**Разработка сайта библиотеки**

3адание

Отзыв руководителя

Рецензия

Введение

1 Теоретические основы построения сайтов

1.1 Понятие сайта. Виды сайтов

1.2 Структура сайта

1.3 Анализ потребностей библиотеки и ее пользователей

2 Анализ средств создания сайта

2.1 Языки разметки и стили

2.2 Языки программирования

2.3Фреймворки и библиотеки

2.4Системы управления контентом (CMS)

2.5Графические редакторы и дизайн-инструменты

2.6Базы данных

2.7 Хостинг-провайдеры и доменные регистраторы

3 Разработка сайта

3.1 Описание предметной области и требований к сайту

3.2 Выбор технологического стека для разработки сайта

3.3 Создание структуры сайта

  3.4 Создание базы данных и реализация функционала для хранения и управления данными

3.5 Разработка пользовательского интерфейса и его компонентов

3.6 Разработка функционала для поиска, фильтрации

3.7 Оптимизация и производительность

…

Заключение

Список использованных источников

Введение

В современном мире информационные технологии и интернет играют важнейшую роль в различных аспектах жизни, включая образование и доступ к информации. Библиотеки, будучи хранилищами знаний и культуры, также активно интегрируются в цифровое пространство, предоставляя свои услуги через интернет. Создание веб-сайта для библиотеки открывает новые возможности для пользователей, позволяя им получать доступ к разнообразным ресурсам в любое время и из любого места. Это особенно актуально в условиях цифровой трансформации и удаленного обучения, когда пользователи предпочитают искать и получать информацию онлайн.

Цель данной дипломной работы заключается в разработке веб-сайта библиотеки, который будет соответствовать современным требованиям и удовлетворять потребности пользователей. Основными задачами сайта станут обеспечение удобного доступа к библиотечным ресурсам, предоставление функционала для поиска и бронирования книг, а также организация обратной связи с пользователями. Разработка такого сайта требует глубокого понимания теоретических основ построения веб-ресурсов, анализа существующих инструментов и технологий, а также учета специфических требований библиотеки и её аудитории.

1. Теоретические основы построения сайтов

1.1 Понятие сайта. Виды сайтов

Сайт представляет собой совокупность веб-страниц, объединённых общей тематикой и структурой, доступных пользователям через сеть Интернет. Веб-страница – это документ, который может содержать текст, изображения, видео, аудио и другие мультимедийные элементы. Сайт функционирует на веб-сервере и доступен через уникальный адрес (URL). Основная цель создания сайта – предоставление информации или услуг пользователям.

Сайты можно классифицировать по различным критериям, таким как назначение, содержание, целевая аудитория и используемые технологии. Рассмотрим основные виды сайтов:

Информационные сайты: предназначены для предоставления пользователям актуальной информации. К таким сайтам относятся новостные порталы, блоги, энциклопедии и образовательные ресурсы. Примеры: Википедия, CNN, BBC.

Коммерческие сайты: ориентированы на продвижение и продажу товаров и услуг. К ним относятся интернет-магазины, корпоративные сайты и платформы для электронной коммерции. Примеры: Amazon, eBay, Alibaba.

Социальные сети: предоставляют пользователям возможности для общения и обмена информацией. Примеры: Facebook, Instagram, Twitter, ВКонтакте.

Форумы: платформы для обсуждения различных тем, где пользователи могут обмениваться мнениями и задавать вопросы. Примеры: Reddit, Stack Overflow.

Образовательные сайты: ресурсы, предлагающие образовательные материалы и курсы. Примеры: Coursera, Khan Academy, edX.

Библиотеки: специализированные сайты, предоставляющие доступ к каталогам книг и другим информационным ресурсам. Примеры: Project Gutenberg, Google Books.

Каждый тип сайта имеет свои особенности и требования к разработке, что необходимо учитывать при создании веб-ресурса.

