

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОБРАБОТКИ ИНФОРМАЦИИ ПО УЧЕТУ СКЛАДСКИХ ЗАПАСОВ

Вишняк А. С. (студент гр. ИТП-41)

Гомельский государственный университет имени П. О. Сухого, Гомель, Республика
Беларусь

Научный руководитель – **Комракова Евгения Владимировна**

(старший преподаватель кафедры «Информационные технологии» ГГТУ им. П. О. Сухого)

Аннотация: в контексте данной работы рассматривается разработка автоматизированной системы обработки информации по учету складских запасов с использованием современных средств разработки. Ожидаемые результаты исследования включают в себя создание функциональной и оптимизированной системы, способной автоматизировать процессы учета складских запасов, улучшить эффективность управления ресурсами, снизить вероятность ошибок и оптимизировать операционные издержки.

Ключевые слова: веб-приложение, система учета, учет складских запасов, автоматизированная система, эффективность управления ресурсами.

Введение

В современном мире, где эффективное управление ресурсами играет ключевую роль в успехе бизнеса, тема автоматизированной обработки информации по учету складских запасов является актуальной. Сложность управления складскими запасами, включая контроль за поступлением и отгрузкой товаров, оптимизацию запасов, и предотвращение излишков или недостатков, требует инновационных решений. Приложение, способное автоматизировать эти процессы, не только повышает эффективность управления запасами, но и уменьшает вероятность ошибок и снижает операционные издержки.

Результаты и обсуждение

Клиентская часть приложения была разработана с использованием фреймворка React. Одним из ключевых аспектов *React* является его компонентная архитектура, которая позволяет разбивать пользовательский интерфейс на переиспользуемые компоненты, что упрощает разработку, поддержку и масштабирование приложения. Кроме того, *React* предлагает виртуальный *DOM*, что повышает производительность приложения. Расширенная экосистема библиотек и инструментов вместе с вышеперечисленным делает *React* превосходным выбором для создания мощных и инновационных пользовательских интерфейсов [1]. Схема архитектуры стека средств разработки приложения представлена на рисунке 1.

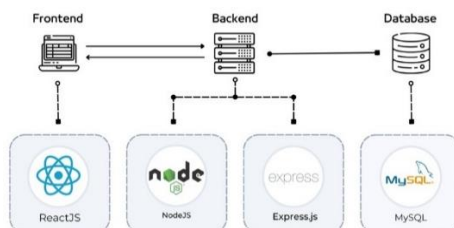


Рисунок 1 – Схема архитектуры стека средств разработки

В данном проекте на стороне сервера основными средствами разработки являлись *Node.js* и фреймворк *Express*. *Node.js* – это среда выполнения *JavaScript*, построенная на движке *V8*, который также используется в браузере *Google Chrome*. *Node.js* обладает асинхронной и событийно-ориентированной архитектурой, что делает его рациональным выбором для создания масштабируемых сетевых приложений.

Express предоставляет гибкий и легковесный веб-фреймворк для разработки серверной части приложения. Он упрощает создание маршрутов, обработку запросов и управление промежуточными слоями. *Express* позволяет эффективно обрабатывать *HTTP*-запросы, управлять сессиями и куками, а также обеспечивает инструменты для разработки надежных и масштабируемых веб-приложений.

Для хранения данных была использована *MySQL* – реляционная система управления

базами данных (СУБД). *MySQL* является одной из наиболее популярных и широко используемых СУБД в мире. Она обладает надежностью, производительностью и широким набором возможностей для управления данными.

В качестве основных средств разработки на стороне сервера используется *Node.js*, для хранения данных – *MySQL* (реляционная СУБД). *Node.js* – асинхронная среда выполнения *JavaScript*, управляемая событиями, предназначенная для создания масштабируемых сетевых приложений. Схематичная архитектура приложения представлена на рисунке 2.

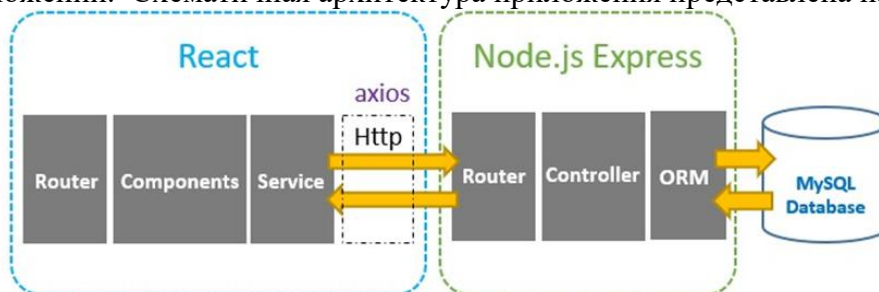


Рисунок 2 – Схематичная архитектура приложения для отправки *HTTP* запросов использовалась *JavaScript* библиотека *Axios*

Заключение

Современный стек средств разработки, включающий *MySQL*, *Express*, *React* и *Node.js*, позволил создать приложение обработки информации по учету складских запасов с рядом важных особенностей. *MySQL* обеспечила высокую производительность, надежность и масштабируемость, а также поддержку транзакций и обеспечение высокого уровня безопасности. *Express* предоставил гибкие инструменты для разработки серверной части приложения. Подход *React* по работе с *DOM* позволил добиться высокой производительности отрисовки пользовательского интерфейса. Помимо перечисленных особенностей *Node.js*, его использование вместе с *React* обеспечило единую кодовую базу на языке *JavaScript*.

Литература

2. Бэнкс Алекс, Порселло Ева. *React и Redux: функциональная веб-разработка* – СПб.: Питер, 2018. – 336 с.: ил. – (Серия «Бестселлеры O'Reilly»).

АНАЛИЗ ДИНАМИКИ ЖЕЛЕЗНОДОРОЖНОГО СОСТАВА В MSC ADAMS

Ворожун А.В. (магистрант гр.МТ-46)

Белорусский государственный университет транспорта, Гомель, Республика Беларусь

Научный руководитель – **Гегедеш Марина Григорьевна**

(к.т.н., доцент кафедры «Управление грузовой и коммерческой работой» БелГУТ)

Аннотация: В статье проведены исследования торможения однородного пассажирского поезда на переломном профиле пути с разностью уклонов до 6 ‰, на основе компьютерного моделирования в инженерном пакете MSC.ADAMS. Выполнен сравнительный анализ с существующими методиками расчета тормозного пути.

Ключевые слова: компьютерное моделирование, железнодорожный транспорт, тормозной путь, переменный уклон.

Введение

Одним из основных факторов, обеспечивающих безопасность движения на железнодорожном транспорте, является безотказная и эффективная работа тормозных систем. Показателем их качества является длина тормозного пути. Цель представленной работы – сравнение результатов расчетов, полученных с помощью различных методов, в том числе, на основе компьютерного моделирования.

Результаты и обсуждения

Разработана компьютерная модель поезда в среде программы MSC.ADAMS (рисунок 1). При моделировании приняты следующие допущения: локомотив и вагоны считаются