|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 клиентских частей интернет-ресурсов

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Клиентская часть интернет-ресурса «Пиццерия»

Студент: Быченков А.К.

Группа: ИКБО-32-21

Работа представлена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись и ф.и.о. студента)

Руководитель: Матчин Василий Тимофеевич

Работа допущена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(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. 2022 г.

Содержание

[ВВЕДЕНИЕ 3](#_Toc119351435)

[1. Общие сведения 3](#_Toc119351436)

[1.1 Наименование клиентской части интернет-ресурса 3](#_Toc119351437)

[1.2. Прикладное программное обеспечение, необходимое для разработки и функционирования клиентской части интернет-ресурса 3](#_Toc119351438)

[1.3. Языки и технологии, с помощью которых реализована клиентская часть интернет-ресурса 4](#_Toc119351439)

[2. Описание логической структуры 5](#_Toc119351440)

[2.1. Анализ предметной области разрабатываемой клиентской части интернет-ресурса 5](#_Toc119351441)

# ВВЕДЕНИЕ

Веб-разработка в последнее время активно развивается. Современные сайты постоянно усложняются, появляются новые технологии для разработки. В настоящее время трудно представить компанию, которая не использует веб-технологии в своей деятельности. Веб-сайты нужны всем.

В рамках данной курсовой работы будет разработана клиентская часть интернет-ресурса для крупной сетевой пиццерии. Необходимо разработать сайт, который позволит пиццерии работать в современном мире веб-технологий: реализовать каталог продаваемых товаров, регистрацию клиентов, оформление заказов на доставку.

# 1. Общие сведения

## 1.1 Наименование клиентской части интернет-ресурса

Клиентская часть интернет-ресурса «Пиццерия».

## 1.2 Функциональное назначение как совокупность свойств клиентской части интернет-ресурса, определяемых конкретными особенностями набора функций, способных удовлетворять заданным или подразумеваемым потребностям

Разрабатываемый интернет-ресурс может использоваться для работы ресторанов, ориентированных на доставку, в том числе пиццерий. Необходимо реализовать возможность просмотра каталога товаров, авторизацию пользователей, оформление заказов на доставку с онлайн-оплатой.

# 1.3 Прикладное программное обеспечение, необходимое для разработки и функционирования клиентской части интернет-ресурса

Для разработки используется редактор кода Visual Studio Code. Для функционирования клиентской части интернет-ресурса используется интерфейс командной строки.

# 2. Описание логической структуры

## 2.1. Анализ предметной области разрабатываемой клиентской части интернет-ресурса

Рассмотрим аналоги разрабатываемого интернет-ресурса. К примеру, сайт пиццерии «Додо Пицца», рис. 1.

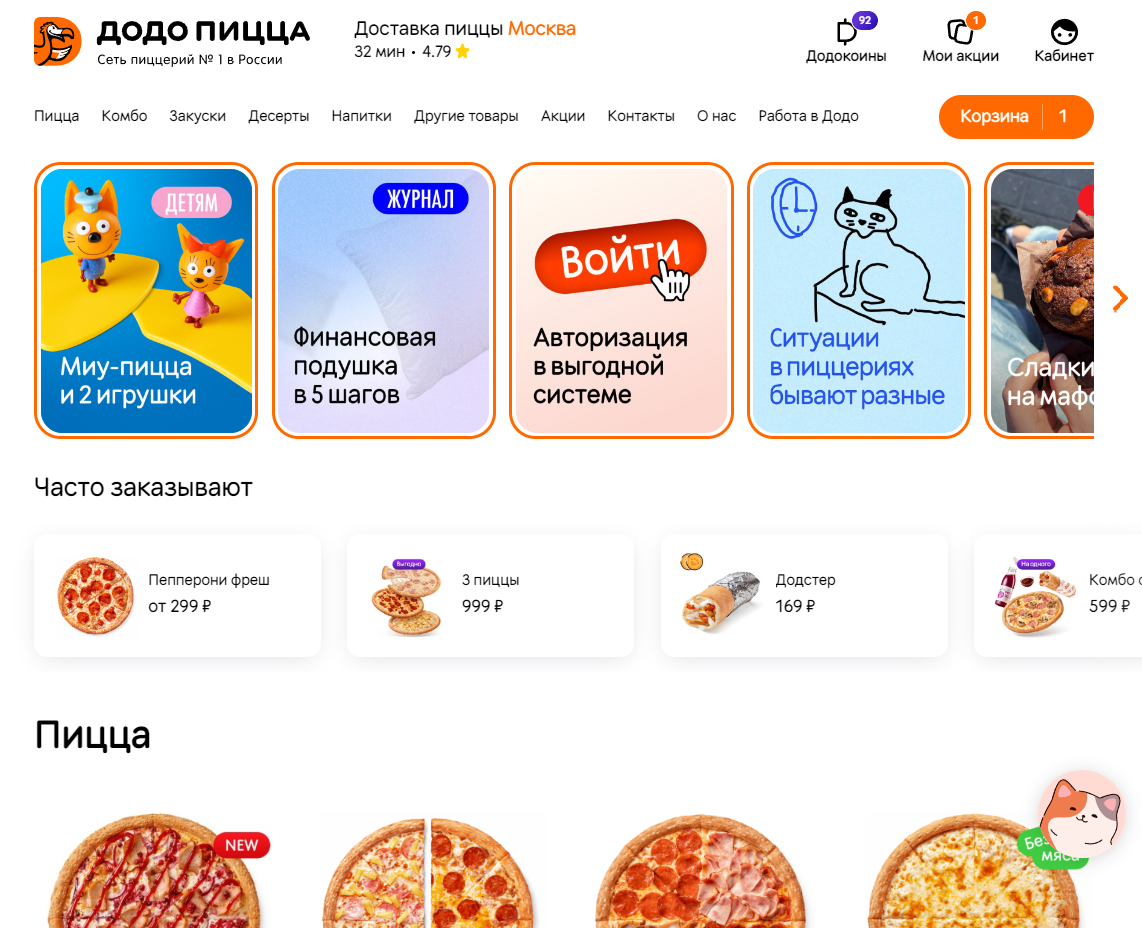


Рисунок 1. Главная страница интернет-ресурса «Додо Пицца».

На главная странице расположен каталог товаров, категории, акции, личный кабинет, кнопка корзины. Для заказа товаров необходимо авторизоваться по номеру телефона. При нажатии на товар можно выбрать размер, добавить ингредиенты по вкусу и добавить в корзину, рис 2.

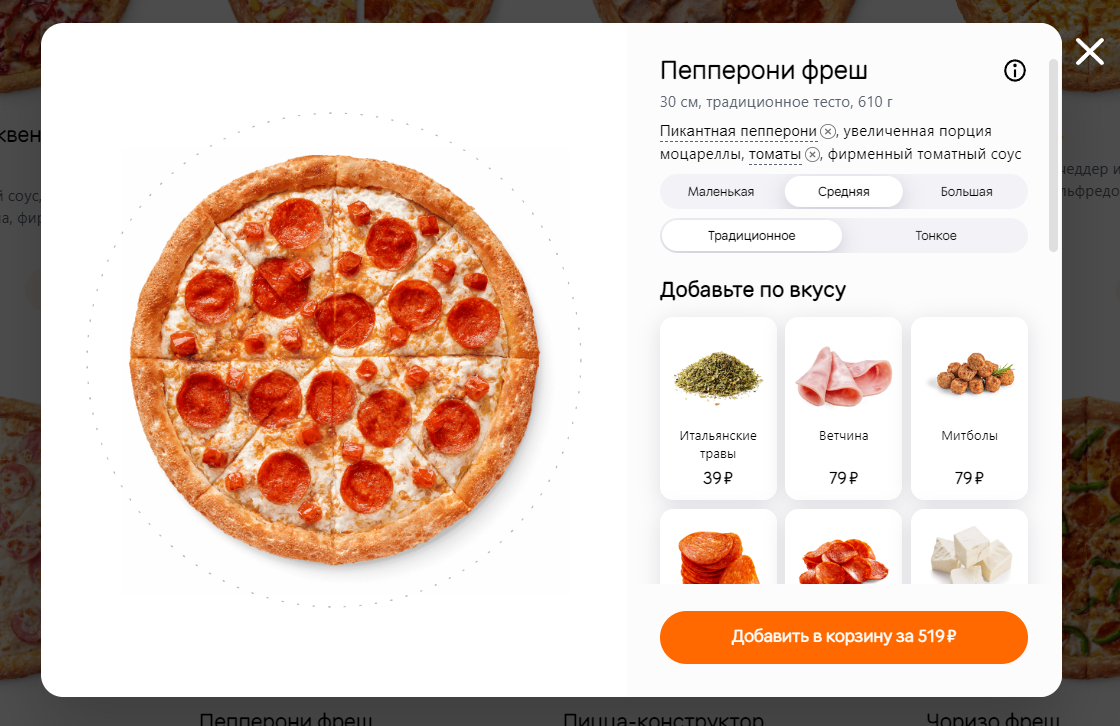


Рисунок 2. Выбор размера пиццы и добавление дополнительных ингредиентов.