1.2 Структура сайта

Структура сайта представляет собой организацию его содержимого и элементов интерфейса, что обеспечивает удобную навигацию и легкость поиска информации для пользователей. Правильная структура сайта способствует улучшению пользовательского опыта и повышению эффективности его использования. Основные элементы структуры сайта включают:

Главная страница: стартовая точка для пользователей, содержащая общую информацию о сайте и ссылки на основные разделы. На главной странице обычно размещаются новости, анонсы, актуальные предложения и основные навигационные элементы.

Разделы и подразделы: логически организованные части сайта, группирующие страницы по тематикам или функциям. Разделы могут включать информацию о компании, продуктах, услугах, новостях и т.д.

Меню навигации: элемент интерфейса, позволяющий пользователям быстро переходить между разделами и страницами сайта. Меню навигации может быть горизонтальным или вертикальным, статическим или выпадающим.

Футер: нижняя часть страницы, содержащая дополнительную информацию и ссылки на важные разделы сайта, такие как контактная информация, политика конфиденциальности, условия использования и т.д.

Форма поиска: инструмент, позволяющий пользователям находить нужную информацию по ключевым словам. Форма поиска обычно размещается в верхней части страницы или в боковой панели.

Личный кабинет пользователя: раздел, доступный после авторизации, где пользователи могут управлять своими данными, настройками и действиями на сайте. Личный кабинет может включать историю заказов, бронирования, настройки профиля и другие функции.

Грамотное проектирование структуры сайта является ключевым этапом его разработки, так как от этого зависит удобство использования и удовлетворенность пользователей.

1.3 Анализ потребностей библиотеки и ее пользователей

Перед началом разработки сайта библиотеки важно провести детальный анализ потребностей как самой библиотеки, так и её пользователей. Это позволит создать ресурс, который будет функциональным, удобным и востребованным. Рассмотрим основные потребности библиотеки и её аудитории:

Потребности библиотеки:

Каталогизация ресурсов: организация и хранение информации о книгах, журналах, электронных изданиях и других материалах. Сайт должен предоставлять возможность быстрого и удобного добавления новых позиций в каталог и редактирования существующих записей.

Управление пользователями: регистрация и учет пользователей, контроль доступа к ресурсам и услугам. Сайт должен обеспечивать простую и безопасную регистрацию, авторизацию и управление учетными записями пользователей.

Обеспечение доступа к цифровым ресурсам: предоставление пользователям доступа к электронным книгам, статьям, аудиокнигам и другим материалам. Это включает организацию онлайн-чтения, скачивания и прослушивания ресурсов.

Потребности пользователей:

Поиск информации: удобный и быстрый доступ к каталогу библиотеки для поиска нужных книг и других ресурсов. Функционал поиска должен поддерживать различные критерии (по автору, названию, жанру и т.д.) и фильтры.

Бронирование книг: возможность забронировать книги для последующего посещения библиотеки и получения их в пользование. Система бронирования должна быть интуитивно понятной и предоставлять информацию о доступности книг.

Просмотр истории заказов и бронирований: доступ к информации о ранее заказанных и забронированных книгах, а также статус текущих заказов. Это позволяет пользователям отслеживать свои действия и планировать дальнейшее использование библиотеки.

Обратная связь: возможность отправлять отзывы, предложения и задавать вопросы администрации библиотеки. Сайт должен включать форму обратной связи, которая обеспечит оперативное взаимодействие с пользователями и помощь в решении их вопросов.

Анализ потребностей библиотеки и её пользователей является важным этапом в процессе разработки сайта, так как позволяет определить ключевые функции и требования к ресурсу, что в конечном итоге приведет к созданию удобного и эффективного веб-сайта.

2. Анализ средств создания сайта

Создание сайта – это многогранный процесс, который включает в себя выбор и использование различных инструментов и технологий. Этот процесс начинается с планирования и проектирования структуры сайта, выбора языков разметки и программирования, и завершается внедрением сайта на сервер и его поддержкой. В данном разделе мы рассмотрим основные средства, используемые для создания веб-сайтов, включая языки разметки и стили, языки программирования, фреймворки и библиотеки, системы управления контентом (CMS), графические редакторы и дизайн-инструменты, базы данных, а также хостинг-провайдеры и доменные регистраторы.

