|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Разработка клиент-серверных приложений

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Проектирование и разработка клиент-серверного фуллстек-приложения для автоматизированного гардероба

Студент: Сидоров Станислав Дмитриевич

Группа: ИКБО-20-21

Работа представлена к защите \_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/Сидоров С.Д./

(подпись и ф.и.о. студента)

Руководитель: Коваленко Михаил Андреевич, ст. преп.

Работа допущена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/Коваленко М.А./

(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)

М. РТУ МИРЭА. 2024 г.

ГЛОССАРИЙ

В настоящем отчете применяют следующие термины:

|  |  |  |
| --- | --- | --- |
| Серверная часть | – | часть приложения, которая занимается обработкой и хранением данных и передачей результата клиенту. |
| Клиентская часть | – | часть приложения, которая предоставляет пользователю данные в удобном для взаимодействия формате. |
| Архитектура приложения | – | описывает проектирование, набор компоненто системы и объясняет взаимодействие между элементами. |
| База данных | – | структурированное хранилище данных. |
| СУБД | – | программное обеспечение, предназначенное для администрирования баз данных. |
| Контейнерезация | – | упаковка программного кода с библиотеками операционной системы и всеми зависимостями, которые необходимы для выполнения кода. |
| АГР | – | автоматизированный гардеробный ряд, обладающий собственной внутренней нумерацией блоков. |

ВВЕДЕНИЕ

В современном мире каждый сталкивался с большими очередями в гардеробах, с необходимостью хранить полученный жетон и с штрафом за его потерю, что создает множество неудобств посетителям общественных мест. Данная проблема возникает из-за неорганизованности посетителей. Для решения данной проблемы возможен вариант переноса системы работы с жетонами и ячейками гардероба в цифровой формат, что избавит от необходимости толкатся в очереди и беспокоиться за свой жетон. Именно поэтому, темой работы было выбрано фуллстек приложение для автоматизированного гардероба.

Цель данной курсовой работы – разработка комплексного приложения, включающего серверную часть на Java с использованием Spring Framework и принципов REST, базу данных PostgreSQL для хранения данных и клиентскую часть на React, HTML и CSS. Это приложение поможет организациям удобно администрировать имеющиеся у них гардеробы, а также улучшит пользовательский опыт взаимодействия с сотрудниками гардероба, убрав необходимость пользования жетонами, и добавит возможность просматривать свою историю посещений. Для обеспечения высокой отказоустойчивости и масштабируемости приложение будет развернуто на облачном сервере Yandex Cloud с использованием Docker.

Для достижения поставленной цели необходимо решить следующие задачи:

1. провести анализ предметной области разрабатываемого веб-приложения;
2. сформировать функциональные требования к приложению;
3. обосновать выбор средств ведения разработки;
4. разработать архитектуру веб-приложения;
5. реализовать слой серверной логики веб-приложения с использованием выбранных технологий и инструментария;
6. реализовать слой логики базы данных;
7. разработать слой клиентского представления веб-приложения;
8. оформить пояснительную записку по курсовой работе;
9. подготовить презентацию выполненной курсовой работы.

В ходе выполнения работы были использованы следующие методы: сравнение, анализ, классификация, обощение, описание и моделирование.

Работа состоит из введения, оглавления, аннотации, глоссария, семи основных разделов, заключения и списка использованных источников.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

* 1. Описание предметной области

Автоматизированный гардероб – это специальная система содержащая в себе информацию об АГР и ячейках, а также о сотрудниках, обслуживающих ряды, и данных пользователей, использующих ту или иную ячейку. Она автоматически присваивает новому пользователю ячейку, а также помогает выбрать более свободный ряд, что позволяет равномерно распределить нагрузку.

Основная цель автоматизированного гардероба - уменьшить количество взаимодействий пользователя с сотрудником гардероба и избавить от необходимости хранить жетон с номером ячейки, а также добавить организованности в процесс пользования гардеробом и уменьшить время, затраченное на его посещение.

