция с сервисами Google (Docs, Drive, Meet);
- простое создание и проверка заданий;
- система уведомлений о сроках сдачи работ.

Достоинства системы:

- бесплатный доступ к базовому функционалу;
- удобный интерфейс для быстрого создания курсов;

- возможность совместной работы над документами в реальном времени;
- мобильное приложение для iOS и Android.

Недостатки системы:

- отсутствие встроенной системы автоматической проверки заданий;
- ограниченные возможности аналитики успеваемости;
- требуется VPN для стабильной работы в России.

### 1.2.2 Система «Stepik»

Stepik – российская образовательная платформа с акцентом на ИТ-дисциплины и точные науки, поддерживающая автоматическую проверку заданий.

На рисунке 2 показан интерфейс системы «Stepik».

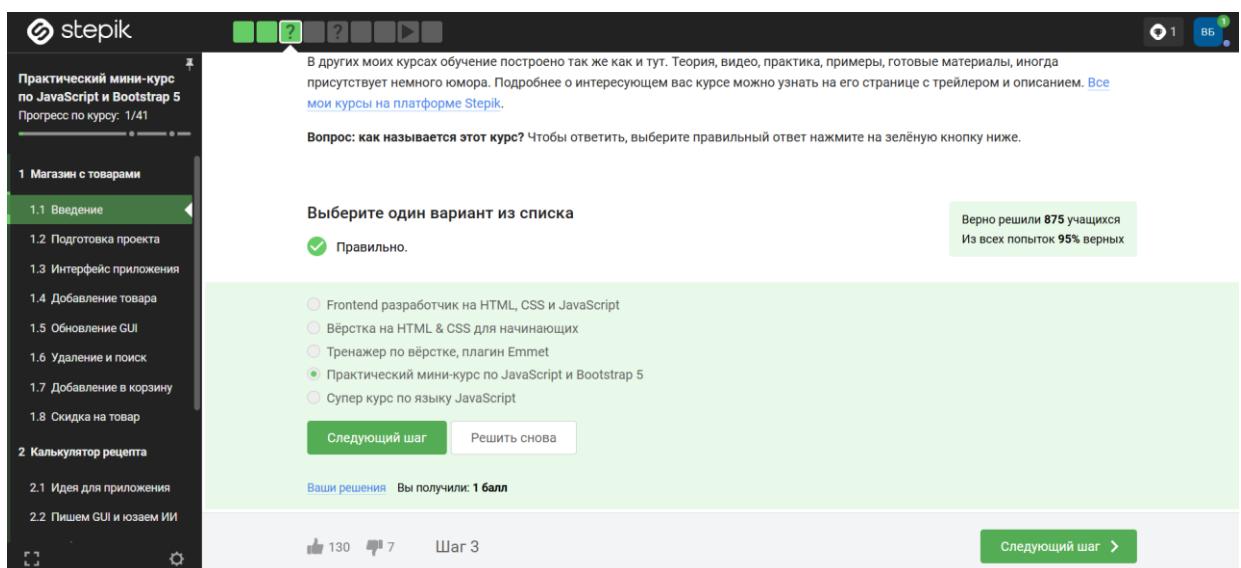


Рисунок 2 – Интерфейс работы программы «Stepik»

Ключевые особенности:

- встроенный инструмент для проверки программного кода;
- возможность создания интерактивных курсов;
- система достижений и сертификатов.

Достоинства системы:

- автоматическая проверка заданий по программированию;
- открытый доступ к большинству курсов;
- интеграция с GitHub;
- подробная статистика по прогрессу учащихся.

Недостатки системы:

- ограниченные возможности для гуманитарных дисциплин;
- требуется время на освоение интерфейса преподавателя;
- отсутствие встроенных инструментов для видеосвязи.

### 1.2.3 Система «iSpring Learn»

iSpring Learn – российская LMS-платформа для корпоративного и дополнительного образования с мощным конструктором курсов.

На рисунке 3 показан интерфейс программы iSpring Learn.