В корзине можно увидеть добавленные товары, итоговую сумму, изменить количество товаров, и ввести промокод для получения скидки, рис. 3.

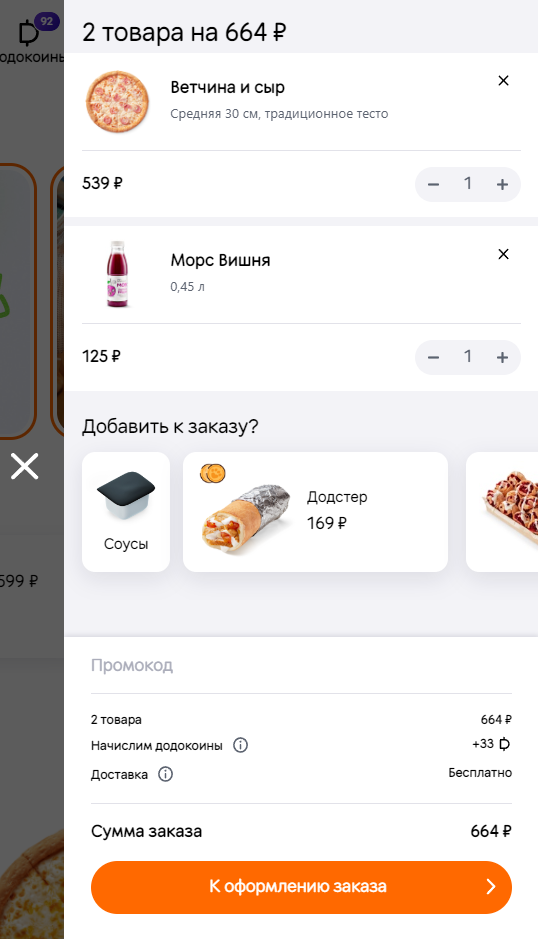


Рисунок 3. Корзина с товарами.

При оформлении заказа необходимо указать адрес доставки, после чего происходит оплата заказа, рис. 4.

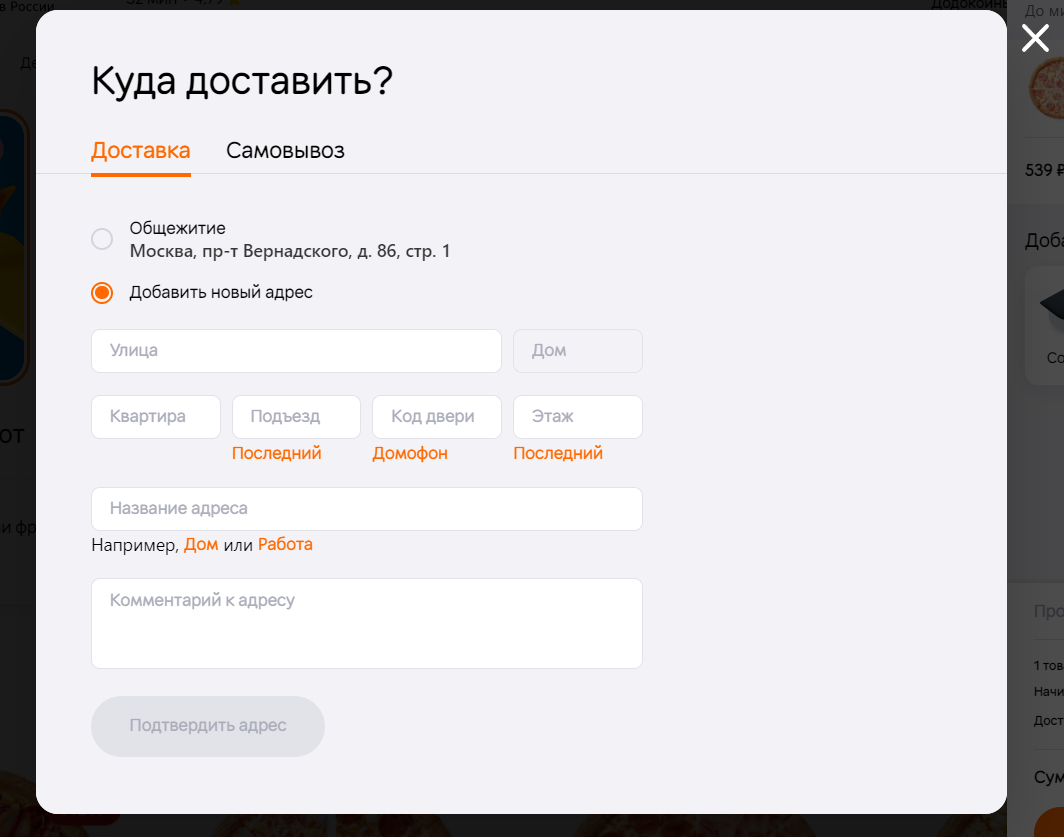


Рисунок 4. Выбор адреса доставки.

Интернет-ресурс «Додо Пицца» будет взят за основу разрабатываемого интернет-ресурса: виды пиццы, описания, цены, добавление в корзину, оформление заказа на доставку.

Также рассмотрим другой аналог разрабатываемого интернет-ресурса: «Вкусно – и точка», рис. 5.

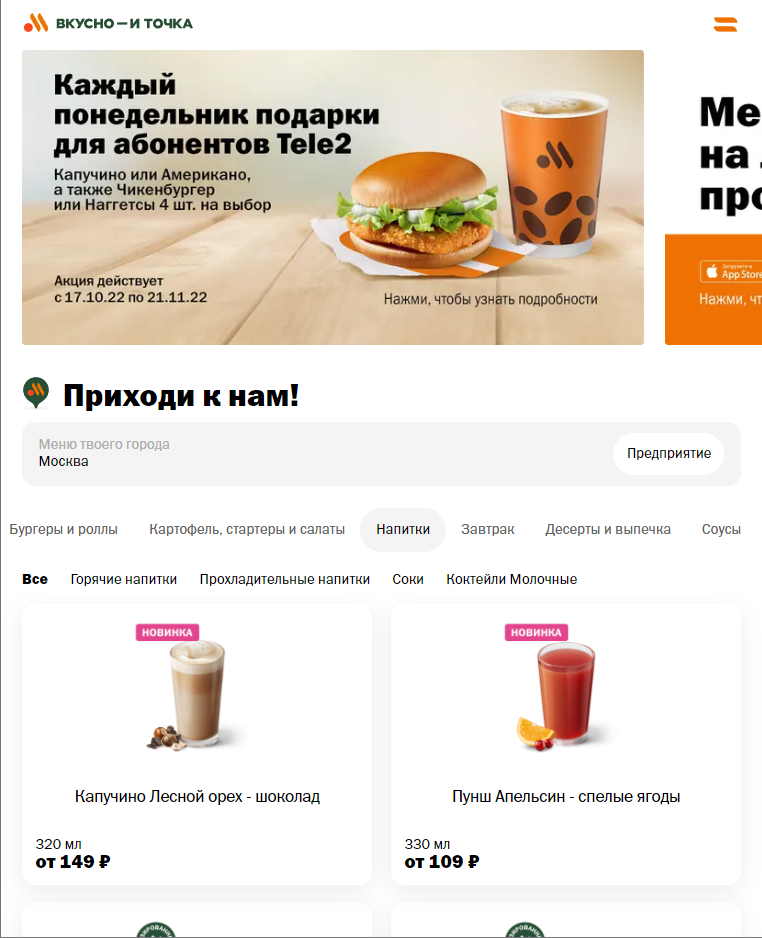


Рисунок 5. Главная страница интернет-ресурса «Вкусно – и точка».

Главная страница построена схожим образом: каталог товаров, категории, акции. В пиццерии обычно не продается одна лишь пицца, к ней можно приобрести закуски, напитки, десерты. Сопутствующие к пицце товары будут взяты именно с интернет-ресурса «Вкусно – и точка».

Также можно заметить, что в аналогах в основе цветовой палитры лежит два цвета: белый и оранжевый. Теплый оранжевый цвет часто используется в ресторанах. Это цвет также будет использоваться в разрабатываемом интернет-ресурсе.

## 2.2. Выбор технологий разработки клиентской части интернет-ресурса

Каждый интернет-ресурс в настоящее время основывается на трёх технологиях: HTML, CSS, JavaScript.

HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») – стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. С помощью этого языка формируется структура сайта.

CSS (от англ. Cascading Style Sheets «каскадные таблицы стилей») – формальный язык описания внешнего вида веб-страницы, написанной с использованием языка разметки HTML. Таким образом, структуре сайта, созданной на HTML, придается определенный внешний вид.

JavaScript (JS) – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. JS используется для придания интерактивности веб-страницам.

В разрабатываемом интернет-ресурсе будет использоваться типизированный JavaScript, а именно TypeScript.

TypeScript — язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript. TypeScript позволяет быстрее и проще писать комплексные решения, которые в дальнейшем будет легче развивать и тестировать, благодаря наличию строгой типизации.