* 1. Анализ ниши веб-приложений автоматизированных гардеробов

На данный момент в сети не представлено большое количество систем, функционал которых соответвует выше описанному. Однако, описанный выше функционал в значительной степени похож на функционал различных приложений для аренды самокатов или автомобилей, так как в данном случае также присутсвует объект аренды, различается только способ выбора и сам объект, которым, в автоматизированном гардеробе, является ячейка. Исходя из этого, для установления необходимого функционала создаваемого веб-приложения были проанализированы 2 приложения с тематикой «кикшеринг» и 1 приложение с тематикой «каршеринг» [1, 2, 3]:

<https://whoosh-bike.ru>,

<https://go.yandex/ru_ru/lp/rides/scooter>,

<https://yandex.ru/drive/>.

На следующих рисунках 1.1 – 1.3 показаны главные страницы анализируемых сайтов.

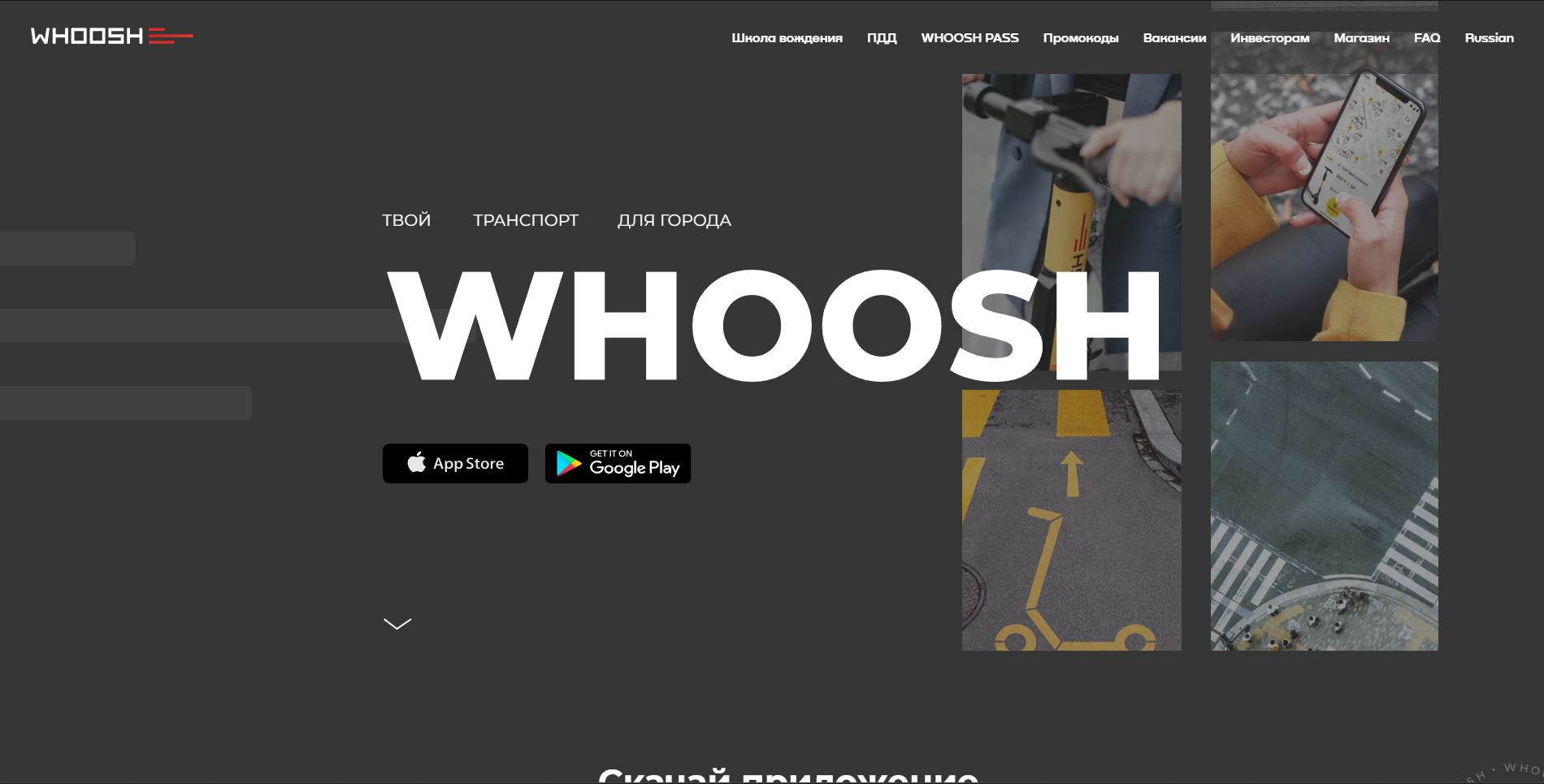


Рисунок 1.1 – Сайт сервиса «Whoosh»