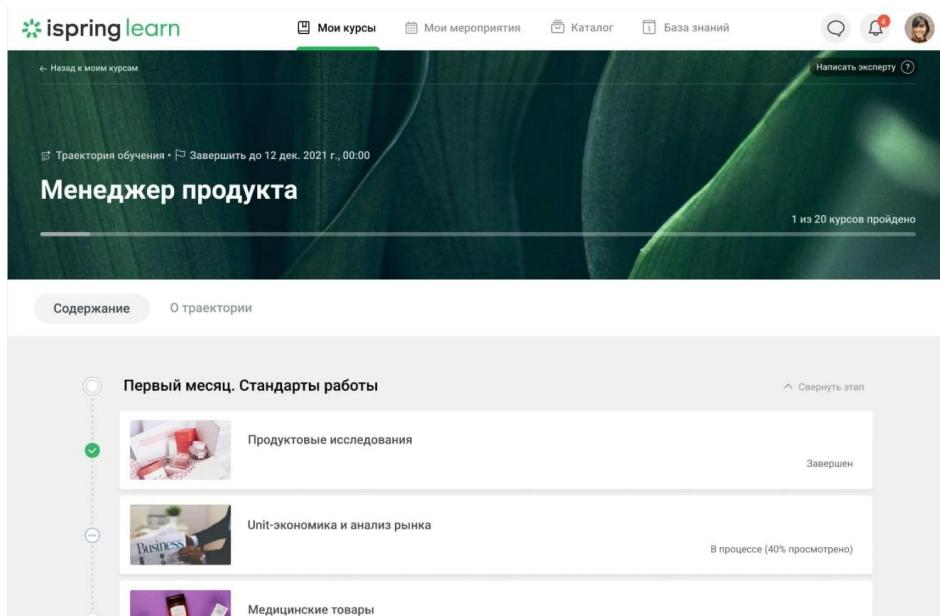


Рисунок 3 – Интерфейс работы программы «iSpring Learn»

Ключевые особенности:

- простой конструктор курсов без навыков программирования;
- система вебинаров и тестирования;
- геймификация обучения.

Достоинства системы:

- интуитивно понятный интерфейс;
- встроенные шаблоны для быстрого создания курсов;
- подробная аналитика успеваемости;
- мобильное приложение для учащихся.

Недостатки системы:

- ограниченный бесплатный период (14 дней);
- отсутствие автоматической проверки кода;
- высокая стоимость коммерческой версии.

### 1.2.4 Сравнительный анализ систем-аналогов

На основании проведенного обзора ключевых платформ был выполнен сравнительный анализ, который позволяет выявить сильные и слабые стороны каждой системы применительно к требованиям автоматизации учебного процесса в учреждениях дополнительного образования. Основные критерии сравнения включали поддержку

автоматической проверки заданий, условия доступа, возможности интеграции и удобство интерфейса.

В таблице 1 приведены характеристики систем-аналогов.

Таблица 1 – Сравнительные характеристики систем-аналогов

Показатели	Системы для сравнения		
	Google Classroom	Stepik	iSpring Learn
Автопроверка заданий	–	+	–
Бесплатный доступ	+	+	–
Интеграция с сервисами	Google	GitHub	Zoom
Геймификация	–	+	+
Понятность интерфейса (1-5)	5	2	3

#### 1.2.5 Постановка задачи

Во время выполнения выпускной квалификационной работы необходимо разработать веб-приложение для автоматизации учебного процесса в школах дополнительного образования. С его помощью администраторы смогут эффективно организовывать образовательный процесс, преподаватели — создавать интерактивные курсы и проверять задания, а учащиеся — получать доступ к учебным материалам, получать знания в доступном формате, изучать то, что им интересно и действительно важно и отслеживать свой прогресс в обучении.

Приложение должно предоставлять пользователям следующие возможности:

- создание и управление учебными курсами;
- формирование расписания занятий;
- публикация учебных материалов различных форматов;
- выполнение и проверка заданий;
- контроль успеваемости учащихся.

Для начала работы с системой пользователю необходимо пройти регистрацию, указав:

- имя (от 2 до 30 символов);
- фамилию (от 4 до 40 символов);
- электронную почту (должна соответствовать формату email);
- пароль (от 8 символов, включая цифры и буквы).