Интерактивный пользовательский интерфейс в разрабатываемом интернет-ресурсе будет создан с использованием библиотеки React.

React – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. Его цель — предоставить высокую скорость разработки, простоту и масштабируемость.

Весь код интернет-ресурса будет написан в редакторе Visual Studio Code.

Тестирование будет проводиться в браузере Google Chrome.

**2.3 Создание веб-страниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript**

Для разрабатываемого интернет-ресурса была разработана серверная часть (backend) для хранения каталога товаров, формирования заказов, авторизации пользователей. Все данные для работы пиццерии хранятся в СУБД SQLite. Взаимодействие с сервером построено по технологии REST API, рис 6.

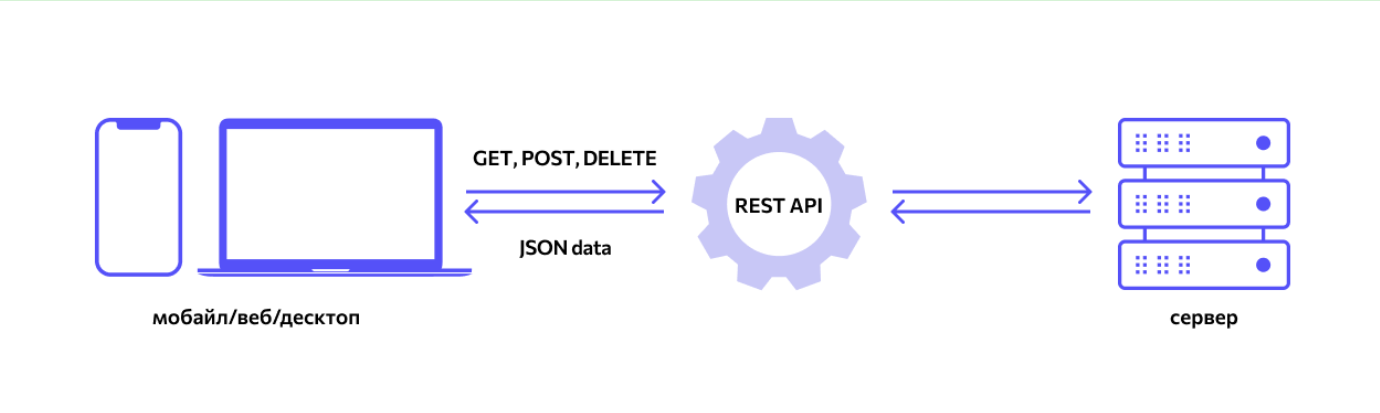


Рисунок 6. Схема REST API взаимодействия.

Клиентская часть интернет-ресурса отправляет запросы по протоколу HTTP и получает ответ в формате JSON, который затем обрабатывает и отображает конечному пользователю. API маршруты показаны на рис 7.



Рисунок 7. Маршруты API на языке Python.

Рассмотрим, как происходит наполнение веб-сайта контентом. Например, заполнение каталога товаров.

При открытии веб-страницы клиент отправляет асинхронный запрос на сервер с использованием метода GET, рис. 8.

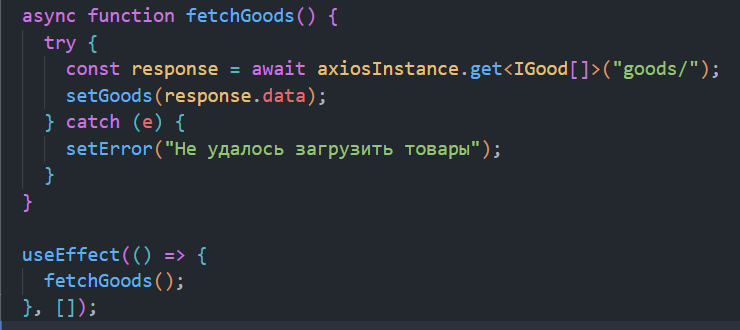


Рисунок 8. Код для отправки GET-запроса на сервер для получения каталога товаров.

В ответе сервер присылает массив товаров, в формате JSON, рис. 9.

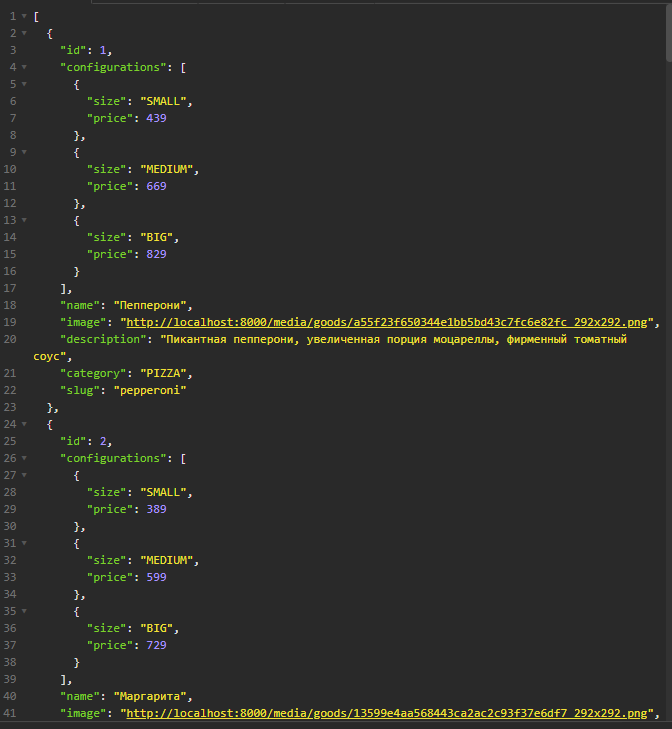


Рисунок 9. Пример ответа сервера.

Полученные данные записываются в массив товаров IGood[], рис. 10.



Рисунок 10. Интерфейс IGood для хранения товаров.

Затем с помощью цикла для каждого товара создаётся React-компонент GoodCard, который визуализирует полученные товары, рис. 11.



Рисунок 11. Компонент GoodCard.

Компоненты GoodCard располагаются в CSS Grid Layout, что обеспечивает адаптивность. Благодаря этому товары легко масштабируются под размер экрана, рис 12.

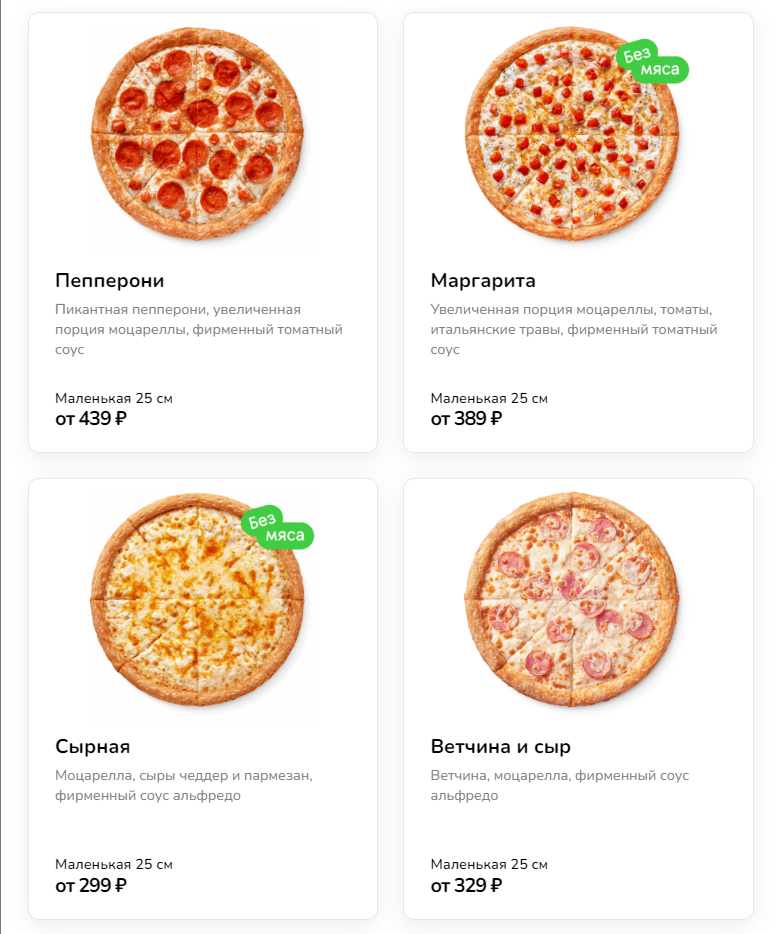


Рисунок 12. Отображение каталога товаров на веб-странице.

Элементы на карточке товара располагаются с помощью CSS Flexbox. Фрагмент CSS-стилей для карточки товара изображен на рис. 13.

