СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc186025163)

[1 ОБЩАЯ ЧАСТЬ 5](#_Toc186025164)

[1.1 Цель разработки 5](#_Toc186025165)

[1.2 Анализ использования разработки 6](#_Toc186025166)

[1.3 Анализ методов решения 7](#_Toc186025167)

[1.4 Обзор средств программирования 8](#_Toc186025168)

[1.5 Описание языка программирования 10](#_Toc186025169)

# ВВЕДЕНИЕ

Цифровые технологии давно стали частью повседневной жизни и сильно изменили систему образования. Современные студенты всё чаще учатся вне аудитории – общаются с преподавателями онлайн, сдают задания дистанционно и проходят тестирование в интернете. Особенно вырос спрос на удобные образовательные платформы после распространения удалённого формата обучения.

Для молодых людей, которые осваивают новую профессию или совершенствуют навыки, важна возможность учиться в удобное время и в комфортном темпе. Поэтому актуально создание онлайн-ресурсов, способных заменить или дополнить традиционное обучение.

Моя платформа предлагает простое и эффективное взаимодействие между студентами и преподавателями. С помощью интуитивного интерфейса и продуманного функционала пользователи могут изучать материал и проходить тесты без необходимости личного присутствия на занятиях. Это экономит время педагогов и делает обучение гибким для студентов.

Цель проекта – разработать веб-сайт, позволяющий размещать учебные материалы и проверять знания с помощью тестов. Для этого были поставлены следующие задачи: создать понятный интерфейс, реализовать доступ к лекциям и тестам, настроить регистрацию и разделение ролей (студент/преподаватель), обеспечить надёжность и стабильность работы платформы.

Финальная версия проекта станет универсальным решением для дистанционного обучения. Он удобен как для студентов, желающих самостоятельно осваивать материал и проверять знания, так и для преподавателей, которым нужен инструмент для автоматизации процессов – от загрузки лекций до анализа успеваемости. Таким образом, платформа способна сделать образовательный процесс более эффективным и комфортным для всех участников.

## 1 ОБЩАЯ ЧАСТЬ

### 1.1 Цель разработки

Целью дипломной работы является разработка образовательного ресурса для изучения дисциплины «Архитектура компьютерных систем», предоставляющий доступ к лекциям, практическим заданиям и автоматизированному тестированию.

Основные задачи: разработать удобный пользовательский интерфейс для студентов, преподавателей и администратора; подготовить и интегрировать учебный контент: лекции, практические материалы и тесты; реализовать систему автоматической проверки тестовых заданий с сохранением результатов; обеспечить возможность управления контентом со стороны преподавателя.

### 1.2 Анализ использования разработки

Разрабатываемый образовательный ресурс предназначен для эффективной поддержки изучения дисциплины «Архитектура компьютерных систем». Сайт предоставляет доступ к структурированным учебным материалам, интерактивным заданиям и проверочным тестам, что способствует повышению качества обучения. Он оснащён удобными инструментами для преподавателей, которые могут проверять результаты тестов, добавлять дополнительные материалы. Преподаватели образовательного процесса смогут управлять содержанием сайта, добавлять или редактировать блоки информации, поддерживая его актуальность и полноту. Программный продукт будет ориентирован на две основные категории пользователей:

Студенты: имеют возможность изучать лекции, выполнять практические задания и проходить тестирование; могут просматривать свои оценки и отслеживать прогресс в обучении; используют платформу для самоподготовки и повторения пройденного материала.

Преподаватели: загружают и обновляют учебные материалы, создают и назначают задания; получают доступ к статистике успеваемости студентов для анализа эффективности обучения.

Разработанная онлайн-платформа позволяет студентам самостоятельно изучать дисциплину «Архитектура компьютерных систем», облегчает преподавателям организацию учебного процесса и ускоряет управление контентом.

### 1.3 Анализ методов решения

Для разработки образовательного ресурса, который будет соответствовать поставленным целям, необходимо выбрать оптимальные технологии и инструменты. Выбор подходящих средств зависит от требований к функциональности, масштаба проекта и доступных ресурсов.

Образовательный сайт разрабатывается для обеспечения: доступа к учебным материалам; возможности выполнения тестов и заданий; взаимодействия студентов, преподавателей и администраторов.