2.1 Языки разметки и стили

Языки разметки и стилизации являются основой для создания структуры и внешнего вида веб-страниц. Они определяют, как контент будет отображаться в браузере.

HTML (HyperText Markup Language):

HTML является стандартным языком разметки для создания веб-страниц. Он используется для описания структуры и содержимого веб-документов. HTML позволяет создавать заголовки, абзацы, ссылки, изображения, таблицы и другие элементы. Основные теги HTML включают <html>, <head>, <body>, <div>, <p>, <a> и другие. HTML5, последняя версия языка, добавила новые элементы и атрибуты, такие как <header>, <footer>, <article>, <section>, <nav>, которые улучшают семантику и структуру веб-документов.

CSS (Cascading Style Sheets):

CSS используется для стилизации HTML-документов. Он позволяет задавать внешний вид и оформление элементов, таких как цвет, шрифт, отступы, границы и расположение. CSS отделяет структуру документа от его презентации, что позволяет улучшить удобство поддержки и обновления кода. CSS3, последняя версия, добавила множество новых возможностей, включая анимации, переходы, градиенты, гибкие сетки (Flexbox) и сетки CSS Grid.

SASS/SCSS и LESS:

SASS и LESS являются препроцессорами CSS, которые добавляют такие возможности, как переменные, вложенные правила, миксины и функции. Они упрощают написание и поддержку CSS-кода, делая его более организованным и модульным. Препроцессоры компилируют свой код в обычный CSS, который затем используется в веб-документе.

JavaScript и библиотеки для анимации:

JavaScript также играет роль в стилизации через динамическое управление DOM (Document Object Model). Библиотеки, такие как jQuery, GSAP (GreenSock Animation Platform) и другие, позволяют создавать сложные анимации и интерактивные элементы на веб-страницах, улучшая пользовательский опыт.

2.2 Языки программирования

Языки программирования являются основным инструментом для разработки логики и функциональности веб-сайтов. Они используются как на стороне клиента, так и на стороне сервера.

JavaScript:

JavaScript является основным языком программирования на стороне клиента. Он позволяет создавать интерактивные элементы, такие как всплывающие окна, валидация форм, динамическое обновление контента без перезагрузки страницы (AJAX). Современные библиотеки и фреймворки, такие как React, Angular и Vue.js, расширяют возможности JavaScript и упрощают разработку сложных пользовательских интерфейсов.

PHP:

PHP (Hypertext Preprocessor) – это популярный серверный язык программирования, используемый для создания динамических веб-страниц и управления серверной логикой. Он часто используется в сочетании с базами данных MySQL и MariaDB. PHP прост в изучении и использовании, что делает его популярным выбором для разработки веб-сайтов. К популярным CMS, таким как WordPress, Joomla и Drupal, также относится PHP.

Python:

Python – это высокоуровневый язык программирования, известный своей простотой и читабельностью. В веб-разработке Python часто используется в сочетании с фреймворками Django и Flask. Django обеспечивает высокую скорость разработки и включает в себя многие встроенные функции, такие как ORM (Object-Relational Mapping) и система администрирования. Flask является более легковесным фреймворком, подходящим для небольших проектов и микросервисов.

Ruby:

Ruby – это язык программирования, известный своей простотой и элегантностью. Ruby on Rails (или просто Rails) – это фреймворк для веб-разработки, основанный на языке Ruby. Rails обеспечивает высокую продуктивность разработки благодаря таким концепциям, как DRY (Don't Repeat Yourself) и Convention over Configuration.

Java:

Java – это язык программирования общего назначения, часто используемый в корпоративных веб-приложениях. Фреймворки, такие как Spring и JavaServer Faces (JSF), упрощают разработку веб-приложений на Java. Java отличается высокой производительностью и масштабируемостью, что делает его подходящим для крупных проектов.