Система должна проверять корректность введенных данных и выводить соответствующие сообщения об ошибках при их наличии.

При работе с веб-приложением пользователи смогут:

- проходить регистрацию и авторизацию в системе;

- создавать и редактировать учебные курсы;
- добавлять учебные материалы и задания;
- выполнять задания и получать результаты проверки;
- отслеживать свой прогресс по курсам;
- просматривать статистику и аналитику;
- редактировать данные профиля.

Вся информация, необходимая для работы системы, будет храниться в базе данных, структура которой включает следующие основные сущности:

- пользователи (id, имя, фамилия, email, пароль, роль);
- курсы (id, название, описание, дата создания, автор);
- учебные материалы (id, название, тип, содержание, курс);
- задания (id, название, описание, срок сдачи, курс);
- результаты выполнения (id, задание, пользователь, статус, оценка).

Необходимо обеспечить целостность и безопасность хранимых данных.

Таким образом, система должна решать следующие основные задачи:

- 1) функции системы:
  - аутентификация и авторизация пользователей;
  - управление учебными курсами и материалами;
  - организация выполнения и проверки заданий;
  - сбор и анализ данных об успеваемости;
  - формирование отчетов и статистики;
- 2) функции пользователей:
  - для администраторов:
    - a) управление пользователями системы;
    - b) создание и настройка курсов;
    - c) назначение преподавателей;
    - d) просмотр аналитических отчетов;
  - для преподавателей:
    - a) создание и редактирование учебных материалов;
    - b) формирование заданий и тестов;
    - c) проверка выполненных работ;
    - d) контроль успеваемости учащихся;
  - для учащихся:
    - a) просмотр учебных материалов;

- b) выполнение заданий и тестов;
- c) отслеживание своего прогресса;
- d) просмотр результатов и рекомендаций.

Система должна быть реализована как веб-приложение с адаптивным интерфейсом, доступным с различных устройств, и обеспечивать стабильную работу при одновременном использовании множеством пользователей.

## 2       Проектирование системы

### 2.1     Выбор и обоснование архитектуры системы

Архитектура системы представляет собой фундаментальную организацию программного обеспечения, определяющую взаимосвязи между его компонентами и принципы их взаимодействия. В современных веб-приложениях преимущественно используются три типа архитектур: локальная, файл-серверная и клиент-серверная.

Локальная архитектура предполагает выполнение всех вычислений и хранение данных непосредственно на устройстве пользователя. Такой подход не требует подключения к сети и внешним сервисам. Он применяется преимущественно для автономных приложений, не нуждающихся в обмене данными между пользователями.

Файл-серверная архитектура организует работу через централизованное хранилище данных. В этом случае пользовательские устройства подключаются к общему файловому серверу, который обеспечивает доступ к совместно используемым ресурсам. Однако основная обработка информации по-прежнему выполняется на клиентских устройствах.

Клиент-серверная архитектура представляет собой более совершенный подход, где серверная часть не только хранит данные, но и выполняет значительную часть обработки информации. Клиентские приложения отправляют запросы на сервер, который осуществляет необходимые вычисления и возвращает результаты. Именно такая архитектура лежит в основе большинства современных веб-приложений, включая образовательные платформы, системы электронной коммерции и корпоративные решения.

Для разрабатываемого веб-приложения автоматизации учебного процесса была выбрана клиент-серверная архитектура, так как она наилучшим образом соответствует требованиям проекта. Данный выбор обусловлен необходимостью централизованного хранения учебных материалов и результатов обучения, обеспечения многопользовательского доступа, а также потребностью в надежной системе аутентификации и авторизации пользователей.