Возможные методы разработки:

1. Использование конструктора сайтов. Подходит для создания простых и быстро реализуемых проектов. Преимущества: интуитивно понятный интерфейс, быстрый запуск, минимальные технические навыки. Ограничения: ограниченная гибкость и возможности кастомизации. Примеры: Tilda, Wix, Squarespace.
2. Системы управления контентом (CMS). CMS, такие как WordPress, Joomla или Drupal, позволяют создавать сайты с расширенной функциональностью благодаря плагинам и темам. Преимущества: большая библиотека готовых решений, простое управление контентом, удобство для администраторов. Ограничения: необходимость настройки и возможное ограничение производительности для сложных задач.
3. Ручная разработка сайта (Custom Development). Включает использование языков программирования и фреймворков, таких как HTML, CSS, JavaScript (с фреймворками React, Vue.js), а также серверные технологии (Python с Django/Flask, PHP с Laravel, Node.js). Преимущества: высокая гибкость, возможность реализовать любые функции, полное соответствие требованиям проекта. Ограничения: требуется больше времени, ресурсов и опыта в веб-разработке.

В ходе анализа всех доступных методов разработки мной был выбран метод ручной разработки. Этот метод представляет максимальную гибкость и контроль над всеми аспектами сайта, также позволяет создать уникальный продукт. Несмотря на то, что этот способ более трудозатратен и требует знаний в области веб-разработки, но именно он позволяет создать действительно эффективный и функциональный образовательный ресурс.

### 1.4 Обзор средств программирования

При разработке образовательного ресурса важно не только определить функциональные требования к системе, но и выбрать подходящие инструменты реализации. От правильного выбора языков программирования и фреймворков зависит не только скорость и качество разработки, но и дальнейшее сопровождение, масштабируемость и безопасность продукта. Я рассмотрела основные варианты, которые могут быть использованы при создании подобной системы:

1. Языки и технологии для клиентской части (Frontend):

HTML(HyperText Markup Language) является основой любой веб-страницы. Он отвечает за структуру контента: заголовки, параграфы, изображения, формы, ссылки и другие элементы. Плюсы: обязательная часть любого сайта; простая структура, легко читается браузерами; поддерживается всеми современными браузерами. Минусы: не позволяет добавлять интерактивность или динамику без дополнительных технологий.

CSS (Cascading Style Sheets) используется для оформления и стилизации страниц. С помощью CSS можно задавать цвета, шрифты, расположение элементов, анимации и адаптивное отображение под разные устройства. Плюсы: позволяет гибко управлять внешним видом сайта; поддерживает медиа-запросы для адаптивности; совместим со всеми браузерами. Минусы: без использования препроцессоров может быть сложно, управлять большим объемом стилей.

JavaScript используется для создания интерактивности на клиентской стороне. JavaScript позволяет манипулировать элементами страницы, обрабатывать события и взаимодействовать с сервером через AJAX-запросы. Плюсы: единственный язык, который выполняется непосредственно в браузере; позволяет создавать динамические веб-приложения; большое количество библиотек. Минусы: может замедлить работу страницы при плохой оптимизации; различия в поведении между браузерами требуют тестирования.

React: Один из самых популярных JavaScript-фреймворков для создания интерфейсов. React позволяет создавать сложные пользовательские интерфейсы с минимальным количеством кода благодаря компонентному подходу. Он идеально подходит для создания сложных веб-приложений с высоким уровнем взаимодействия. Плюсы: компонентный подход упрощает разработку и повторное использование кода; поддерживает реактивность и односторонний поток данных; множество готовых решений; отлично подходит для динамических приложений с высоким уровнем взаимодействия. Минусы: требует знания JSX и хорошего понимания JavaScript; для полноценной разработки нужны дополнительные библиотеки.

Vue.js: Легковесный и гибкий JavaScript-фреймворк, который обеспечивает более простой подход к разработке интерфейсов по сравнению с React. Vue.js позволяет создавать динамичные страницы с реактивностью, но имеет более простую и понятную документацию для новичков. Плюсы: легко внедряется в существующие проекты; простой синтаксис и понятная документация; реактивные данные. Минусы: меньше возможностей по сравнению с React при создании крупных приложений; меньше готовых решений.