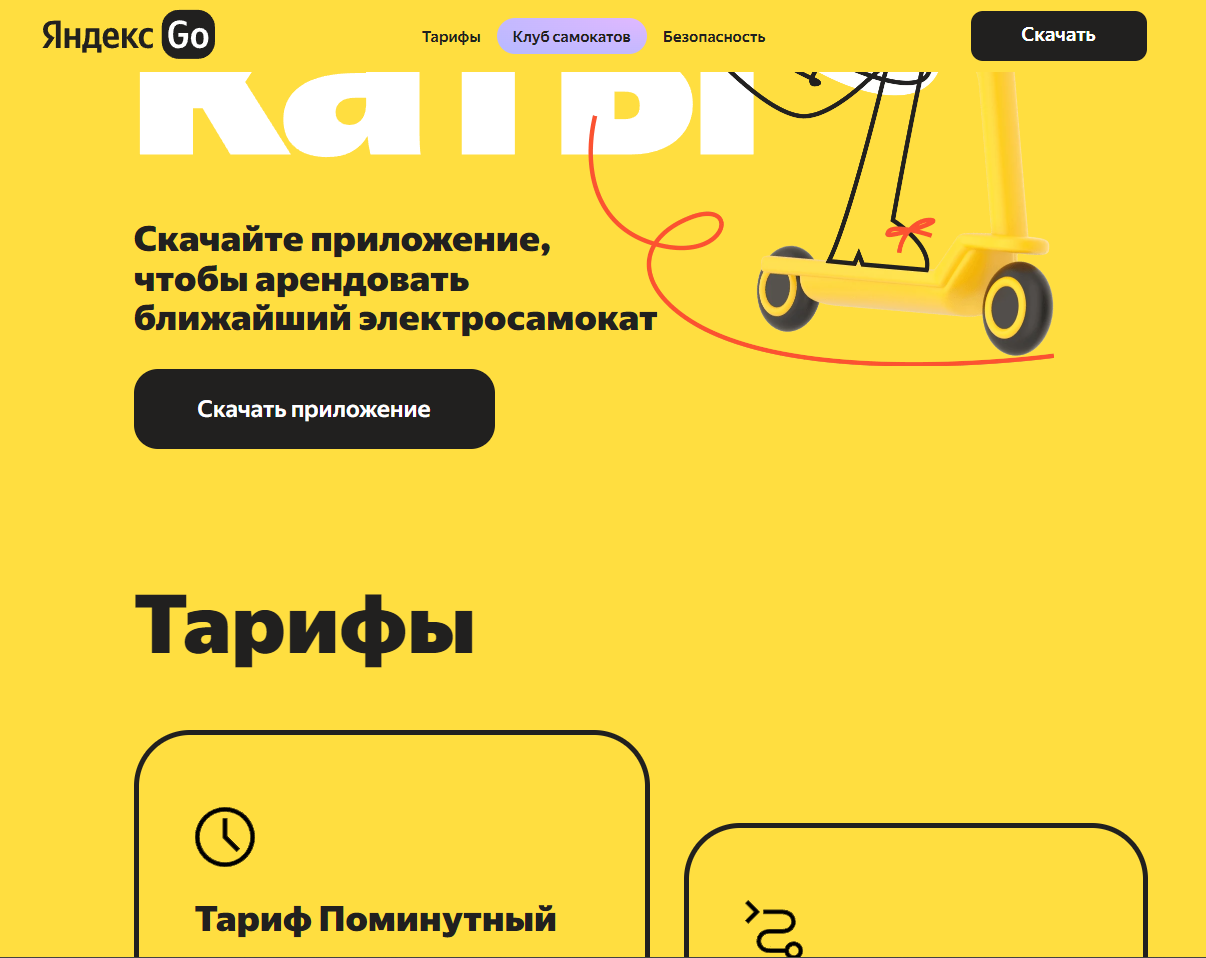


Рисунок 1.2 – Сайт сервиса «Яндекс GO»