#### 2.1.1   Выбор систем для разработки клиентской части

Разработка клиентской части веб-приложения для автоматизации учебного процесса требует тщательного подхода к выбору технологий. В современной веб-разработке существует несколько популярных фреймворков, каждый из которых предлагает уникальные возможности для создания интерактивных пользовательских интерфейсов. Основное внимание было уделено трем наиболее востребованным решениям:

- React.js — это JavaScript-библиотека от Facebook для создания динамических интерфейсов, использующая компонентный подход и Virtual DOM для эффективного рендеринга;

- Vue.js — прогрессивный фреймворк, отличающийся простотой освоения и гибкостью;
- Angular — полнофункциональный фреймворк от Google для разработки сложных корпоративных приложений.

После детального анализа требований проекта и сравнения характеристик различных технологий, выбор пал на React.js. Это решение обусловлено рядом ключевых факторов. Во-первых, компонентная архитектура React.js идеально подходит для образовательной платформы, где многие элементы интерфейса (карточки курсов, блоки заданий, элементы прогресса) могут быть выделены в переиспользуемые компоненты [4]. Во-вторых, виртуальный dom обеспечивает высокую производительность при работе с большими объемами данных, что критически важно для системы, которая будет обрабатывать множество учебных материалов и результатов учащихся.

Для расширения возможностей React.js и реализации всех требований к интерфейсу образовательной платформы был сформирован следующий технологический стек:

- Redux — для централизованного управления состоянием приложения;
- Material-ui — библиотека компонентов для создания единого визуального стиля;
- Axios — для организации взаимодействия с серверным API;
- Jest — для комплексного тестирования компонентов интерфейса.

Преимущества выбранного технологического стека особенно ярко проявляются в контексте образовательной платформы. Во-первых, React.js в сочетании с Material-UI позволяет создавать интуитивно понятные интерфейсы с плавными переходами и анимациями, что улучшает пользовательский опыт. Во-вторых, использование redux обеспечивает предсказуемость состояния приложения, что упрощает отладку и дальнейшее развитие системы. В-третьих, богатая экосистема react предоставляет множество готовых решений для типовых задач, таких как загрузка файлов, отображение прогресса или организация сложных форм.

Важным аспектом выбора стало сообщество разработчиков вокруг React.js. Большое количество обучающих материалов, активное развитие инструментов и регулярные обновления гарантируют, что платформа будет оставаться современной и поддерживаемой в течение длительного времени. Кроме того, популярность React.js среди российских разработчиков упрощает процесс формирования команды для дальнейшего развития проекта.

### 2.1.2 Выбор систем для разработки серверной части

Серверная часть образовательной платформы требует тщательного подхода к выбору технологий, так как от этого зависит надежность, производительность и масштабируемость всего решения. Рассмотрим основные варианты, доступные для реализации нашего проекта.

При анализе backend-фреймворков мы рассматривали четыре основных варианта, каждый из которых имеет свои особенности. Node.js с фреймворком Express показал наилучший баланс характеристик для нашего случая. Его event-loop архитектура особенно хорошо подходит для образовательной платформы, где важна обработка множества одновременных подключений студентов и преподавателей. В отличие от PHP, который демонстрирует более низкую производительность, или Java, требующей значительных ресурсов, Node.js обеспечивает оптимальное соотношение скорости работы и потребления памяти.

Особенно стоит отметить поддержку TypeScript в Node.js-экосистеме. Это важное преимущество, позволяющее разрабатывать более надежный и поддерживаемый код по сравнению с Python или PHP-решениями. Для образовательного проекта, где важна стабильность работы и возможность долгосрочной поддержки, это критически важно.

Таблица 3 – Сравнительный анализ серверных технологий

Особенности	Node.js + Express	Python + Django	Java + Spring
Производительность	Высокая (event-loop)	Средняя	Высокая
Экосистема	Богатая (npm)	Хорошая	Очень богатая
Кривая обучения	Низкая	Средняя	Высокая
Поддержка TypeScript	Полная	Частичная	Через дополнительные инструменты
Сообщество	Очень активное	Активное	Крупное
Использование памяти	Низкое	Среднее	Высокое

Для хранения данных мы выбрали гибридный подход, сочетающий PostgreSQL и MongoDB. PostgreSQL идеально подходит для структурированных данных: информации о пользователях, курсах, заданиях и успеваемости. Его надежная система транзакций гарантирует целостность данных при одновременном доступе множества пользователей.