C# и ASP.NET:

C# – это язык программирования, разработанный компанией Microsoft. ASP.NET – это фреймворк для веб-разработки на платформе .NET, который использует C#. ASP.NET обеспечивает высокую производительность и интеграцию с другими продуктами Microsoft, такими как SQL Server и Azure.

2.3 Фреймворки и библиотеки

Фреймворки и библиотеки облегчают разработку веб-приложений, предоставляя готовые компоненты и инструменты для решения типичных задач. Они ускоряют процесс разработки и улучшают качество кода.

Frontend-фреймворки и библиотеки:

React: библиотека для создания пользовательских интерфейсов, разработанная Facebook. React использует компонентный подход и виртуальный DOM для высокой производительности.

Angular: фреймворк для создания одностраничных приложений (SPA), разработанный Google. Angular использует TypeScript и предоставляет множество встроенных инструментов и компонентов.

Vue.js: прогрессивный фреймворк для создания пользовательских интерфейсов. Vue.js сочетает в себе лучшие качества React и Angular, предлагая гибкость и простоту.

Backend-фреймворки:

Django: фреймворк для веб-разработки на Python, известный своей скоростью разработки и множеством встроенных функций.

Flask: микрофреймворк на Python, подходящий для небольших проектов и микросервисов.

Ruby on Rails: фреймворк на Ruby, известный своей простотой и элегантностью.

Spring: фреймворк на Java, подходящий для корпоративных приложений.

ASP.NET Core: фреймворк на C#, обеспечивающий высокую производительность и интеграцию с экосистемой Microsoft.

CSS-фреймворки и библиотеки:

Bootstrap: популярный CSS-фреймворк, разработанный Twitter. Bootstrap предоставляет множество готовых компонентов и стилей, упрощая создание адаптивных веб-сайтов.

Tailwind CSS: утилитарный CSS-фреймворк, позволяющий создавать индивидуальные стили с использованием классов.

Bulma: современный CSS-фреймворк, основанный на Flexbox, который обеспечивает простоту и гибкость при создании адаптивных интерфейсов.

2.4 Системы управления контентом (CMS)

CMS (Content Management Systems) – это платформы, которые позволяют создавать и управлять веб-сайтами без необходимости написания кода. CMS предоставляют удобные интерфейсы для добавления и редактирования контента, управления пользователями и настройки сайта.

WordPress:

WordPress – это самая популярная CMS в мире, используемая для создания более 40% всех веб-сайтов. WordPress предоставляет широкий выбор тем и плагинов, которые позволяют легко настраивать внешний вид и функциональность сайта. Он подходит как для блогов и новостных сайтов, так и для интернет-магазинов и корпоративных сайтов.

Joomla:

Joomla – это мощная и гибкая CMS, которая позволяет создавать сложные веб-сайты и порталы. Joomla предлагает множество расширений и шаблонов, что делает её подходящей для различных типов проектов, включая социальные сети, форумы и интернет-магазины.

Drupal:

Drupal – это CMS, известная своей масштабируемостью и безопасностью. Она используется для создания крупных и сложных веб-сайтов, таких как государственные порталы, образовательные ресурсы и корпоративные сайты. Drupal предоставляет широкие возможности для настройки и расширения функциональности через модули.

Magento:

Magento – это CMS, специально разработанная для создания интернет-магазинов. Magento предлагает мощные инструменты для управления товарами, заказами и клиентами, а также поддерживает интеграцию с различными платёжными системами и службами доставки.

Shopify:

Shopify – это облачная платформа для создания интернет-магазинов. Shopify позволяет быстро и легко настроить интернет-магазин, не требуя глубоких технических знаний. Платформа предоставляет множество тем и приложений для расширения функциональности.

2.5 Графические редакторы и дизайн-инструменты

Графические редакторы и дизайн-инструменты используются для создания макетов, дизайна пользовательских интерфейсов и разработки графических элементов для веб-сайтов.

Adobe Photoshop:

Adobe Photoshop – это мощный графический редактор, используемый для редактирования растровых изображений и создания графики для веб-сайтов. Photoshop предлагает широкий набор инструментов для обработки изображений, создания макетов и прототипирования интерфейсов.