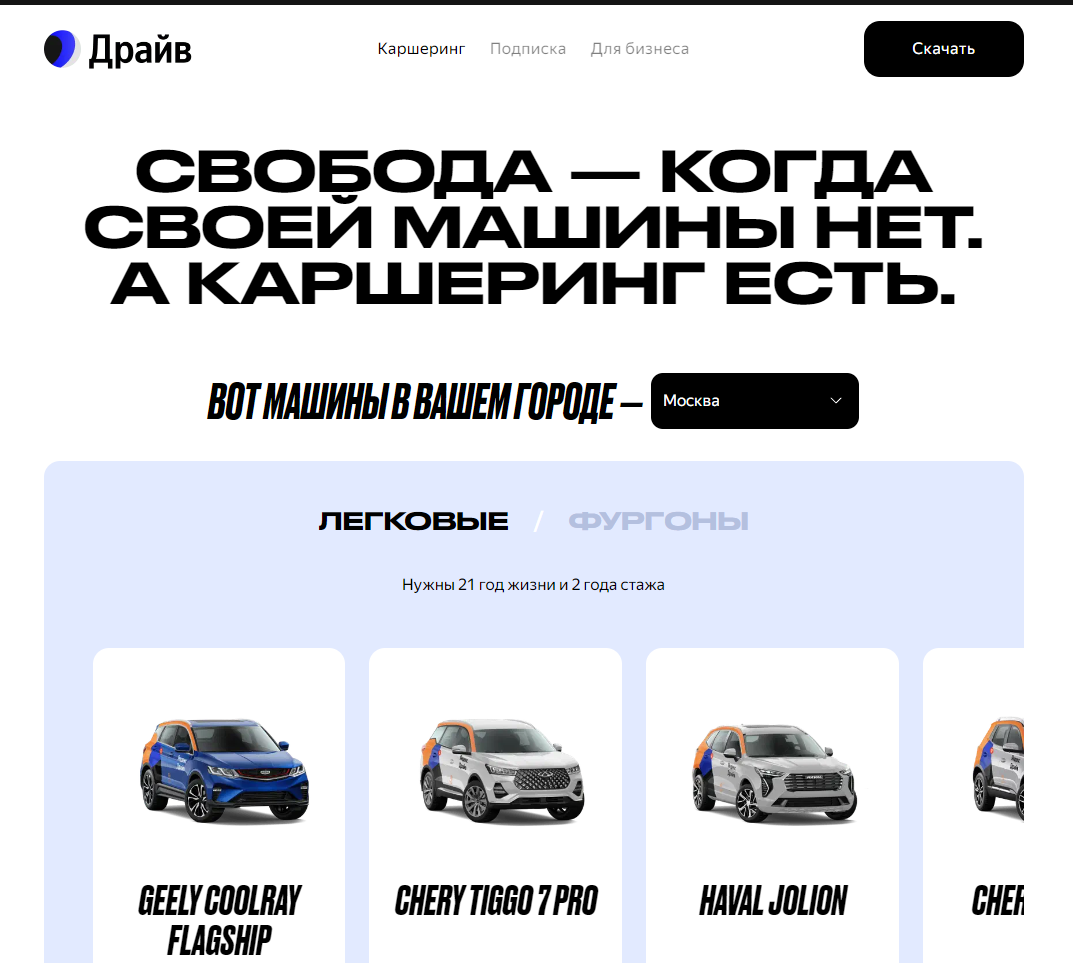


Рисунок 1.3 – Сайт сервиса «Яндекс Драйв»