MongoDB, в свою очередь, будет использоваться для хранения менее структурированных данных: чатов, комментариев, медиафайлов. Его гибкая схема позволяет легко адаптировать хранилище под меняющиеся требования образовательного процесса. Такой комбинированный подход дает нам преимущества обоих миров: надежность реляционной базы и гибкость NoSQL-решения.

Выбранный технологический стек – Node.js + Express на Google Cloud Platform с гибридной базой данных - представляет собой оптимальное решение для образовательной платформы. Он сочетает в себе:

- высокую производительность и масштабируемость;
- гибкость для реализации сложной образовательной логики;

- надежное хранение и защиту данных;
- экономическую эффективность;
- простоту интеграции с другими образовательными сервисами.

Этот выбор позволяет нам создать платформу, которая будет стабильно работать при любых нагрузках, обеспечивая бесперебойный образовательный процесс для всех участников. При этом архитектура системы предусматривает возможность дальнейшего масштабирования и добавления новых функций по мере развития проекта.

## 2.2 Выбор комплекса программных средств

Для разработки веб-приложения автоматизации учебного процесса был сформирован оптимальный набор технологических решений, учитывающий специфику образовательной платформы и современные тенденции веб-разработки.

В клиентской части основой выступила связка React.js и TypeScript. React.js был выбран благодаря своему компонентному подходу, который идеально соответствует модульной структуре образовательной платформы. Каждый элемент интерфейса - карточка курса, блок задания, прогресс-бар - может быть реализован как независимый переиспользуемый компонент. TypeScript добавляет статическую типизацию, что значительно повышает надежность кода и упрощает его поддержку, особенно важную для долгосрочных образовательных проектов. Виртуальный DOM в React обеспечивает высокую производительность при работе с большими объемами учебных материалов.

Серверная часть реализована на Node.js с использованием фреймворка Express.js. Этот выбор обусловлен несколькими ключевыми факторами. Во-первых, единая JavaScript-среда на клиенте и сервере ускоряет разработку и упрощает взаимодействие между командами. Во-вторых, event-loop архитектура Node.js оптимально подходит для обработки множества одновременных подключений студентов и преподавателей. Для сложных бизнес-процессов, таких как проверка заданий или формирование отчетов, используется NestJS - мощный фреймворк, предоставляющий готовые решения для масштабируемых приложений.

Система хранения данных построена на гибридной архитектуре. PostgreSQL отвечает за структурированные данные: информацию о пользователях, курсах, заданиях и успеваемости. Его надежная транзакционная модель гарантирует целостность данных при интенсивной учебной нагрузке. Для менее структурированного контента – чатов, комментариев, медиафайлов – используется MongoDB, чья гибкая схема позволяет адаптировать хранилище под динамически меняющиеся требования образовательного процесса.

Облачная инфраструктура развернута на Google Cloud Platform, которая предоставляет масштабируемые вычислительные ресурсы и удобные инструменты управления. Интеграция

с Firebase дает готовые решения для аутентификации пользователей, хранения файлов и работы в реальном времени. Это сочетание обеспечивает надежную основу для образовательной платформы, позволяя сосредоточиться на разработке функциональности, а не на инфраструктуре.

Дополнительные инструменты включают Docker для контейнеризации приложения, что упрощает развертывание и масштабирование, Kubernetes для оркестрации контейнеров при росте нагрузки, Git для контроля версий и Jest для комплексного тестирования. Такой набор технологий обеспечивает полный цикл разработки - от создания прототипа до промышленной эксплуатации.

Выбранный технологический стек сочетает в себе несколько ключевых преимуществ. Для разработчиков это единая среда на TypeScript, богатая экосистема и активное сообщество. Для образовательного процесса - высокая производительность под нагрузкой, надежное хранение данных и гибкость. Для администрации - простота развертывания, понятный мониторинг и экономическая эффективность. В совокупности это создает прочную основу для создания современной, масштабируемой и удобной образовательной платформы.

