

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**ИСПОЛЬЗОВАНИЕ MEAN СТЕКА ДЛЯ РАЗРАБОТКИ E-COMMERCE
СЕРВИСОВ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Умбетова Асмана Амыржановича

Научный руководитель:

доцент

_____ Е.В. Кудрина

подпись, дата

Зав. кафедрой:

к. ф.-м. н., доцент

_____ М.В. Огнева

подпись, дата

Саратов 2020

ВВЕДЕНИЕ

Актуальность темы. На сегодняшний день электронная коммерция является богатой сферой экономической деятельности для многих людей. Принимая во внимание современные условия рынка и устоявшиеся стандарты бизнеса в интернете, к разработке программных продуктов в сфере электронной коммерции предъявляется большое количество требований. Для удовлетворения этих требований было разработано большое количество программных платформ и инструментов, которые могли бы помочь в достижении целей, продиктованных современными условиями электронной коммерции.

Основными проблемами разработок в сфере электронной коммерции можно выделить:

- необходимость в большой скорости разработки
- необходимость в масштабируемости финального продукта
- необходимость в стабильности и быстродействии финального продукта
- необходимость в гибкости при разработке.

Современные технологии дают возможность решить все эти проблемы за счет использования гибких методологий разработки и продвинутого инструментария для разработки программного продукта.

Одним из актуальных стеков разработки программного продукта в сфере электронной коммерции является MEAN стек. Данный стек является особенным из-за своей необычной архитектуры, которая позволяет данному стеку удовлетворять наибольшему числу бизнес задач сферы электронной коммерции.

Цель бакалаврской работы – научиться использовать технологии MEAN-стека для разработки e-commerce-сервисов.

Поставленная цель определила **следующие задачи**:

1. Проанализировать сферу электронной коммерции и ее особенности.
2. Представить обзор современных инструментов для разработки программных продуктов в сфере электронной коммерции

3. Изучить технологий MEAN стека и возможности их применения для разработок в сфере электронной коммерции
4. Разработать и реализовать e-commerce сервис «Интернет-магазин» с использованием MEAN стека.

Методологические основы «Использование MEAN стека для разработки e-commerce сервисов» представлены в работах А. Дорофеева, А. Майера, Х. Штормера, Т. Сингха, С. Малика, Д. Саркара, Д. Полтера, С. Джонсона, С. Кокса.

Практическая значимость бакалаврской работы. Практической значимостью работы является веб-приложение «Интернет-магазин». Данное приложение является шаблонным решением для разработки сервисов в сфере электронной коммерции. Приложение построено использованием MEAN стека и обладает возможностями для расширения и масштабирования. Приложение не является специфичным для конкретной области электронной коммерции.

Структура и объём работы. Бакалаврская работа состоит из введения, теоретического раздела, практического раздела, заключения, списка использованных источников и приложения. Общий объём работы – 54 страниц, из них 41 страница – основное содержание, включая 7 рисунков, список использованных источников информации – 11 наименований. Выпускная работа содержит два приложения: Приложение А содержит программную реализацию основного функционала приложения «Интернет-магазин», Приложение Б – диск с исходным кодом приложения «Интернет-магазин».

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические основы использования MEAN стека для разработки e-commerce» посвящен обзору сферы интернет коммерции. В данном разделе приведена основная информация о сфере интернет коммерции, основные определения и понятия. Также приведена историческая справка с разбором развития сферы электронной коммерции.

Электронный бизнес – это любая деловая активность, использующая возможности глобальных информационных сетей для преобразования внутренних и внешних связей с целью создания прибыли.

Электронная коммерция является важнейшим составным элементом электронного бизнеса. Под электронной коммерцией (e-commerce) подразумеваются любые формы деловых сделок, при которых взаимодействие сторон осуществляется электронным способом вместо физического обмена или непосредственного физического контакта, и в результате которого право собственности или право пользования товаром или услугой передается от одного лица другому.

Разработка сервисов электронной коммерции, таких как интернет-магазины или приложения для интернет-торговли, сильно завязана на требованиях, предъявляемых владельцами торговых площадок к конечному продукту. Также на процесс разработки влияют предпочтения потенциальных клиентов сервиса, поэтому подбор нужного инструментария для разработке является критически важным этапом при проектировании интернет платформы для предоставления услуг электронной коммерции.

Основными требованиями, предъявляемыми к сервисам электронной коммерции можно выделить:

- Скорость загрузки веб-страницы приложения
- Надежность работы приложения
- Скорость пользовательских интеракций внутри приложения
- Внешний вид приложения
- Скорость разработки приложения

- Расширяемость и гибкость в настройке приложения.

Инструментарий для разработки интернет приложения делится на две основные части: бекенд стек и фронтенд стек разработки.