Adobe Illustrator:

Adobe Illustrator – это инструмент для создания векторной графики. Он используется для разработки логотипов, иконок, иллюстраций и других графических элементов. Векторная графика имеет преимущество в том, что её можно масштабировать без потери качества.

Sketch:

Sketch – это популярный инструмент для дизайна пользовательских интерфейсов и прототипирования, используемый в основном на macOS. Sketch предлагает удобные инструменты для создания макетов, работы с символами и компонентами, а также интеграцию с другими инструментами для разработки.

Figma:

Figma – это облачный инструмент для дизайна и прототипирования, который позволяет командам работать совместно в реальном времени. Figma предлагает мощные инструменты для создания макетов, прототипов и анимаций, а также поддержку плагинов для расширения функциональности.

InVision:

InVision – это платформа для прототипирования и разработки пользовательских интерфейсов. InVision позволяет создавать интерактивные прототипы, проводить тестирование с пользователями и управлять дизайном проекта. Платформа также поддерживает интеграцию с другими инструментами, такими как Sketch и Photoshop.

2.6 Базы данных

Базы данных используются для хранения и управления информацией на веб-сайтах. Они позволяют организовать данные в структурированном виде и обеспечивают быстрый доступ к ним.

MySQL:

MySQL – это реляционная база данных, которая является одной из самых популярных в мире. MySQL используется в сочетании с языком программирования PHP для создания динамических веб-сайтов. Она предлагает высокую производительность, надёжность и масштабируемость.

PostgreSQL:

PostgreSQL – это мощная реляционная база данных с открытым исходным кодом, известная своей расширяемостью и поддержкой сложных запросов. PostgreSQL поддерживает широкий спектр функций, таких как индексы, транзакции, триггеры и функции на стороне сервера, что делает её подходящей для сложных и требовательных приложений.

SQLite:

SQLite – это легковесная встраиваемая база данных, которая не требует настройки и управления сервером. SQLite часто используется для небольших проектов, мобильных приложений и прототипирования. Она хранит все данные в одном файле и обеспечивает быстрый доступ к информации.

MongoDB:

MongoDB – это нереляционная (NoSQL) база данных, которая хранит данные в виде документов JSON-подобных объектов. MongoDB подходит для приложений, требующих высокой производительности и гибкости в управлении данными. Она используется для хранения больших объёмов данных и поддержки горизонтального масштабирования.

Redis:

Redis – это база данных в памяти, используемая для кэширования и управления сеансами. Redis обеспечивает высокую производительность и низкую задержку доступа к данным, что делает её подходящей для использования в качестве кэша, очередей сообщений и систем отслеживания в реальном времени.

2.7 Хостинг-провайдеры и доменные регистраторы

Для того чтобы сайт стал доступен пользователям через Интернет, необходимо разместить его на сервере хостинг-провайдера и зарегистрировать доменное имя.

Хостинг-провайдеры:

Shared Hosting: Общий хостинг – это наиболее доступный вариант, при котором несколько веб-сайтов размещаются на одном сервере. Это экономичный вариант для небольших сайтов с низкой нагрузкой, но он имеет ограниченные ресурсы и производительность.

VPS Hosting (Virtual Private Server): Виртуальный частный сервер предоставляет больше ресурсов и возможностей по сравнению с общим хостингом. VPS хостинг разделяет физический сервер на несколько виртуальных машин, каждая из которых работает независимо. Это подходящий вариант для средних и крупных сайтов с умеренной нагрузкой.

Dedicated Hosting: Выделенный сервер предоставляет все ресурсы физического сервера для одного веб-сайта. Это дорогой, но мощный вариант, который подходит для крупных проектов с высокой нагрузкой и особыми требованиями к безопасности и производительности.

Cloud Hosting: Облачный хостинг использует ресурсы нескольких серверов для обеспечения высокой доступности и масштабируемости. Он предлагает гибкость в управлении ресурсами и оплату за использование. Это подходящий вариант для проектов с переменной нагрузкой и высокими требованиями к отказоустойчивости.

