Титульный лист

МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

Государственное бюджетное профессиональное образовательное учреждение Московской области «Ногинский колледж»

Курсовой проект

по МДК.09.01 Проектирование и разработка веб-приложений

ПМ.09 Проектирование, разработка и оптимизация веб-приложений

Тема:

ИСПОЛЬЗОВАНИЕ ФРЕЙМВОРКА NodeJS В РАЗРАБОТКЕ ВЕБ-ПРИЛОЖЕНИЯ «Служба доставки еды «ДодоПтица»

Разработчик: студент группы 3ИСПР

Иванов И.И.

(подпись)

Оценка защиты курсового проекта

« »

Дата защиты

« » 2024

Руководитель проекта:

преподаватель Степанов С.О.

(подпись)

Ногинск, 2024 г

## СОДЕРЖАНИЕ

[ВВЕДЕНИЕ](#%D0%B2%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0)

[ОСНОВНАЯ ЧАСТЬ](#%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D0)  
2.1. [Общие сведения о разработке веб-приложения на фреймворке Node.js](#%D0%BE%D0%B1%D1%89%D0%B8%D0%B5-%D1%81%D)  
2.2. [Инсталляция необходимых программ и предпроектная подготовка](#%D0%B8%D0%BD%D1%81%D1%82%D0%B0%D0%BB%D0)  
2.3. [Руководство программиста](#%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0)  
2.4. [Установка веб-приложения на общедоступный хостинг](#%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0)

[ЗАКЛЮЧЕНИЕ](#%D0%B7%D0%B0%D0%BA%D0%BB%D1%8E%D1%87%D0)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0)

[ПРИЛОЖЕНИЯ](#%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0)

## ВВЕДЕНИЕ

### Задание на курсовой проект

Цель данного курсового проекта — разработать веб-приложение «Служба доставки еды» с использованием фреймворка Node.js и сопутствующих технологий. В проекте будет реализована возможность управления заказами, обработки платежей и интеграции с базой данных.

### Актуальность темы

С развитием электронной коммерции и увеличением потребности в онлайн-сервисах доставка еды становится все более востребованной. Современные веб-приложения для доставки еды позволяют пользователям удобно заказывать еду из ресторанов и отслеживать статус заказов. В связи с этим разработка эффективных и масштабируемых решений на основе Node.js и других современных технологий является актуальной задачей.

### Обзор текущего состояния

Сегодня на рынке присутствуют многочисленные платформы для доставки еды, такие как Uber Eats, DoorDash и Delivery Club. Эти системы предоставляют широкий функционал, включающий управление меню, обработку заказов, интеграцию с платежными системами и отслеживание доставки. Использование таких инструментов, как Node.js, React.js, MongoDB и Stripe, позволяет создать конкурентоспособное веб-приложение, способное эффективно обрабатывать большой объем данных и обеспечивать высокую производительность.

## ОСНОВНАЯ ЧАСТЬ

Для реализации веб-приложения «ДодоПтица» был выбран следующий стек технологий:

* **Node.js**: Серверная платформа, основанная на движке JavaScript V8, обеспечивающая высокую производительность и масштабируемость. Node.js был выбран благодаря его асинхронной природе и возможности использовать JavaScript как на стороне сервера, так и на стороне клиента.
* **Express.js**: Легкий и гибкий фреймворк для Node.js, используемый для построения серверной части приложения. Express.js обеспечивает простой и интуитивно понятный интерфейс для создания маршрутов и обработки HTTP-запросов.
* **React.js**: Библиотека JavaScript для построения пользовательских интерфейсов. React.js позволяет создавать динамичные и отзывчивые интерфейсы, которые обеспечивают удобство и простоту использования для конечных пользователей.
* **MongoDB**: Документо-ориентированная база данных, идеально подходящая для хранения данных веб-приложений благодаря своей гибкости и масштабируемости. MongoDB обеспечивает эффективное хранение и доступ к данным, что особенно важно для приложений с большим количеством запросов.
* **Stripe**: Платежный сервис, интегрированный в приложение для обеспечения безопасных и удобных платежей. Stripe предоставляет надежные API для обработки транзакций, что делает его отличным выбором для веб-приложений электронной коммерции.

### Основные функции и требования

Разработка веб-приложения «ДодоПтица» включала в себя реализацию ряда функциональных требований, обеспечивающих полный цикл обслуживания пользователей и управления контентом. Основные требования к сайту включают:

1. **Авторизация и регистрация**: Пользователи должны иметь возможность создавать учетные записи и входить в систему. Для этого была реализована система авторизации и регистрации с использованием JSON Web Tokens (JWT) для обеспечения безопасного хранения и передачи данных аутентификации.
2. **Ежедневные и еженедельные акции**: В системе предусмотрено управление акциями, которые могут обновляться ежедневно и еженедельно. Эти акции отображаются на главной странице и привлекают внимание пользователей, предлагая специальные предложения и скидки.
3. **Каталог товаров**: Каталог включает не менее 40 товаров, сгруппированных по категориям. Пользователи могут просматривать товары, фильтровать их по категориям.
4. **Корзина товаров**: Пользователи могут добавлять товары в корзину, изменять их количество или удалять ненужные позиции. Корзина сохраняется на протяжении сеанса пользователя и готова к оформлению заказа в любой момент.
5. **Промокоды на скидку**: Возможность применения промокодов для получения скидок на заказ. Промокоды могут быть введены на этапе оформления заказа, и система автоматически пересчитывает итоговую сумму с учетом скидки.
6. **Выбор способа доставки**: Пользователи могут выбрать между доставкой по указанному адресу или самовывозом из пункта выдачи. Эта опция позволяет гибко адаптировать услуги под предпочтения клиентов.
7. **Уведомления по электронной почте**: После оформления заказа пользователю и администратору отправляется электронное письмо с подробной информацией о заказе. Это обеспечивает удобство для клиента и помогает администратору своевременно обрабатывать заказы.
8. **Отслеживание статуса заказа**: Пользователи могут отслеживать текущий статус своего заказа через интерфейс приложения. Заказы проходят через несколько этапов: оформлен, принят, доставляется и завершен.
9. **Панель администратора**: Административная панель предоставляет доступ к функциям управления контентом и заказами. Администраторы могут добавлять и удалять акции, промокоды, товары, а также просматривать и изменять статус заказов. Также предусмотрена возможность создания учетных записей для курьеров.
10. **Функционал для курьеров**: Курьеры имеют доступ к своим заказам через интерфейс приложения. Они могут авторизоваться, просматривать текущие заказы и обновлять их статус до «доставлено» после выполнения доставки.

### Архитектура приложения

Веб-приложение «ДодоПтица» разработано на основе архитектуры клиент-сервер.

* **Клиентская часть** реализована с использованием React.js и взаимодействует с сервером через RESTful API. Компоненты React обеспечивают динамическое обновление пользовательского интерфейса без необходимости перезагрузки страницы.
* **Серверная часть** построена на базе Node.js с использованием фреймворка Express.js. Сервер обрабатывает запросы от клиента, взаимодействует с базой данных MongoDB и управляет платежами через интеграцию с Stripe.

#### 3.2.1 Инсталляция программного обеспечения

1. Установка Node.js

Скачайте установщик Node.js с официального сайта https://nodejs.org/en/download/package-manager для вашей операционной системы (Windows, macOS, Linux) и следуйте инструкциям мастера установки.

Проверьте установку, открыв командную строку (Windows) или терминал (macOS/Linux) и введя команды node -v и npm -v. Должны отобразиться версии Node.js и npm соответственно.

2. Установка Nodemon

Откройте командную строку/терминал, перейдите в каталог вашего проекта и выполните команду npm install nodemon --save-dev.

Проверьте установку командой nodemon -v. Должна отобразиться версия Nodemon.

3. Создание репозитория на GitHub

3.1 Зарегистрируйтесь на GitHub (если еще нет) на https://github.com/ и создайте новый репозиторий, например, my-node-js-app.

4. Создание локального репозитория и подключение к удаленному

4.1 Инициализация локального репозитория Git:

В командной строке/терминале перейдите в каталог проекта и выполните команду git init.

4.2 Подключение к удаленному репозиторию:

Скопируйте URL-адрес вашего репозитория GitHub и выполните команду git remote add origin <URL-адрес-репозитория>.

5. Создание и отправка первого коммита

5.1 Добавление файлов в отслеживание Git:

Введите команду git add . для добавления всех файлов проекта в отслеживание.

5.2 Создание коммита:

Введите команду git commit -m "Первый коммит".

5.3 Отправка коммита в удаленный репозиторий:

Введите команду git push -u origin master.

6. Использование GitHub Desktop (необязательно)

6.1 Скачайте и установите GitHub Desktop с сайта https://desktop.github.com/ для вашей операционной системы.

6.2 Подключитесь к своей учетной записи GitHub в приложении.

6.3 Клонируйте репозиторий:

Нажмите кнопку "Клонировать репозиторий", введите URL-адрес вашего репозитория, выберите папку для клонирования и нажмите "Клонировать".

6.4 Работайте с локальным репозиторием:

## В GitHub Desktop вы можете отслеживать изменения, создавать коммиты и отправлять их в удаленный репозиторий, а также использовать дополнительные функции. В GitHub Desktop вы можете отслеживать изменения, создавать коммиты и отправлять их в удаленный репозиторий, а также использовать дополнительные функции.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Node.js. Официальный сайт: [nodejs.org](https://nodejs.org)

React.js. Официальный сайт: [reactjs.org](https://reactjs.org)

MongoDB. Официальный сайт: [mongodb.com](https://www.mongodb.com)

Express.js. Официальный сайт: [expressjs.com](https://expressjs.com)

Stripe. Официальный сайт: [stripe.com](https://stripe.com)

Видеоруководство по созданию приложения: [YouTube](https://www.youtube.com/watch?v=DBMPXJJfQEA)

Репозиторий проекта на GitHub: [GitHub](https://github.com/Takezo123/kurs)

## ПРИЛОЖЕНИЯ

### Приложение 1: Структура проекта

### Приложение 2: Скриншоты системы контроля версий

### Приложение 3: Пример кода

Приложение 4: Итоговый вид веб-приложения