2. Языки и фреймворки для серверной части (Backend)

Python: Один из самых популярных языков для веб-разработки благодаря своей читаемости, простоте и широкому применению. Python идеально подходит для серверной логики, работы с базами данных, обработки файлов и реализации бизнес-логики. Плюсы: простой и понятный синтаксис; большое количество библиотек и фреймворков; отлично подходит для работы с данными, машинного обучения и API. Минусы: низкая производительность по сравнению с компилируемыми языками; не лучший выбор для CPU-тяжёлых задач.

Django: Это мощный фреймворк для Python и для быстрого создания веб-приложений. Django предоставляет всё необходимое для разработки, включая работу с базами данных, систему аутентификации, обработку форм и административную панель. Он помогает быстро разрабатывать сложные и масштабируемые приложения. Плюсы: встроенные инструменты: ORM, аутентификация, административная панель; высокая безопасность; подходит для сложных проектов с большими объемами данных. Минусы: большая «весовая» нагрузка – не самый подходящий выбор для простых сайтов; требует глубокого понимания архитектуры.

PHP: Это один из самых популярных языков программирования для веб-разработки, особенно в области создания динамичных веб-приложений. Он активно используется для серверной логики, обработки форм, работы с базами данных и создания динамических веб-страниц. Плюсы: простота освоения; поддерживается большинством хостингов; широкая экосистема и большое количество готовых решений. Минусы: исторически не очень строгая типизация; разные стандарты в разных версиях.

Laravel: Это один из самых популярных PHP-фреймворков, который предоставляет множество встроенных инструментов для упрощения разработки и улучшения качества кода. Этот фреймворк активно используется для создания веб-приложений благодаря своим удобным и мощным возможностям. Плюсы: удобная система маршрутов; встроенная поддержка авторизации, очередей, рассылок; чистый и понятный синтаксис; хорошо документирован. Минусы: по сравнению с Node.js, медленнее; требует некоторых технических навыков для тонкой настройки.

Node.js: Платформа для выполнения JavaScript на сервере. Она позволяет создавать быстрые и масштабируемые веб-приложения, основанные на событиях и асинхронных запросах. Node.js используется для разработки серверной части веб-приложений, API и обработчиков. Плюсы: асинхронная, событийно-ориентированная архитектура; возможность использовать JavaScript на всех уровнях (fullstack); быстрая обработка запросов. Минусы: не лучший выбор для CPU-интенсивных задач; требует аккуратной организации кода.

После анализа различных языков программирования и фреймворков для разработки образовательного ресурса был выбран стек технологий, включающий HTML, CSS, JavaScript для клиентской части и PHP для серверной стороны. Такой выбор обусловлен несколькими ключевыми факторами: HTML и CSS являются базовыми инструментами создания веб-страниц и отвечают за структуру и внешний вид интерфейса. Использование современных возможностей CSS (например, Flexbox) позволяет создать адаптивный и удобный дизайн, который корректно отображается на разных устройствах. JavaScript добавляет интерактивность к статичным страницам: обработка форм, динамическое обновление контента без перезагрузки страницы, проверка тестов в реальном времени. Он обеспечивает необходимый уровень взаимодействия между пользователем и системой, делая обучение более эффективным и увлекательным. PHP выбран в качестве серверного языка благодаря его широкому распространению, простоте внедрения и хорошей совместимости с базами данных. Он отлично подходит для реализации таких функций, как авторизация пользователей, работа с тестами, хранение результатов и управление контентом со стороны преподавателей. PHP не требует сложной настройки сервера и легко развертывается на большинстве хостингов, что делает его практичным решением для проектов средней сложности.

Выбранный стек технологий полностью соответствует поставленным задачам и позволяет реализовать все ключевые функции образовательного ресурса: от предоставления учебного материала до автоматизации проверки знаний и управления пользователями.

### 1.5 Описание языка программирования

HTML (HyperText Markup Language) – это стандартный язык разметки, используемый для создания структуры веб-страниц. Основной функцией HTML является определение структуры и содержания страницы: заголовков, параграфов, изображений, таблиц, списков и других элементов.

Основные функции:

* Создание структуры страницы: HTML позволяет задавать такие элементы, как текст, изображения, видео, формы и другие компоненты, которые составляют содержимое веб-страницы;
* Теги и атрибуты, Гиперссылки;
* Встраивание мультимедиа: HTML поддерживает встраивание аудио, видео, а также поддержку различных форматов мультимедийных данных;

Формы для ввода данных: HTML предоставляет теги для создания форм, которые позволяют пользователям вводить данные, такие как текст, числа, даты и т.д.