## ЗАКЛЮЧЕНИЕ

В ходе проведенного исследования была разработана концепция веб-приложения для автоматизации учебного процесса в учреждениях дополнительного образования. Проанализированы современные подходы к созданию подобных систем, что позволило сформировать обоснованный технологический выбор для будущей реализации проекта.

Были рассмотрены различные варианты технологических решений для клиентской части приложения. На основании сравнительного анализа выбран стек на основе React.js и TypeScript, который, согласно исследованиям и опыту аналогичных проектов, должен обеспечить создание удобного и производительного пользовательского интерфейса для всех категорий пользователей платформы.

Для серверной части проекта предложена архитектура на базе Node.js с использованием Express.js и NestJS. Такой выбор обусловлен требованиями к производительности при работе с образовательным контентом и необходимостью обеспечения масштабируемости системы. Анализ возможных подходов к хранению данных показал перспективность гибридного решения с использованием PostgreSQL и MongoDB.

Исследование облачных платформ продемонстрировало потенциальные преимущества Google Cloud Platform в сочетании с Firebase для образовательных проектов. Предложенная клиент-серверная архитектура, по данным аналогичных реализаций, должна обеспечить необходимую надежность и безопасность работы с учебными данными.

Проведенный анализ подтвердил теоретическую обоснованность выбранных решений для создания системы автоматизации учебного процесса. Дальнейшая работа будет направлена на практическую реализацию предложенной концепции с учетом выявленных в ходе исследования требований и ограничений.

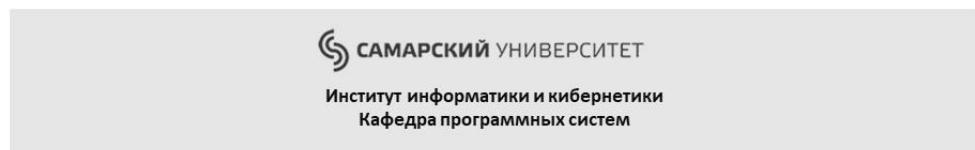
В приложении А приведены презентационные материалы для наглядного ознакомления с изученной темой.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Хуторской, А.В. Дидактика электронного обучения. / А.В. Хуторской – Москва: Издательство «Национальное образование», 2023. – 288 с. (дата обращения: 24.06.2025)
- 2 Архитектура информационных систем [Электронный ресурс]. URL: [https://kpfu.ru/staff\\_files/F\\_1284155032/Arkhitektura\\_informacionnykh\\_sistem.pdf](https://kpfu.ru/staff_files/F_1284155032/Arkhitektura_informacionnykh_sistem.pdf) (дата обращения: 25.06.2025)
- 3 Соловов, А.В. Проектирование и разработка систем управления обучением (LMS). / А.В. Соловов – Санкт-Петербург: Издательство «БХВ-Петербург», 2022. – 416 с. (дата обращения: 25.06.2025)
- 4 Документация по React.js [Электронный ресурс] / URL: <https://react.dev/> (дата обращения: 25.06.2025)

# ПРИЛОЖЕНИЕ А

## Слайды презентационного материала



### НАЗВАНИЕ ПРЕЗЕНТАЦИИ: РАЗРАБОТКА ВЕБ–ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ УЧЕБНЫХ ОНЛАЙН И ОФЛАЙН ПРОЦЕССОВ В УЧРЕЖДЕНИЯХ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ

Научно–исследовательская работа бакалавра (получение первичных навыков научно–исследовательской работы)

Выполнил: студент 6201–020302Д  
Бренева В.А.  
Научный руководитель: доцент кафедры  
программных систем, к.т.н., доцент  
Д. А. Попова–Коварцева

Самара 2025

Рисунок А.1 – Титульный слайд

### ПОСТАНОВКА ЗАДАЧИ

Цель работы:

Разработка веб–приложения для автоматизации учебных онлайн и офлайн процессов в учреждениях дополнительного образования

Задачи:

- изучить предметную область;
- изучить методы, технологии, алгоритмы и методики, значимые для рассматриваемой предметной области;
- провести анализ актуальности выбранной задачи;
- рассмотреть системы—аналоги;
- составить список функций разрабатываемого приложения;
- произвести выбор архитектуры разрабатываемой системы и средств реализации.