Фронтенд стек разработки включает в себя инструментарий для разработки части приложения, отвечающей за взаимодействие с пользователем, Основными технологиями данного стека являются:

- HTML (HyperText Markup Language)
- CSS (Cascading Style Sheets)
- JS (JavaScript)

Разберем данные технологии подробнее.

HTML и CSS – очень важная часть разработки любого интернет приложения. Данные технологии отвечают за внешний вид приложения. HTML (от англ. HyperText Markup Language – «язык гипертекстовой разметки») – стандартизированный язык разметки документов в интернете. С помощью HTML задается семантическая структура внешнего вида интернет приложения.

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки HTML. CSS предназначена для построения визуального представления программного продукта.

JavaScript (JS) – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Говоря о бекенд части веб-приложения, основными компонентами можно выделить:

- Язык программирования
- Бекенд фреймворк

- База данных
- Веб-сервер

В качестве языка программирования для бекенд части разработчики используют большое количество языков, например Java, C#, PHP, Ruby, Python, JS и т.д. и выбор конкретного языка всегда зависит от требований к проекту и целей

MEAN (аббревиатура от MongoDB, Express.js, Angular, Node.js) — набор программного обеспечения, который используется для веб-разработки.

Основные технологии MEAN-стека:

- база данных MongoDB,
- фреймворк для создания веб-приложений Express.js
- платформа Node.js,
- фронтенд фреймворк Angular (либо React или Vue.js).

Основными преимуществами MEAN стека является тот факт, что все компоненты набора поддерживают программирование на JavaScript, и серверная и клиентская часть MEAN-приложений может быть написана на этом языке программирования. Это позволяет ускорить процесс разработки приложений с помощью данного стека.

Второй раздел «Разработка и реализация e-commerce сервиса «Интернет-магазин» на основе MEAN стека» посвящен реализации e-commerce сервиса «интернет-магазин» на основе MEAN стека.

«Интернет-магазин» является шаблонным решением для разработки сервисов в сфере электронной коммерции на основе MEAN стека.

Архитектура приложения состоит из четырёх основных компонентов:

1. База данных MongoDB,
2. Фреймворк Express.js,
3. Платформа Node.js,
4. Библиотека React.

Схема архитектуры приложения показана на Рисунке 1.

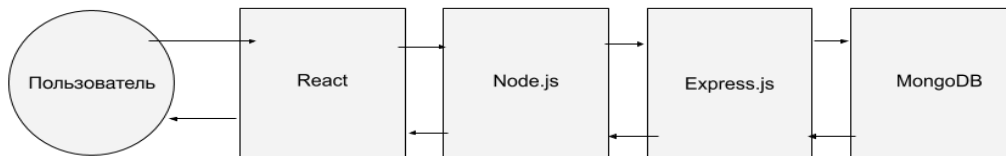


Рисунок 1 – Схема архитектуры приложения «Интернет-магазин»

Каждый компонент системы коммуницирует с соседним передавая информацию от пользователя в базу данных и обратно.

Серверная часть приложения «Интернет-магазин» состоит из базы данных MongoDB, библиотеки mongoose, фреймворка Express.js и платформы Node.js.

Рабочий процесс серверной части состоит из следующих этапов:

1. Node.js регистрирует запрос на указанный порт сервера и передает информацию о запросе в Express.js.
2. Express.js ищет среди описанных в эндпоинтов подходящий и в случае успеха передает управление обработчику эндпоинта.
3. Обработчик эндпоинта выполняет необходимые вычисления или подгружает данные с помощью mongoose из базы данных и отправляет обратно в качестве ответа на запрос.
4. Express.js передает ответ в обработчик Node.js
5. Node.js формирует HTTP ответ, который отсылается в ответ на запрос.

Файловая структура серверной части приложения отображена на Рисунке 2.

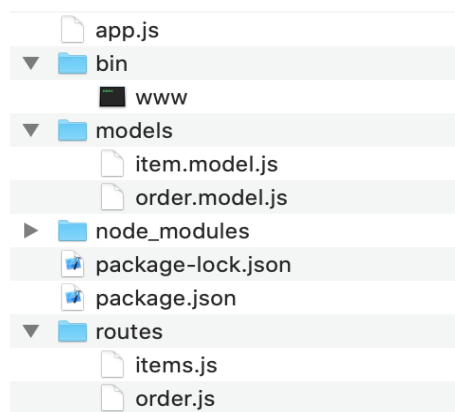


Рисунок 2 – Файловая структура серверной части приложения.

Клиентская часть приложения «Интернет-магазин» построена на платформе Node.js так же, как и серверная часть. Клиентская часть приложения состоит из 4 основных компонентов:

- Платформа Node.js,
- Фреймворк Express.js,
- Библиотека React,
- Библиотека Redux,
- Сборщик Webpack.