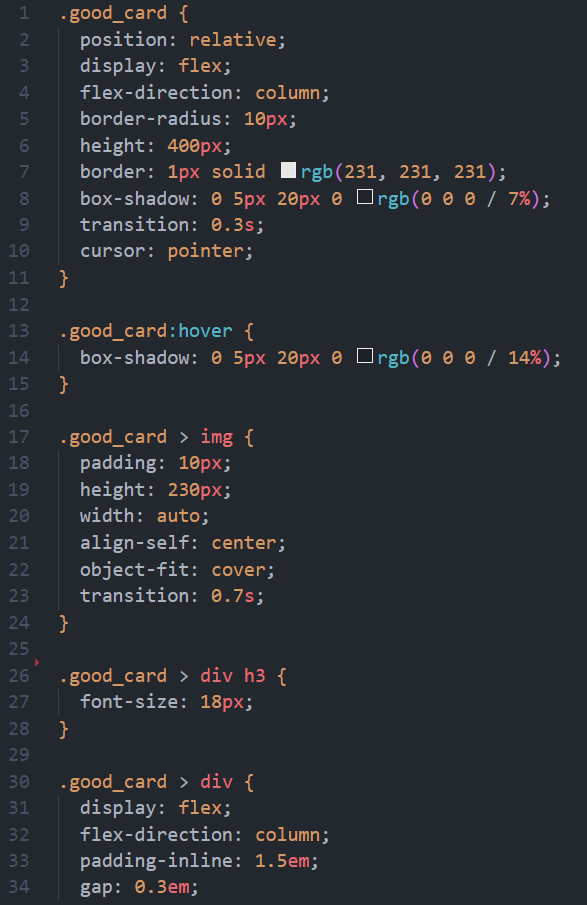


Рисунок 13. Фрагмент CSS-стилей карточки товара.

## 2.4 Создание межстраничной навигации

Межстраничная навигация реализована с помощью React Router. В зависимости от URL отображаются различные React-компоненты, рис. 14.



Рисунок 14. Фрагмент кода межстраничной навигации.

Межстраничная навигация интернет-ресурса изображена на рис. 15.

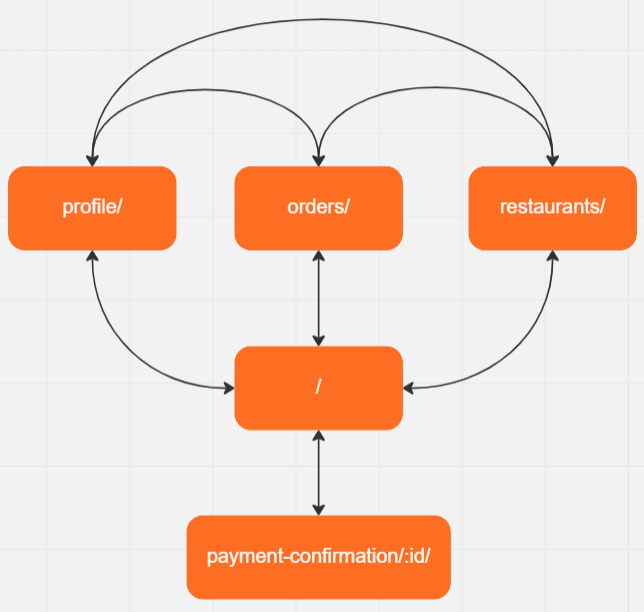


Рисунок 15. Схема межстраничной навигации.

Межстраничная навигация осуществляется при нажатии на ссылки в «шапке» веб-страницы, изображенной на рис. 16.

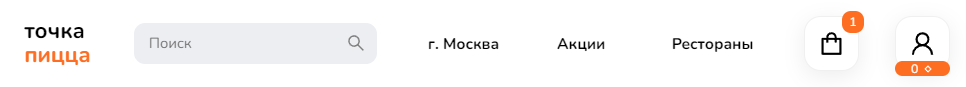


Рисунок 16. «Шапка» веб-страницы.

## 2.5 Реализация слоя клиентской логики веб-страниц с применением технологии JavaScript

Рассмотрим клиентскую логику веб-страниц на примере оформления заказа. Для начала клиенту необходимо авторизоваться. Нажмем на иконку пользователя в «шапке» веб-страницы, откроется окно входа, рис. 17. Код, запускаемый при нажатии на иконку пользователя, изображен на рис. 18, 19.

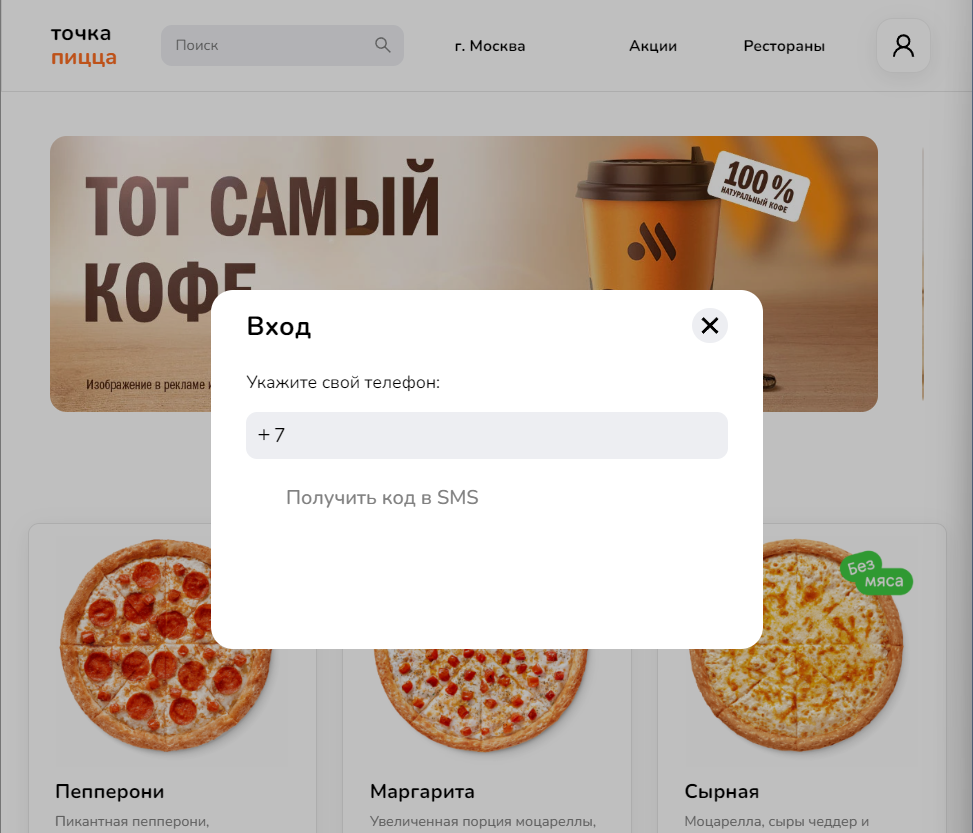


Рисунок 17. Окно входа.

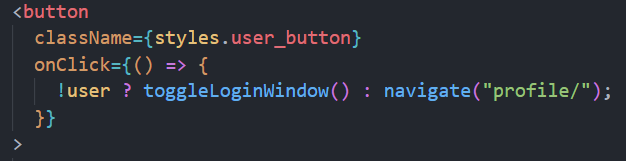


Рисунок 18. Код открытия окна входа.

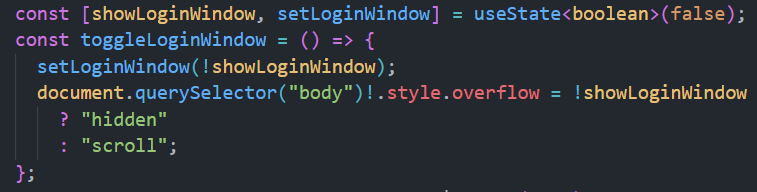


Рисунок 19. Функция изменения состояния окна входа.

Введем номер телефона и нажмём на кнопку получения кода в СМС. Будет отправлен POST-запрос на сервер с введённым пользователем номером, рис. 20, а окно входа поменяет своё состояние, рис. 21.

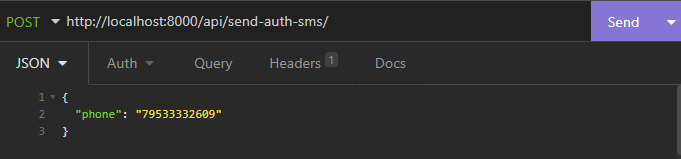


Рисунок 20. Пример запроса на отправку СМС с кодом подтверждения.