Разработка веб–приложения для автоматизации учебных онлайн и офлайн процессов в учреждениях дополнительного образования

2/12

Рисунок А.2 – Постановка задачи

## ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

**Автоматизация учебного процесса** – это комплекс программных решений, направленных на оптимизацию организации, проведения и контроля образовательной деятельности. Включает управление расписанием, материалами, заданиями и аналитикой успеваемости.

**Веб–приложение для образования** – это облачная платформа, обеспечивающая:

- централизованное хранение учебных материалов;
- интерактивные инструменты для преподавателей и учащихся;
- интеграцию с внешними образовательными сервисами;
- мульти платформенный доступ (ПК, планшеты, смартфоны).



Разработка веб–приложения для автоматизации учебных онлайн и оффлайн процессов в учреждениях дополнительного образования

3/12

Рисунок А.3 – Описание предметной области

## АНАЛИЗ СИСТЕМ—АНАЛОГОВ

### 1. «Google Classroom»

#### Функционал:

- организация курсов и учебных материалов;
- распределение заданий и сбор работ;
- интеграция с сервисами Google (Документы, Диск, Календарь);
- система оценивания и обратной связи.

A screenshot of the Google Classroom web interface. The main header says 'История России'. Below it, there's a large yellow banner with a compass icon and the text 'Выберите тему будущего дела'. On the left, there's a sidebar with 'Представления' (Presentations) and 'Все задания' (All assignments). In the center, there's a list of assignments and users. One user, 'Павел Кравчин', is shown with the note 'В курсе изложены истории России с древнейших времен до наших дней с учетом новых данных, полученных исторической науки.' At the bottom, there's a button 'Добавьте комментарий' (Add comment).

Разработка веб–приложения для автоматизации учебных онлайн и оффлайн процессов в учреждениях дополнительного образования

4/12

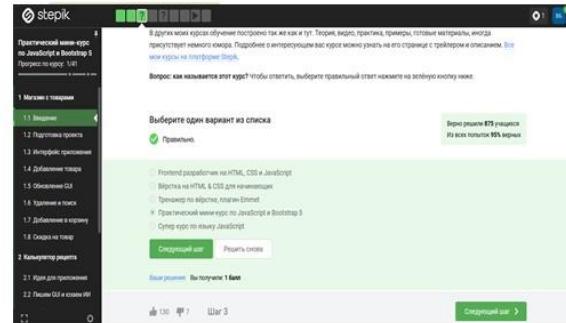
Рисунок А.4 – Анализ приложения «Google Classroom»

## АНАЛИЗ СИСТЕМ—АНАЛОГОВ

### 2. «Stepik»

#### Функционал:

- поддержка различных форматов контента (видео, тесты, практические задания);
- система достижений и сертификатов;
- API для интеграции с внешними сервисами.



Разработка веб—приложения для автоматизации учебных онлайн и оффлайн процессов в учреждениях дополнительного образования

5/12

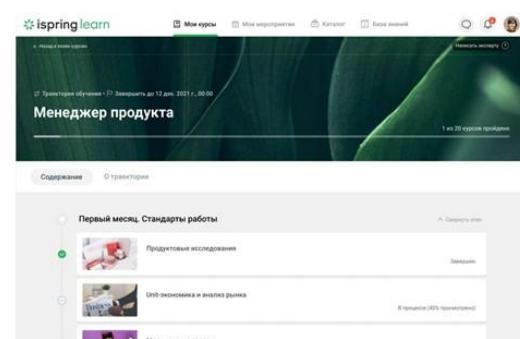
Рисунок А.5 – Анализ приложения «Stepik»

## АНАЛИЗ СИСТЕМ—АНАЛОГОВ

### 3. «iSpring Learn»

#### Функционал:

- платформа для организации дистанционного обучения;
- инструменты для быстрого создания интерактивных курсов;
- поддержка различных форматов учебного контента (видео, презентации, тесты);
- интеграция с видеоконференциями.