CSS (Cascading Style Sheets) используется для описания внешнего вида HTML-элементов. С помощью CSS можно задавать стили, такие как шрифты, цвета, отступы, выравнивание и анимацию.

Основные функции:

* Стилизация элементов: CSS позволяет изменять внешний вид элементов на странице, например, задавать цвет текста, фона, размер шрифта, рамки и другие параметры.
* Адаптивный дизайн: CSS предоставляет средства для создания адаптивного дизайна, который изменяет внешний вид страницы в зависимости от размера экрана устройства (мобильных телефонов, планшетов, компьютеров).
* Макетирование: CSS предлагает различные способы расположения элементов на странице, включая flexbox и grid-системы, которые позволяют создавать сложные макеты.
* Анимации и переходы: CSS поддерживает создание плавных анимаций и переходов, что помогает улучшить пользовательский опыт и визуальную привлекательность.
* Использование фреймворков: CSS-фреймворки, такие как Bootstrap и Tailwind CSS, помогают ускорить разработку и обеспечивают готовые компоненты для UI (покрытие всех аспектов дизайна, включая кнопки, формы, модальные окна и т.д.).

JavaScript (с фреймворком React) – это язык программирования, который используется для создания интерактивности на веб-страницах. JavaScript позволяет манипулировать элементами HTML-документа и взаимодействовать с сервером, что делает веб-приложение более динамичным.

Основные функции:

* Манипуляция DOM (Document Object Model): JavaScript позволяет изменять структуру и содержание веб-страницы в реальном времени. Это позволяет обновлять данные, элементы интерфейса и выполнять различные действия без перезагрузки страницы.
* Обработка событий: JavaScript позволяет реагировать на действия пользователя (например, клики, ввод данных, прокрутка и т.д.).
* Работа с асинхронными запросами: JavaScript использует технологии, такие как AJAX или Fetch API, для асинхронных запросов к серверу, что позволяет обновлять данные на странице без её перезагрузки.
* Фреймворк React:
  + Компонентная архитектура: React помогает строить приложения, разделяя интерфейс на повторно используемые компоненты. Это упрощает управление состоянием и обновления интерфейса.
  + Виртуальный DOM: React использует виртуальный DOM для оптимизации изменений в реальном DOM, что повышает производительность при рендеринге больших и сложных интерфейсов.
  + Интерактивность: React идеально подходит для создания динамичных и интерактивных интерфейсов с минимальными усилиями, что важно для образовательных ресурсов с тестами и заданиями.

PHP с фреймворком Laravel – это серверный язык программирования, который широко используется для создания динамичных веб-сайтов. Он взаимодействует с сервером и базой данных для обработки запросов, отправки данных и генерации веб-страниц.

Основные функции PHP:

* Работа с базами данных: PHP позволяет взаимодействовать с различными СУБД, включая MySQL, PostgreSQL и SQLite. Это важно для образовательного ресурса, так как он требует хранения и обработки большого объема данных, таких как результаты тестов.
* Обработка форм: PHP используется для обработки данных, отправляемых через формы, например, для регистрации пользователей, отправки тестов или работы с результатами.
* Сеансовое управление: PHP предоставляет функции для управления сессиями и куки, что важно для реализации авторизации пользователей.
* Безопасность: PHP включает в себя механизмы защиты от атак, таких как SQL-инъекции, XSS и CSRF.

Фреймворк Laravel:

* Структурированность кода: Laravel использует архитектуру MVC (Model-View-Controller), что помогает организовать код, обеспечивая его поддержку и масштабируемость.
* ORM (Eloquent): Laravel предоставляет удобный инструмент для работы с базами данных через ORM Eloquent, который позволяет легко взаимодействовать с базой данных без необходимости писать сложные SQL-запросы.
* Миграции и сиды: Laravel поддерживает миграции базы данных, что позволяет легко управлять изменениями в структуре данных, а также сиды, которые позволяют наполнить базу данных тестовыми данными.
* Безопасность: Laravel включает встроенные механизмы для защиты от основных уязвимостей, таких как CSRF и XSS, а также функции для хэширования паролей и безопасной аутентификации пользователей.
* Шаблонизация: В Laravel используется шаблонизатор Blade, который позволяет легко создавать динамичные HTML-шаблоны и улучшать разделение логики и представления.