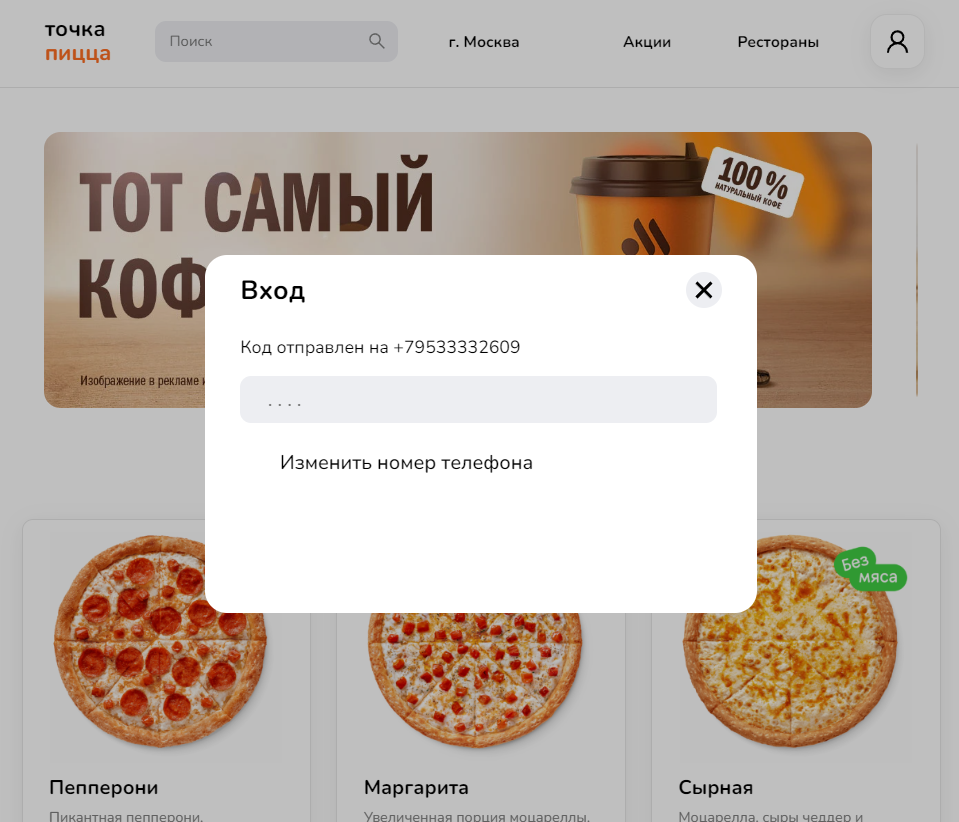


Рисунок 21. Ввод полученного кода подтверждения.

После этого сервер отправит запрос стороннему серверу, предоставляющему услуги отправки СМС и на номер пользователя будет отправлен код подтверждения, рис. 22.

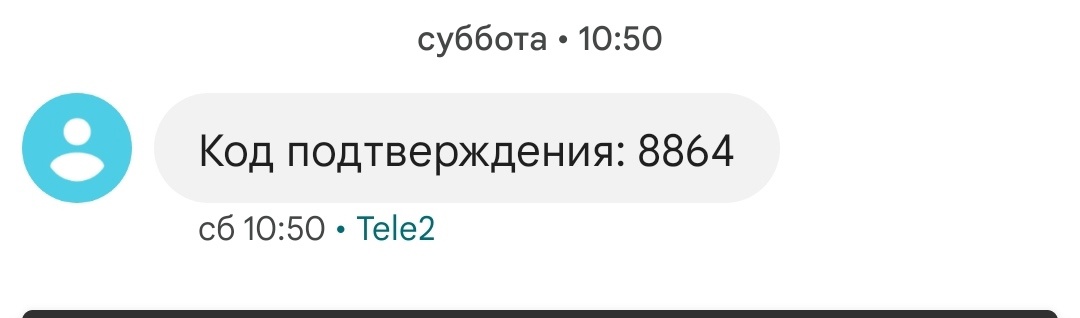


Рисунок 22. Полученный в СМС код подтверждения

Введём полученный код, произойдёт успешная авторизация. «Шапка» изменит своё состояние, рис. 23.

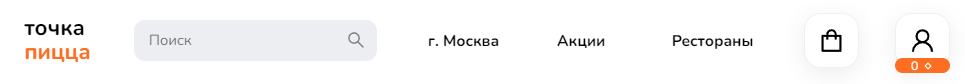


Рисунок 23. «Шапка» авторизованного пользователя.

Клиентом был отправлен POST-запрос с телефоном и кодом на сервер. В ответе клиент получил токен авторизации, который записал в локальное хранилище. Этот токен затем будет отправляться в заголовке запросов для аутентификации пользователя. Код авторизации пользователя изображен на рис. 24.



Рисунок 24. Код авторизации пользователя.

Теперь пользователь авторизован, добавим товары в корзину. К примеру, нажмем на товар «Маргарита». Появится окно выбора диаметра пиццы, рис 25.

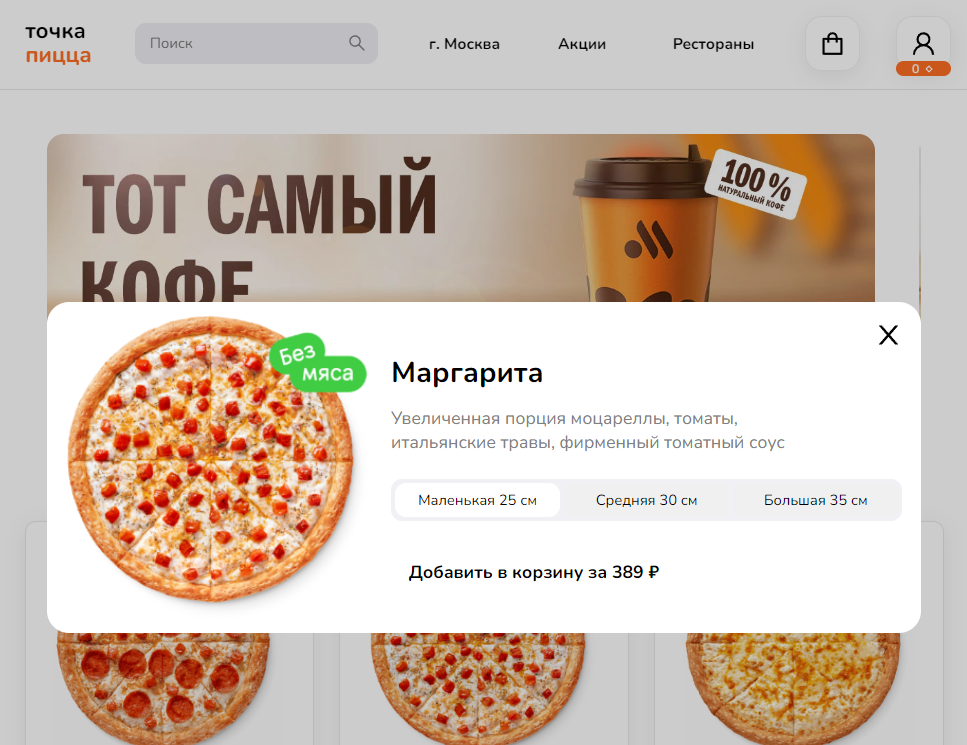


Рисунок 25. Выбора диаметра пиццы «Маргарита».

Выберем 30 см и добавим в корзину. Также добавим напиток. Для этого переключим категорию на напитки, рис. 26, и добавим товар в корзину. Код React-компонента категорий изображен на рис 27.

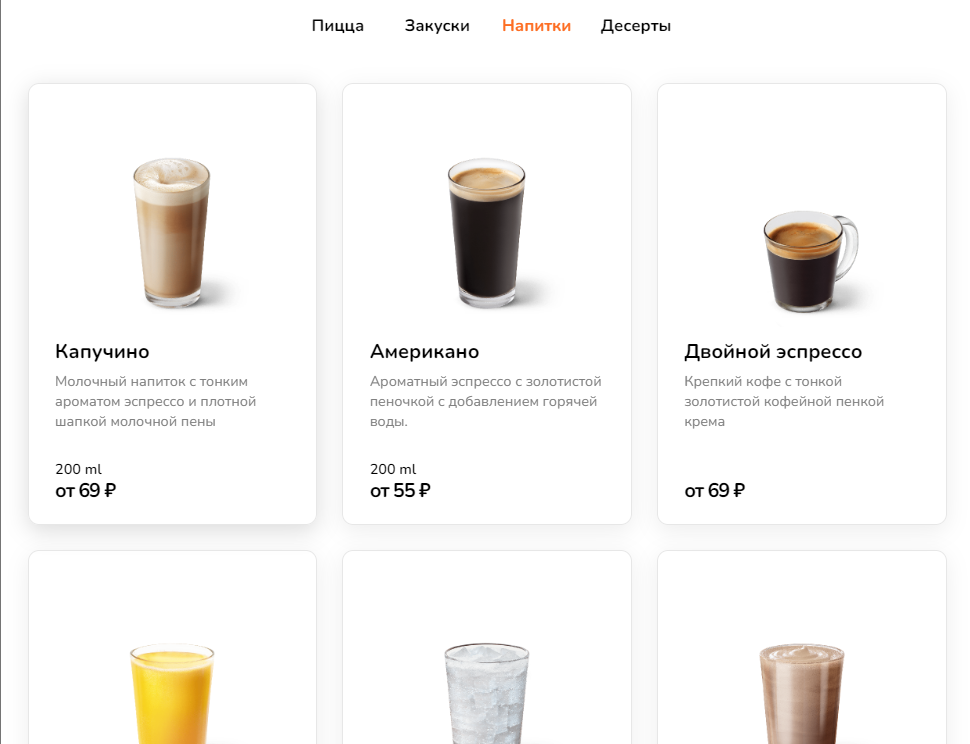


Рисунок 26. Выбор напитка.