Разработка веб—приложения для автоматизации учебных онлайн и оффлайн процессов в учреждениях дополнительного образования

6/12

Рисунок А.6 – Анализ приложения «iSpring Learn»

## АНАЛИЗ СИСТЕМ—АНАЛОГОВ

Показатели	Веб-приложения для сравнения		
	Google Classroom	Stepik	iSpring Learn
Автопроверка знаний	-	+	-
Бесплатный доступ	+	+	-
Интеграция с сервисами	Google	Github	Zoom
Геймификация	-	+	+
Понятность интерфейса (1–5)	5	2	3

Разработка веб—приложения для автоматизации учебных онлайн и оффлайн процессов в учреждениях дополнительного образования

7/12

Рисунок А.7 – Сравнительная характеристика систем аналогов

## ПЕРЕЧЕНЬ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

Функции системы:

- система аутентификации и регистрации пользователей;
- каталог учебных курсов с фильтрацией и поиском;
- личные кабинеты для разных типов пользователей (администратор/преподаватель/учащийся);
- инструменты создания и управления учебными материалами;
- система назначения и проверки заданий;
- аналитика успеваемости и генерация отчетов;
- интеграция с сервисами видеоконференций.

Разработка веб—приложения для автоматизации учебных онлайн и оффлайн процессов в учреждениях дополнительного образования

8/12

Рисунок А.8 – Перечень функций системы

## ПЕРЕЧЕНЬ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

Функции для преподавателей:

- разработка учебных программ и материалов;
- назначение и проверка заданий;
- контроль успеваемости учащихся;
- проведение онлайн-занятий.

Функции для учащихся:

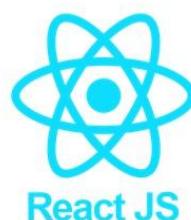
- доступ к учебным материалам 24/7;
- выполнение и отправка заданий;
- отслеживание персонального прогресса;
- общение с преподавателями.

Разработка веб-приложения для автоматизации учебных онлайн и офлайн процессов в учреждениях дополнительного образования

9/12

Рисунок А.9 – Перечень функций пользователя

## ВЫБОР СИСТЕМ ДЛЯ РАЗРАБОТКИ КЛИЕНТСКОЙ ЧАСТИ



React.js обеспечит компонентный подход и высокую производительность за счет виртуального DOM.



TypeScript добавит статическую типизацию, повысив надежность и поддерживаемость кода.

Разработка веб-приложения для автоматизации учебных онлайн и офлайн процессов в учреждениях дополнительного образования

10/12

Рисунок А.10 – Выбор систем для разработки клиентской части

## ВЫБОР СИСТЕМ ДЛЯ РАЗРАБОТКИ СЕРВЕРНОЙ ЧАСТИ



Node.js обеспечит высокую производительность и масштабируемость для обработки множества одновременных подключений.



NestJS предоставит готовые решения для сложной бизнес-логики, такой как проверка заданий и формирование отчетов.

Firebase и Google Cloud Platform были выбраны из—за их мощных возможностей и удобства использования



PostgreSQL — для структурированных данных (пользователи, курсы, задания).



MongoDB — для менее структурированного контента (чаты, комментарии, медиафайлы).

Разработка веб—приложения для автоматизации учебных онлайн и оффлайн процессов в учреждениях дополнительного образования

11/12

Рисунок А.11 – Выбор систем для разработки серверной части

## Заключение / Выводы

В результате разработки проекта были выполнены следующие задачи:

1. Изучена предметная область.
2. Изучены методы, технологии, алгоритмы и методики, значимые для рассматриваемой предметной области.
3. Проведен анализ актуальности выбранной задачи.
4. Рассмотрены системы—аналоги.
5. Составлен список функций разрабатываемого приложения.
6. Произведен выбор архитектуры разрабатываемой системы и средств реализации.

## БЛАГОДАРЮ ЗА ВНИМАНИЕ!

Разработка веб—приложения для автоматизации учебных онлайн и оффлайн процессов в учреждениях дополнительного образования

12/12

Рисунок А.12 – Итоги работы