Managed Hosting: Управляемый хостинг предоставляет услуги по управлению сервером, обновлению программного обеспечения, резервному копированию и безопасности. Это позволяет сосредоточиться на разработке и управлении сайтом, оставив технические задачи хостинг-провайдеру.

Доменные регистраторы:

Доменное имя – это уникальный адрес сайта в Интернете. Доменные регистраторы предоставляют услуги по регистрации и управлению доменными именами.

GoDaddy: Один из крупнейших доменных регистраторов в мире, предлагающий широкий спектр услуг, включая регистрацию доменов, веб-хостинг и SSL-сертификаты.

Namecheap: Популярный регистратор доменов, известный своими конкурентоспособными ценами и качественным обслуживанием клиентов.

Bluehost: Помимо регистрации доменов, Bluehost предлагает услуги веб-хостинга и управляемого хостинга WordPress.

Google Domains: Услуга регистрации доменов от Google, предлагающая простую интеграцию с другими сервисами Google, такими как G Suite и Google Cloud.

HostGator: Регистратор доменов и веб-хостинг провайдер, предлагающий различные планы хостинга, включая общий хостинг, VPS и выделенные серверы.

Выбор хостинг-провайдера и регистратора доменного имени зависит от специфики проекта, требований к производительности, бюджету и предпочтений разработчиков. Правильный выбор этих сервисов играет ключевую роль в успешной работе и доступности веб-сайта.

3 Разработка сайта

Разработка веб-сайта библиотеки представляет собой многослойный процесс, охватывающий несколько ключевых этапов, каждый из которых направлен на создание удобного и функционального ресурса. На первоначальном этапе важно определить цели и задачи сайта, а также детально изучить требования, предъявляемые к нему со стороны пользователей и администрации библиотеки. Это позволяет чётко сформулировать концепцию проекта и понять, какие функции и возможности должны быть реализованы.

Выбор технологического стека для разработки сайта играет решающую роль, так как от этого зависит стабильность, производительность и масштабируемость будущего ресурса. Не менее важным является проектирование структуры сайта, которое включает разработку логической схемы, определение основных разделов и страниц, а также создание прототипов интерфейсов, обеспечивающих удобную навигацию для пользователей.

Создание базы данных и реализация функционала для хранения и управления данными представляют собой фундаментальную часть процесса разработки. Это включает в себя проектирование структуры базы данных, определение таблиц и связей между ними, а также разработку механизмов для работы с данными.

Особое внимание уделяется разработке пользовательского интерфейса, который должен быть интуитивно понятным и адаптивным, обеспечивая корректное отображение на различных устройствах. Кроме того, важным аспектом является разработка функционала для поиска и фильтрации информации, что позволяет пользователям быстро находить нужные ресурсы.

На заключительном этапе проводится оптимизация сайта, направленная на улучшение его производительности, уменьшение времени загрузки страниц и обеспечение надежной работы под нагрузкой. Весь процесс разработки требует тщательного планирования и выполнения для создания качественного и эффективного веб-ресурса, который будет удовлетворять потребности пользователей и соответствовать современным стандартам.

3.1 Описание предметной области и требований к сайту

Предметная область

Предметная область данного проекта – это библиотека, предоставляющая доступ к широкому спектру книг и других печатных материалов в электронном формате. Интернет-библиотека призвана удовлетворять потребности пользователей, предоставляя удобный доступ к ресурсам, упрощая процесс поиска и бронирования книг, а также обеспечивая взаимодействие между библиотекой и её пользователями.

Основные элементы предметной области:

Книги и печатные материалы: основное содержимое библиотеки, включающее книги, журналы, статьи и другие материалы, доступные в электронном виде.

Пользователи: читатели, которые могут просматривать, искать, бронировать книги, оставлять отзывы и предлагать свои пожелания.

Администраторы: сотрудники библиотеки, ответственные за управление контентом, пользователями и обработку запросов на бронирование.