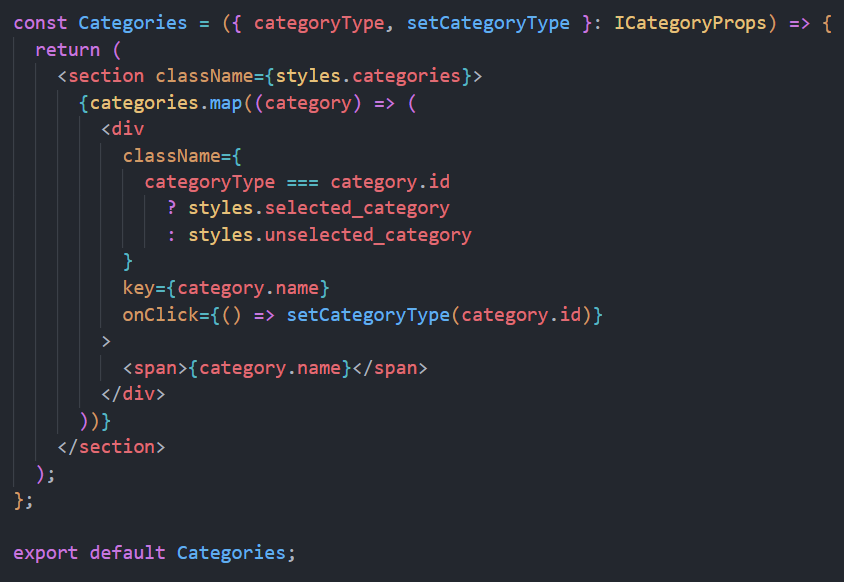


Рисунок 27. React-компонент категорий товаров.

Добавим еще несколько товаров и откроем корзину, нажатием на иконку корзины в «шапке», рис 28.

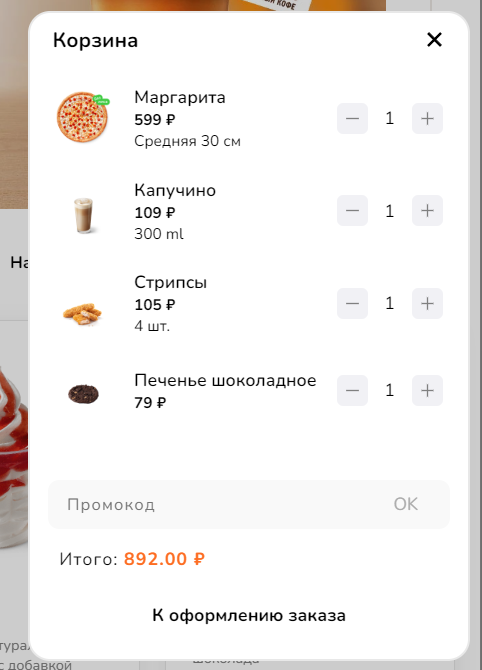


Рисунок 28. Корзина с товарами.

Добавим ещё одну пиццу нажатием на иконку «+», сработает код изображенный на рис. 29.



Рисунок 29. Код изменения количества определенного товара.

В корзине количество товара «Маргарита» изменится на «2».

Также добавим промокод на скидку. Введём «МИРЭА» в поле для ввода и нажмём кнопку «OK». Будет отправлен POST-запрос на сервер с введённым промокодом (рис. 30), и пользователь получит скидку 10%, рис 31.

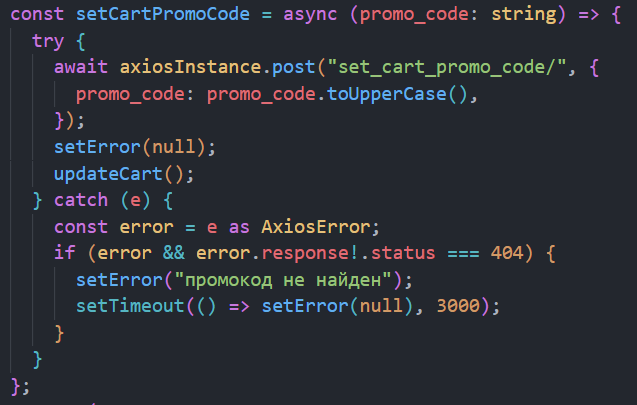


Рисунок 30. Код установки промокода.

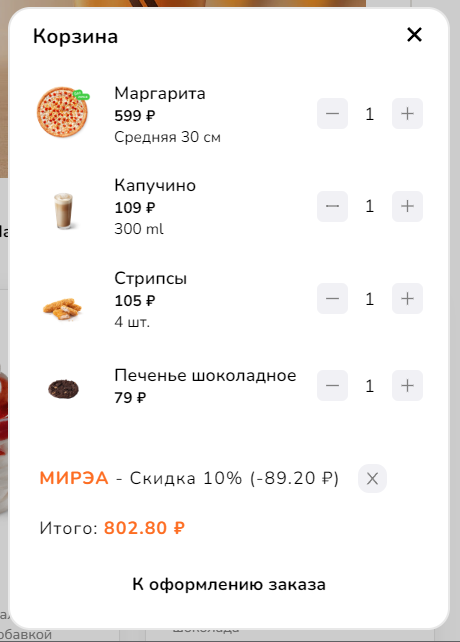


Рисунок 31. Корзина с применённым промокодом «МИРЭА».

Далее перейдем к оформлению заказа. Нажмем на кнопку «К оформлению заказа». Откроется окно выбора адреса доставки, рис. 32.

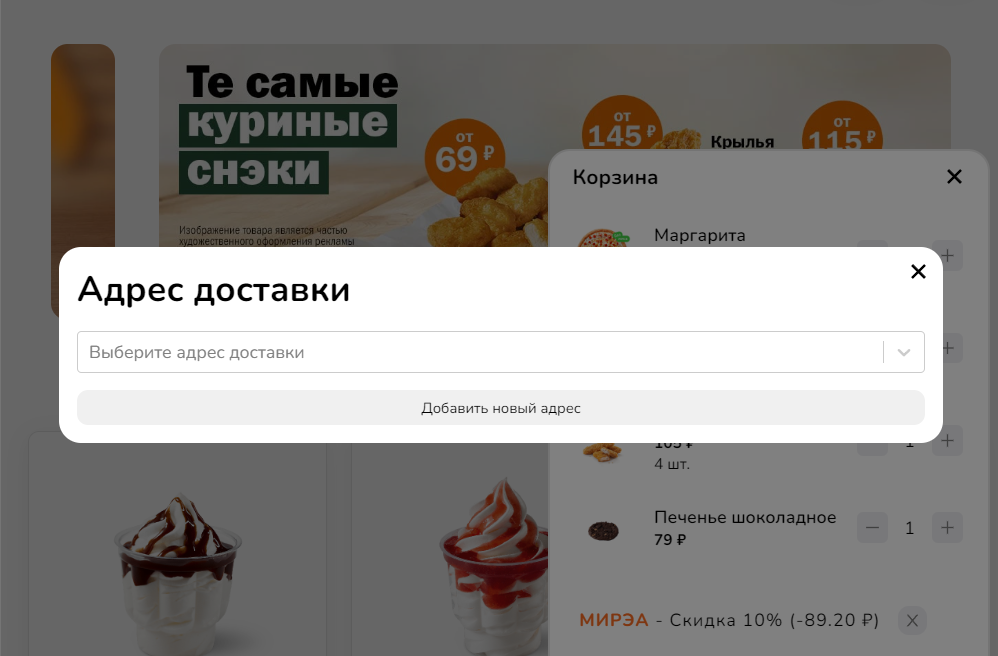


Рисунок 32. Выбор адреса доставки.

Добавим новый адрес, нажатием на кнопку «Добавить новый адрес», появится окно с добавлением нового адреса, рис. 33.

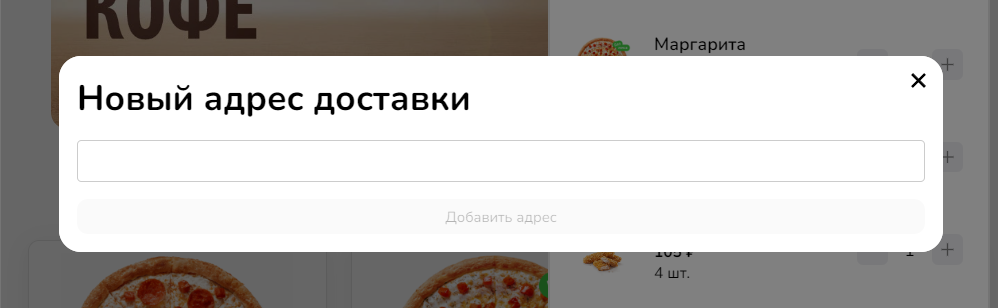


Рисунок 33. Добавление нового адреса доставки.

Начнём вводить адрес, появятся варианты автозаполнения из стороннего API, рис 34.