Общий принцип работы данной части приложения состоит из следующих этапов:

- Библиотеки React и Redux в связке формируют кодовую базу, которая будет выполняться на стороне клиента в браузере. Данная часть приложения отвечает за внешний вид приложения и взаимодействие с пользователем.
- Сборщик Webpack предназначен для формирования «бандла» (от англ «bundle» – пучок, пакет), это весь пользовательский код, который будет выполняться на стороне клиента, собранный в минимизированные сжатые файлы. Сборщик нужен для оптимизации и минимизации клиентской кодовой базы.
- Node.js в связке с Express.js работают для предоставления пользовательской кодовой базы клиенту по запросу на указанный порт.

Во второй части раздела приведена демонстрация работы финального e-commerce сервиса «Интернет-магазин». Приведена информация работе серверной части приложения, приведены скриншоты запросов к серверной части приложения, продемонстрирована работа клиентского приложения, описана, приведены скриншоты работы клиентского приложения.

Файловая структура клиентской части приложения отображена на Рисунке 3.

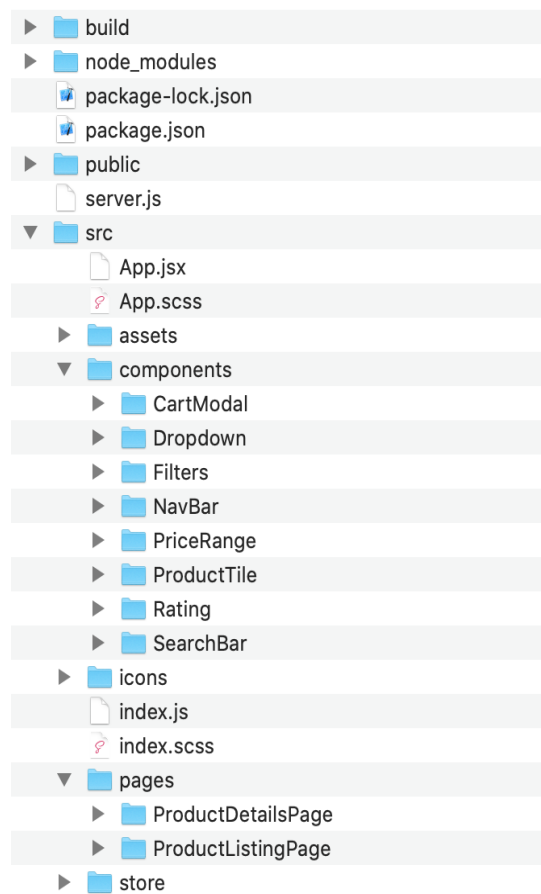
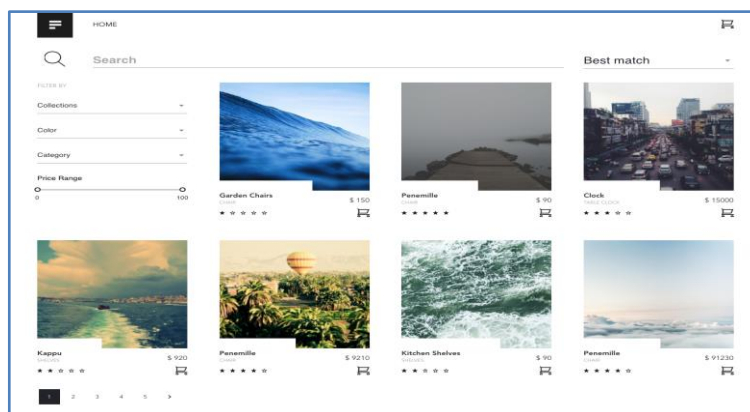
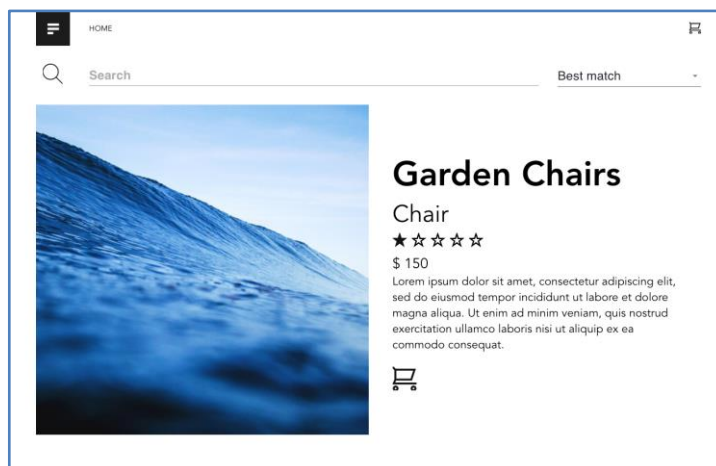


Рисунок 3 – Файловая система клиентской части приложения.