Требования к сайту

Для успешной реализации проекта интернет-библиотеки, сайт должен соответствовать следующим функциональным и нефункциональным требованиям:

Функциональные требования:

* Регистрация и авторизация пользователей:
  + Возможность создания нового аккаунта.
  + Авторизация существующих пользователей.
* Управление профилем пользователя:
  + Просмотр и редактирование личной информации.
  + Просмотр статистики прочитанных книг.
* Каталог книг:
  + Отображение списка доступных книг.
  + Фильтрация и сортировка книг по различным критериям (автор, жанр, год издания и т.д.).
  + Просмотр детальной информации о каждой книге (описание, обложка, автор, дата издания и т.д.).
* Поиск книг:
  + Поиск по ключевым словам.
  + Фильтрация результатов поиска.
  + Бронирование книг:
* Обратная связь:
  + Форма для отправки отзывов и предложений.
  + Возможность задать вопросы администрации библиотеки.
  + Нефункциональные требования:
* Удобство использования:
  + Интуитивно понятный и удобный интерфейс.
* Производительность:
  + Быстрая загрузка страниц.
  + Оптимизированный код и изображения для минимизации времени загрузки.
* Безопасность:
  + Защита личных данных пользователей.
  + Защита от взломов и атак (например, SQL-инъекции, XSS-атаки).
* Масштабируемость:
  + Возможность добавления новых функций и расширения функционала без значительных изменений в архитектуре сайта.
* Надёжность:
  + Минимизация времени простоя и сбоев в работе сайта.
  + Регулярное резервное копирование данных.

Таким образом, сайт интернет-библиотеки должен обеспечивать полный спектр функций для удобного доступа к библиотечным ресурсам, предоставлять пользователям возможность эффективно взаимодействовать с библиотекой, а также соответствовать современным стандартам безопасности, производительности и удобства использования.

3.2 Выбор технологического стека для разработки сайта

При разработке веб-сайта библиотеки важно выбрать технологический стек, который обеспечит стабильность, производительность и масштабируемость проекта. Основываясь на предоставленных данных, для проекта были выбраны следующие технологии:

Целевая платформа

Сайт разрабатывается для корректной работы на ПК с возможностью адаптации на другие платформы в будущем. Это обеспечит масштабируемость проекта и его доступность для более широкой аудитории.

Языки программирования и разметки

Для фронтенда и бэкенда используются следующие языки:

HTML: используется для создания структуры веб-страниц.

CSS: отвечает за стилизацию и внешний вид элементов.

JavaScript: добавляет интерактивность и динамическое поведение на страницах.

PHP: серверный язык программирования, используемый для обработки запросов и управления данными.

Фреймворки и библиотеки

Для улучшения разработки и стилизации использованы следующие инструменты:

Bootstrap: CSS-фреймворк, который упрощает создание адаптивного и современного дизайна.

Анализ структуры страниц проекта не выявил использование других фреймворков.

Система управления базами данных (DBMS)

Для хранения и управления данными выбрана база данных MySQL, используемая через MyAdmin в OpenServer. Это популярный выбор для веб-приложений, обеспечивающий надежность и производительность.

Система управления контентом (CMS)

Использование CMS не планируется, что позволяет создать более кастомизированное решение, точно соответствующее требованиям проекта.

Хостинг и домен

На начальном этапе разработки проект размещается на локальном сервере. В дальнейшем планируется использовать облачный хостинг или выделенный сервер, что обеспечит высокую доступность и масштабируемость ресурса.

Инструменты для дизайна и прототипирования

Для разработки интерфейса и графических элементов используются:

Adobe Photoshop: для создания и редактирования логотипов и изображений.

Bootstrap: для стилизации интерфейса.

Интеграция и API

На данном этапе интеграция с внешними сервисами и API не планируется, что упрощает архитектуру проекта.

Безопасность

Для обеспечения безопасности данных пользователей используются следующие меры:

Шифрование паролей: на данный момент используется алгоритм MD5. В дальнейшем рекомендуется перейти на более надежные алгоритмы шифрования, такие как bcrypt или Argon2, для повышения безопасности.