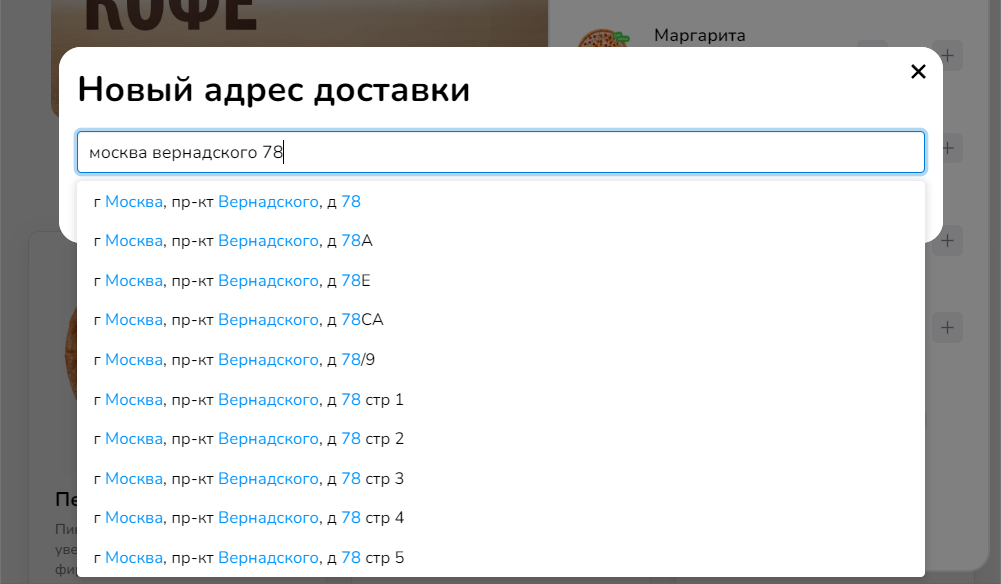


Рисунок 34. Варианты автозаполнения адреса доставки.

Выберем нужный адрес и появится стороняя карта с маркером, показывающая адрес доставки, рис 35.

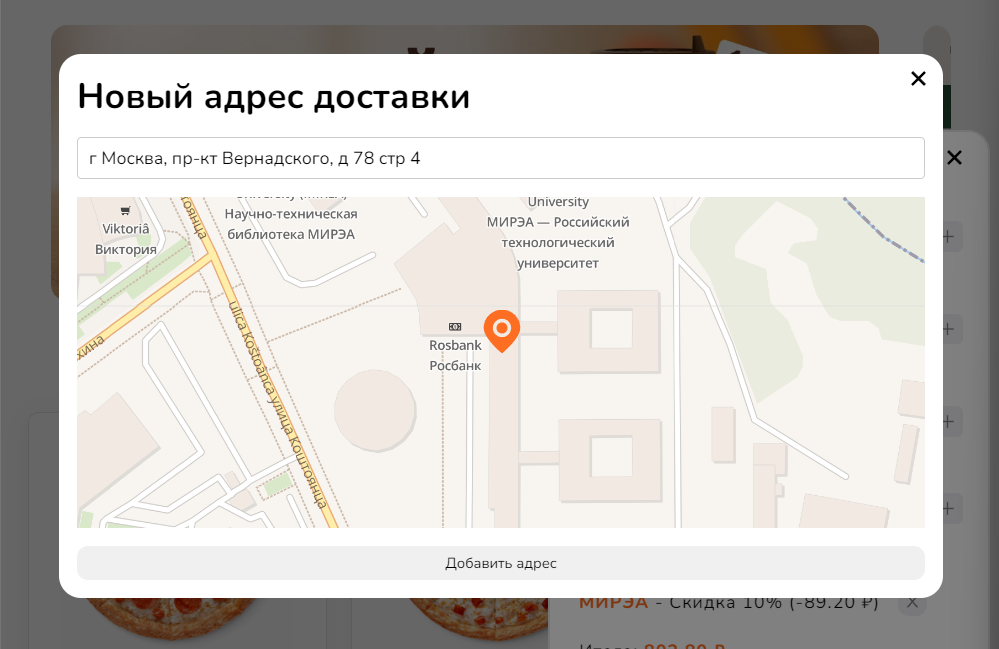


Рисунок 35. Карта с указанным адресом.

Нажмём на кнопку «Добавить адрес». Запустится функция добавления нового адреса доставки, отправляющая POST-запрос на сервер, рис. 36.

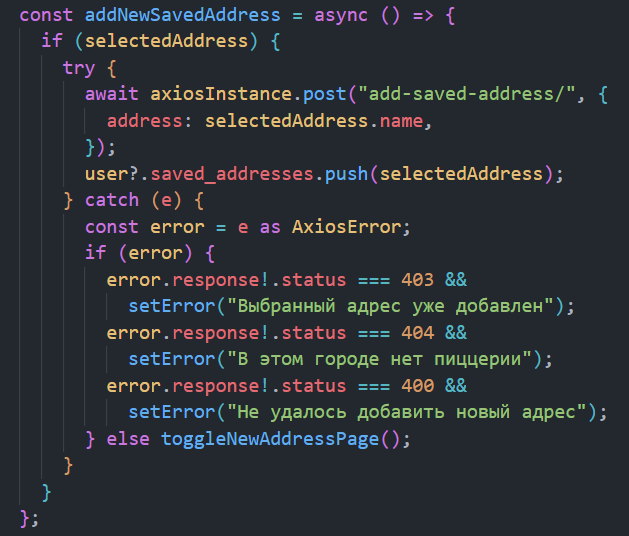


Рисунок 36. Функция добавления нового адреса доставки.

Далее выберем из списка добавленный адрес, рис 37.

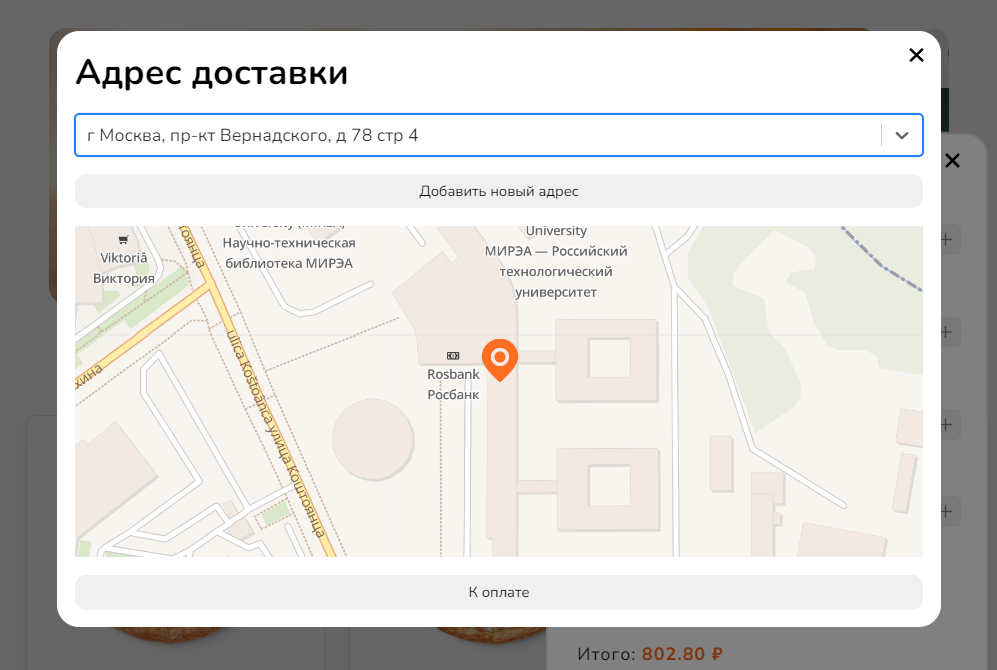


Рисунок 37. Выбор адреса доставки.

Затем нажмем на кнопку «К оплате». Отправится POST-запрос на сервер (рис 38.), в ответе будет получен URL для оплаты, произойдет переадресация на стороннюю страницу оплаты, рис 39, 40.

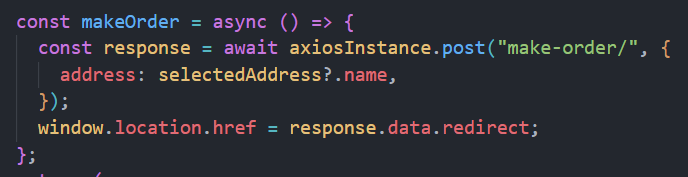


Рисунок 38. Функция создания заказа.