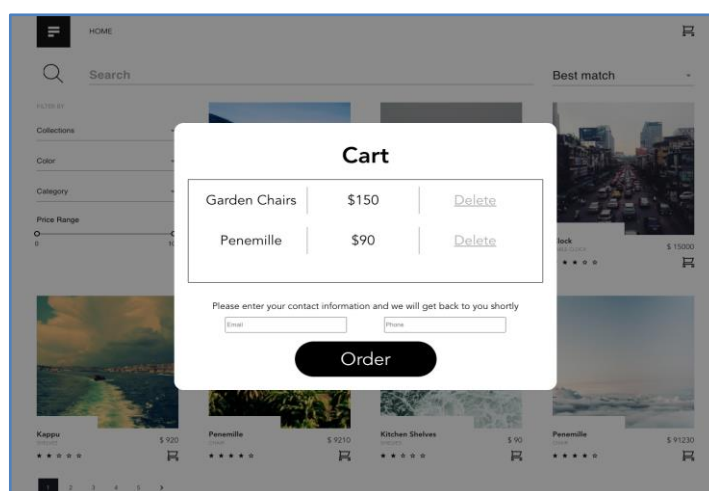
Клиентское приложение для приложения «Интернет-магазин» является одностраничным веб-приложением, запускаемым в браузере. На Рисунке 4 приведены скриншоты приложения «Интернет-магазин».



а) страница со списком товаров



б) страница конкретного товара



в) корзина покупателя

Рисунок 4 – Скриншоты клиентской части приложения «Интернет-магазин».

Анализ качества финального результата будет проводиться по следующим критериям:

- скорость загрузки клиентского приложения
- скорость ответа серверного приложения
- браузерная поддержка

Скорость загрузки клиентского приложения составляет в среднем 3.2 секунд. Данный результат может меняться в зависимости от мощности системы и скорости интернет соединения. Средняя скорость загрузки страницы в 2019 году составила 22 секунды. Результат загрузки в 3.2 является положительным результатом относительно среднего результата в индустрии.

Средняя скорость ответа серверного приложения 47.3 мс на POST запрос и 93 мс на GET запрос. Данные результаты являются намного превосходящими по скорости среднее значение в индустрии. По статистике на 2019 год, среднее значение времени отклика сервера составляло 400-700 мс. Данный результат может измениться в зависимости от количества данных и загруженности сервиса MongoDB Atlas.

Проверка браузерной поддержки сайта осуществлена на основе 3-х наиболее используемых браузеров: Google Chrome, Mozilla Firefox и Apple Safari.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были решены все поставленные задачи, что позволило достичь запланированной цели – научиться использовать MEAN стек для разработки приложений в сфере интернет коммерции.

Результатом данной работы стала разработка и реализация шаблонного решения «Интернет-магазин» для использования в сфере электронной коммерции, построенное на базе MEAN стека. Данное приложение не является специфичным для конкретной области электронной коммерции, является масштабируемым и расширяемым за счёт возможностей платформ MEAN стека.

Исследования, проведенные в ходе эксплуатации данного приложения, показали, что: скорость загрузки клиентского приложения составляет в среднем 3.2 секунд (средний результат в индустрии на 2019 год – 22 сек); средняя скорость ответа серверного приложения 47.3 мс на POST запрос и 93 мс на GET запрос (средний результат в индустрии на 2019 год – 400-700 мс). Данные результаты могут измениться в зависимости от количества данных и загруженности сервиса MongoDB Atlas, но в целом являются позитивными.

ОСНОВНЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ

1. Andreas Meier, Henrik Stormer. eBusiness & eCommerce: Managing the Digital Value Chain – Springer Science & Business Media, Apr 3, 2009 – 222 с.: ил. ISBN 978-3-540-89327-1
2. Алексей Дорофеев: Электронный бизнес. Учебное пособие.. — Кнорус, 2019 г. — 144 с.: ил. ISBN 978-5-406-07172-4
3. Kenneth C. Laudon, Carol G. Traver : E-Commerce: Business, Technology, Society, Second Edition 2nd Edition — Addison Wesley; 2 edition (August 15, 2003) — 144 с.: ил. ISBN-13: 978-0321200563
4. Singh, T., Malik, S., & Sarkar, D. (2016). E-commerce website quality assessment based on usability. 2016 International Conference on Computing, Communication and Automation (ICCCA). doi:10.1109/ccaa.2016.7813698
5. Cao, M., Zhang, Q. and Seydel, J. (2005), "B2C e-commerce web site quality: an empirical examination", Industrial Management & Data Systems, Vol. 105 No. 5, pp. 645-661. doi:10.1108/02635570510600000
6. Poulter, A. J., Johnston, S. J., & Cox, S. J. (2015). Using the MEAN stack to implement a RESTful service for an Internet of Things application. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). doi:10.1109/wf-iot.2015.7389066