Другие инструменты и технологии

Дополнительные инструменты и технологии в данном проекте не используются.

Таким образом, выбранный технологический стек обеспечивает выполнение всех необходимых функций для разработки сайта библиотеки, а также закладывает основу для его дальнейшего масштабирования и улучшения.

3.3 Создание структуры сайта

Создание структуры сайта является важным этапом разработки веб-сайта библиотеки, так как от нее зависит удобство навигации, доступность информации и общее впечатление пользователей. Структура сайта должна быть интуитивно понятной и функциональной, обеспечивая быстрый доступ к основным разделам и функциям. В данном разделе описывается структура сайта библиотеки, основанная на изученном проекте.

Главная страница

Главная страница является стартовой точкой для всех пользователей. Она включает в себя следующие элементы:

* Список всех книг: Все книги представлены с пагинацией по 10 книг на странице, что позволяет удобно просматривать большой объем данных.
* Фильтры: Пользователи могут фильтровать книги по странам и жанрам, что облегчает поиск нужного материала.
* Поле поиска: Обеспечивает возможность поиска книг по названию или автору, делая процесс поиска быстрым и удобным.
* Хедер: Включает логотип и название сайта, а также кнопку авторизации. Если пользователь авторизован, отображаются кнопки профиля и выхода из аккаунта. Для администратора доступна кнопка перехода на административную страницу.
* Футер: Содержит логотип и название сайта, а также ссылки на страницы "О проекте" и "Обратная связь".

Каталог книг

Каталог книг организован следующим образом:

* Список книг: Отображается список всех книг с возможностью фильтрации и поиска.
* Пагинация: Книги разбиты на страницы по 10 элементов для удобной навигации.

Детали книги

Страница с детальной информацией о книге предоставляет:

* Изображение книги: Обложка книги для визуальной идентификации.
* Основные данные: Название, автор, жанр, страна, издательство, год выпуска, описание книги.
* Кнопки действий: Кнопка "Читать" открывает страницу для чтения книги, кнопка "Скачать" загружает PDF-файл книги.
* Статус прочтения: Выпадающий список позволяет авторизованному пользователю установить статус прочтения книги.
* Блок с похожими книгами: Отображает 4 случайные книги, схожие по автору, жанру, стране и другим критериям.
* Хедер и футер: Аналогичны главной странице, обеспечивая единый стиль навигации.

Регистрация и авторизация

Процессы регистрации и авторизации реализованы на отдельных страницах:

* Страница регистрации: Содержит форму для создания нового аккаунта, требующую введения необходимых данных.
* Страница авторизации: Форма для входа в существующий аккаунт, позволяющая пользователям получить доступ к персонализированным функциям.

Профиль пользователя

Страница профиля пользователя предоставляет следующие возможности:

* Изображение и данные пользователя: Отображаются имя, фамилия, имя пользователя, электронная почта.
* Кнопка редактирования: Перенаправляет на страницу изменения данных профиля, где можно обновить всю необходимую информацию.
* Списки книг: Включает блоки со списками прочитанных, запланированных и читаемых книг, что помогает пользователям отслеживать свою активность.

Обратная связь

Страница обратной связи включает:

* Форма обратной связи: Пользователи могут отправлять отзывы и предложения, которые сохраняются в базе данных.
* Методы связи: Список дополнительных методов связи с разработчиком для обеспечения взаимодействия и поддержки.

Административная панель

Административная панель предоставляет следующие функции для администраторов:

* Формы ввода данных: Для добавления новых записей в таблицы "Книг", "Авторов", "Жанров", "Стран", "Издательств".
* Таблицы данных: Отображение существующих данных, что облегчает управление и редактирование записей.

Эта структура сайта разработана с целью обеспечения удобного и интуитивно понятного интерфейса как для обычных пользователей, так и для администраторов, что способствует эффективному использованию библиотеки и повышению удовлетворенности пользователей.