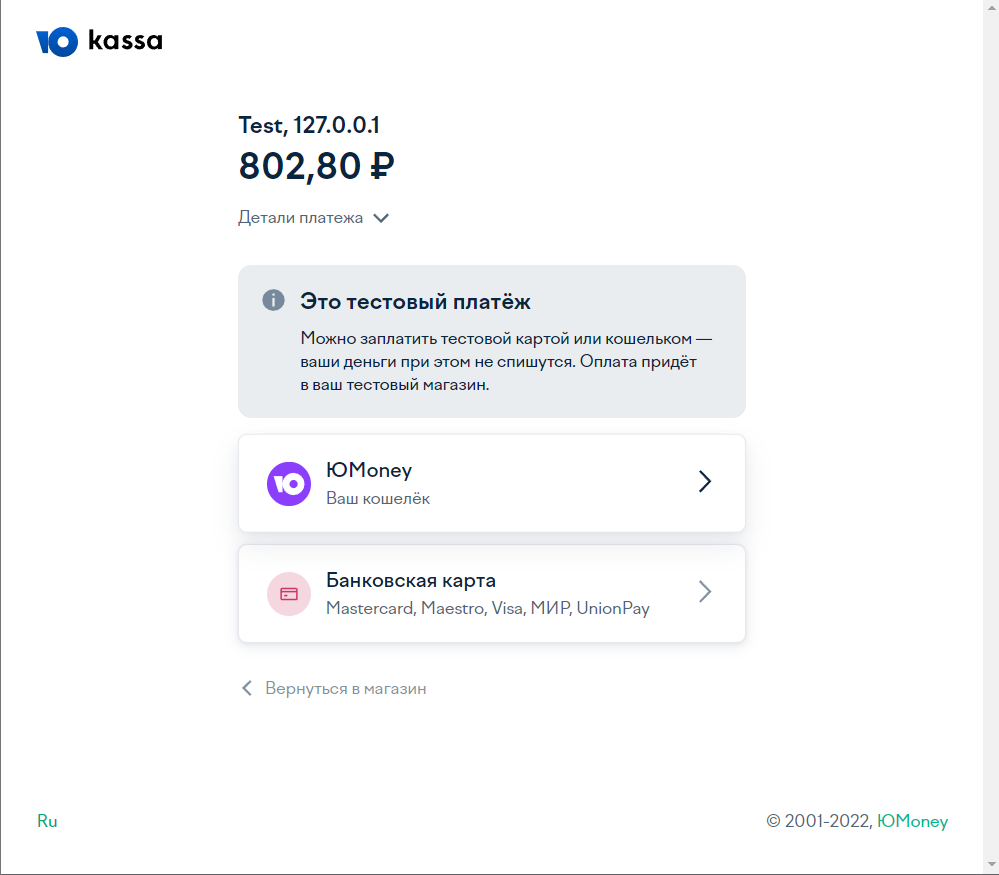


Рисунок 39. Оплата заказа.

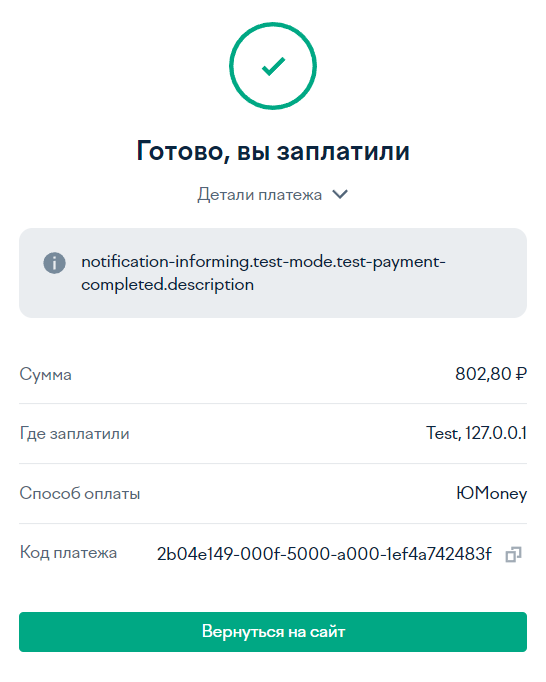


Рисунок 40. Оплаченный заказ.

Нажмём на кнопку «Вернуться на сайт», откроется страница подтверждения оплаты заказа, рис. 41. При переходе на эту страницу каждые 5 секунд будет отправляться GET-запрос с подтверждением оплаты, рис 42. Заказ можно отменить, либо оплатить, если он не был оплачен.

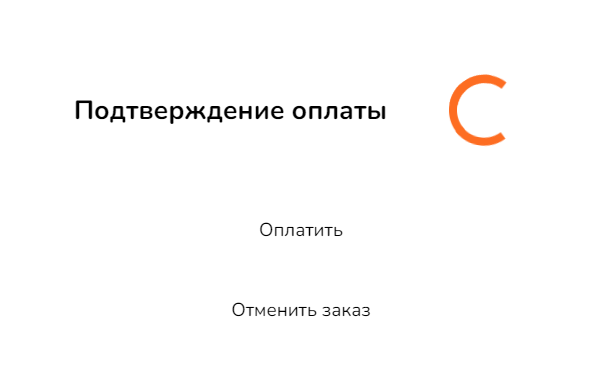


Рисунок 41. Подтверждение оплаты.



Рисунок 42. Код подтверждения оплаты.

После подтверждения оплаты будет написано «Оплата прошло успешно» (рис. 43), и произойдет переадресация на главную страницу.

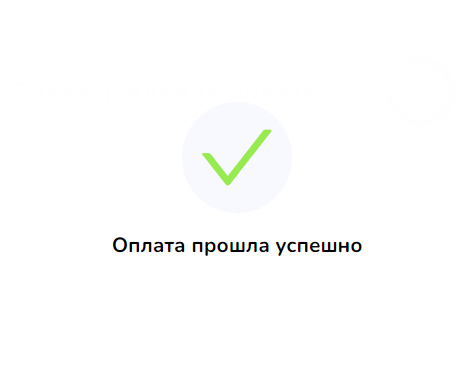


Рисунок 43. Успешная оплата заказа.

Заказ был создан и оплачен, и «шапка» веб-страницы поменяла вид, рис. 44.



Рисунок 44. В «Шапке» появилась кнопка «Активный заказ».

При нажатии на кнопку «Активный Заказ» открывается страница с активным заказом, рис. 45.

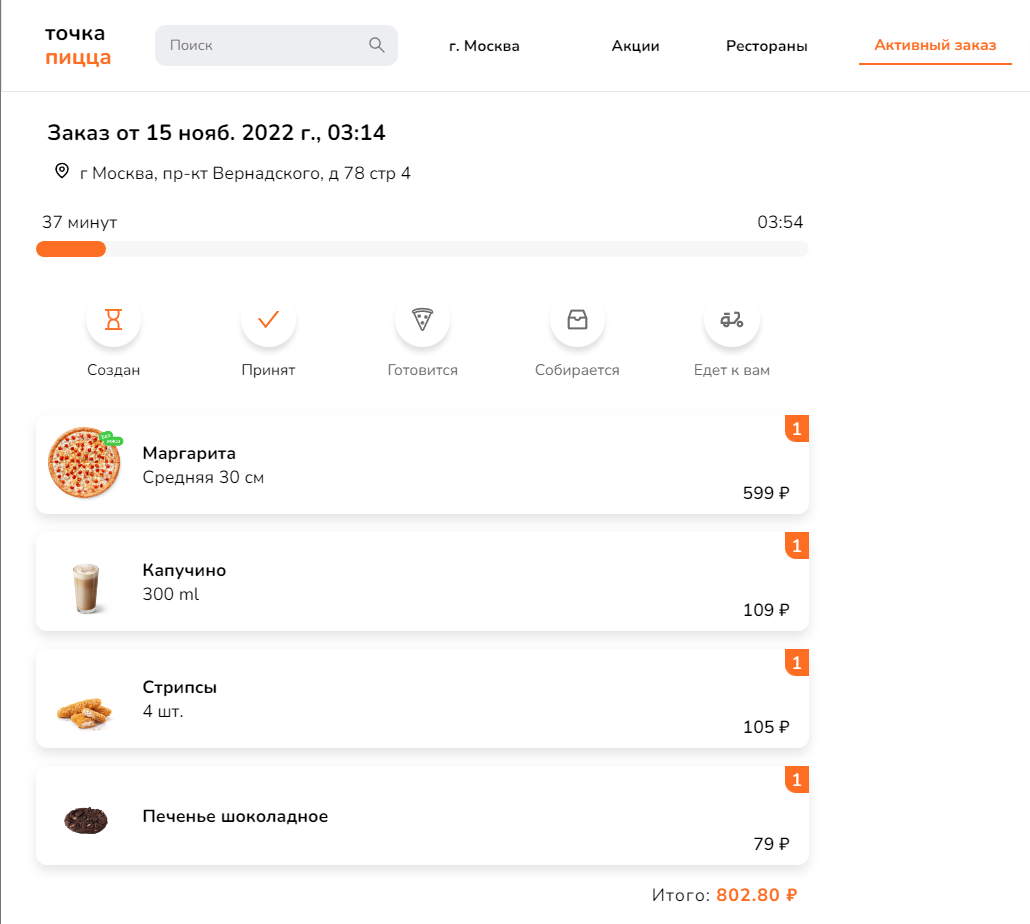


Рисунок 45. Страница с активным заказом.

Можно увидеть данные о заказе: время заказа, адрес доставки, примерное время ожидания, статусы заказа, список товаров, итоговая сумма.

Статусы обновляются в реальном времени, благодаря использованию WebSocket соединения, рис 46.



Рисунок 46. Обновление статуса заказа с использованием WebSocket соединения.