1.2.1 Авторизация и аутентификация

**Анализ**

Во всех 3-х сервисах необходима аутентификация для того, чтоюы пользователь мог пользоваться функционалом аренды.

**Вывод**

В разрабатываемом программном продукте должна поддерживаться аутентификация. Так доступный функционал будет ограничен в зависимости от роли пользователя.

1.2.2 Аренда

**Анализ**

Сервисы «Whoosh» и «Яндекс Go» позволяют арендовать сразу несколько самокатов, а сервис «Яндекс Драйв» только один автомобиль. Все сервисы позволяют выбрать арендуемый объект.

**Вывод**

С учетом специфики автоматизированного гардероба, создаваемый программный продукт должен иметь возможность арендовать ячейка в, выбранном, пользователем ряду и организации. Однако количество возможных аренд должно быть ограничено до одной, чтобы избежать оставления вещей на длительное хранение.

1.2.3 История пользования

**Анализ**

Все 3 сервиса предоставляют возможность просмотреть информацию о предыдущих использованиях сервиса, содержащую данные об объекте и времени аренды.

**Вывод**

Создаваемое приложение должно поддерживать функционал просмотра данных о посещениях пользователя, содержающую информацию об арендуемой ячейке и времени начала и окончания аренды.

1.2.4 Пользовательский интерфейс

**Анализ**

Все 3 сервиса обладают современным пользовательским интерфейсом.

**Вывод**

Интерфейс должен быть совеременный и достаточно простой для взаимодействия с ним. Он должен включать в себя разделы отображающие основную информацию, а также информацию для навигации по веб-приложению.

1.2.5 Рекомендация объекта аренды

**Анализ**

Два из трех рассматриваемых сервисов рекомендуют пользователю наиболее удобный для пользователя объект аренды.

**Вывод**

Создаваемое приложение должно иметь функционал для рекомендации пользователю наиболее подходящего гардеробного ряда, а также автоматическое назначение арендуемой ячейки.

* 1. Функциональные требования на основе анализа

Веб-приложение должно быть разработано с учетом данных требований:

1. Веб-приложение должно обеспечивать возможность регистрации и аутентификации пользователей;
2. Создаваемый программный продукт должен поддерживать возможность одновременной аренды одной ячейки с возможностью выбора расположения конкретного ряда;
3. Должна быть реализована система рекомендации оптимального ряда;
4. Пользователь должен иметь доступ к совершенным посещениям и информации о них;
5. Создаваемое приложение должно реализовывать полноценный функционал администрирования автоматизированного гардероба;
6. Интерфейс приложения должен быть совеременным и интуитивно понятным. Он должен включать в себя раздел с основной информацией и раздел с навигацией.

2 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ РАЗРАБОТКИ ПРИЛОЖЕНИЯ

На этапе, когда ясны все требования к программному продукту, необходимо определить технологический стек, который будет использоваться

для достижения поставленных целей.

2.1 Основные используемые технологии

Приложение функционирует в клиент-серверной архитектуре, где большая часть обработки данных выполняется на серверной стороне. Для реализации такого подхода были использованы Spring [4], PostgreSQL [5], React [6].

Использование фреймворка Spring обусловлено его популярностью и наличием большого количества различных бибилиотек, позволяющих создавать приложения различного уровня сложности.

Система управления базами данных PostgreSQL выбрана для хранения данных из-за удобства использования, надежности и стабильности. Эта СУБД обладает всеми необходимыми функциями, а также является одной из самых востребованных на текущий момент.

Для разработки клиентской части были выбраны React, CSS и HTML, так как они обеспечивают эффективную разработку масштабируемых веб-приложений, позволяя создавать переиспользуемые компоненты для более удобной поддержки работоспособности пользовательского интерфейса.

2.2 Краткое обоснование использования технологий

2.2.1 Язык програмиирования Java

Данный язык зарекомендовал себя в написании приложений различного уровня сложности, а также обладает обширным списком библиотек позволяющих решать множество задач.

2.2.2 Spring

Данный фреймворк отлично зарекомендовал себя на рынке решений для разработчки веб-приложения на языке Java, а также обладает большим колчиество различных дополнений расширяющих его функционал.

2.2.3 Spring Boot

Данное средство упрощает создание автономных приложений на основе Spring, облегчая настройку и развертывание приложений.

2.2.4 Spring Web

Данный веб-фреймворк позволяет удобно создавать REST-приложения на основе Spring.

2.2.5 Spring Data JPA

Данная технология является частью семейства Spring Data и позволяет удобно взаимодействовать с базами данных через Java Persistence API [7].

2.2.6 Spring Security

Данный фреймворк позволяет гибко настраивать аутентификацию и фактически является стандартом для защиты основанных на Spring приложений.

2.2.7 Maven

Система автоматизированный сборки проекта, используемая для упрощения сборки, тестирования и развертывания приложений.

2.2.8 Bruno

Bruno – это инструмент для тестирования и разработки API веб-сервисов. Он обеспечивает простой и удобный интерфейс, возможность настривать переменные окружения, а также возможность локально сохранять запросы.

2.2.9 JSON Web Tokens

Открытый стандарт для создания токенов безопасности в формате JSON. Данный стандарт часто применяется для аутентификации и авторизации пользователей.

2.2.10 IntelliJ IDEA

Интегрированная среда разработки (IDE) для Java от JetBrains с мощными инструментами для создания, отладки и управления проектами. Она включает интеллектуальные подсказки, автоматическое исправление кода, интеграцию с системами сборки и анализ кода.

2.2.11 Git

Представляет собой распределённую систему контроля версий, отслеживающая изменения в файлах и обычно используемая для координации работы при совместной разработке исходного кода, а также для возможности отменить изменения в случае появления кретичских ошибок в работе программ.

**2.2.12 GitHub**

Веб-платформа, предназначенная для хостинга и совместной разработки программного обеспечения с использованием системы контроля версий Git. Предоставляет широкий инструментарий совместно с приятным и удобным интерфейсом.

2.2.15 PostgreSQL

Система управления реляционными базами данных, часто используемая в веб-разработке и других сферах, обладающая высокой надежностью, стабильностью и большим сообществом разработчиков.

2.2.16 HyperText Markup Language (HTML)

Язык разметки, который применяется для создания структуры и представления контента на веб-страницах и поддерживается всеми современными браузерами.

2.2.17 Cascading Style Sheets (CSS)

Язык таблиц стилей, который используется для оформления веб-страниц, позволяющий изменять стандартное отображение тегов HTML.

2.2.18 JavaScript (JS)

Язык программирования, обеспечивающий возможность динамического изменения содержимого веб-страниц, поддерживаемый современными браузерами и лежащий в основе большого количества фреймворков.

2.2.19 React

Библиотека JavaScript библиотека для создания внешних пользовательских интерфейсов. С помощью компонентного подхода React и встроенных систем оптимизации можно просто создавать нагруженные и масштабирумеые приложения.

2.2.20 Docker

Является монополистом в сфере контейнеризации приложений, необходимой для удобного и быстрого развертывания в различных системах.

2.2.21 Docker-compose

Данное программное обеспечение для автоматизации развертывания и управления приложениями. Оно позволяет с помощью конфигурационного файла собирать и запускать сразу несколько сервисов, что облегчает разработку и масштабирование.