|  |  |
| --- | --- |
|  | Министерство образования и науки Российской Федерации  Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н.Ельцина» (УрФУ)  ИРИТ-РТФ  Базовая кафедра «Аналитика больших данных и методы видео анализа» |

Оценка по проекту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель проекта \_\_Медведев M.A.\_\_\_

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет по проекту на тему

Доставка еды «ИРИТ-КАФЕ»

Студенты: Смирнов Никита Сергеевич   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пономаренко Павел Андреевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО) (Подпись)

Группа: РИ-311055

Екатеринбург

2024

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc168440374)

[АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc168440375)

[ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ 5](#_Toc168440376)

[ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РАЗРАБОТКИ 7](#_Toc168440377)

[ДИЗАЙН-ШАБЛОНЫ СТРАНИЦ ПРОЕКТА 8](#_Toc168440378)

[РАЗРАБОТКА КЛИЕНТСКОЙ И СЕРВЕРНОЙ ЧАСТИ ПРИЛОЖЕНИЯ 11](#_Toc168440379)

[ЗАКЛЮЧЕНИЕ 27](#_Toc168440380)

# ВВЕДЕНИЕ

***Актуальность темы***

Создание онлайн-сервиса для заказа еды на дом актуально в современном обществе, так как позволяет экономить время и усилия пользователей. С ростом популярности онлайн-услуг потребность в удобных и функциональных платформах возрастает, особенно в условиях активного ритма жизни.

***Цель работы***

Целью данной работы является разработка веб-приложения для онлайн-заказа еды с удобным и интуитивно понятным интерфейсом, который обеспечит быстрый и простой процесс оформления заказа.

***Задачи***

* Выбрать инструменты и технологии для разработки.
* Разработать прототипы и дизайн-шаблоны.
* Реализовать клиентскую часть приложения.
* Провести тестирование и отладку системы.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Концепция данного проекта предполагает создание веб-платформы интернет-магазина, где пользователи смогут просматривать и приобретать товары. Система будет включать в себя следующие основные компоненты:

* Пользователи:

Активные участники системы, которые посещают веб-сайт магазина, просматривают товары и осуществляют покупки.

* Товары:

Широкий ассортимент продукции, представленный на веб-сайте для продажи.

* Корзина:

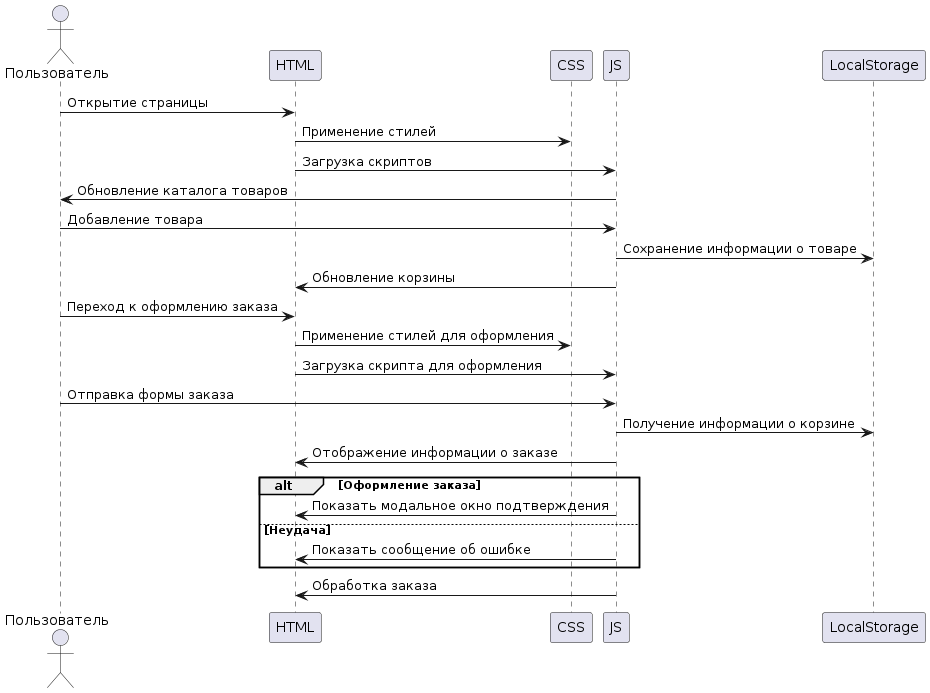
Важный функционал, который позволяет пользователям добавлять товары в корзину перед оформлением заказа. Корзина обеспечивает удобное собирание выбранных товаров и подготовку их к финальному оформлению покупки.

* Форма заказа:

Интерфейс, предоставляющий пользователям возможность ввода персональной информации, адреса доставки и выбора методов оплаты. Форма заказа является важным инструментом для оформления покупок и завершения процесса транзакции.

Эти компоненты будут интегрированы в систему интернет-магазина, обеспечивая пользователям удобный и эффективный процесс покупок онлайн.

# ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ



**Посещение главной страницы:**

Пользователь заходит на главную страницу сайта, где отображается широкий выбор товаров из каталога.

**Выбор и добавление товаров:**

Пользователь выбирает необходимые товары, просматривая каталог, и добавляет их в корзину, используя соответствующие кнопки или иконки.

**Просмотр содержимого корзины:**

Пользователь просматривает содержимое корзины, чтобы удостовериться, что выбрал все нужные товары перед оформлением заказа.

**Оформление заказа:**

Пользователь заполняет форму заказа, предоставляя свои контактные данные и информацию о доставке.

**Проверка данных и подтверждение заказа:**

Интерфейс проверяет правильность заполнения формы заказа.

В случае успешного заполнения формы, пользователю показывается модальное окно с подтверждением заказа.

Если данные формы заполнены некорректно, пользователь получает сообщения об ошибках для исправления.

**Обработка заказа:**

Интерфейс обрабатывает заказ, сохраняя информацию о нем в локальном хранилище браузера. Это может понадобиться для последующего отображения пользователю или для дальнейшей обработки заказа.

# ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РАЗРАБОТКИ

**HTML:**

HTML играет ключевую роль в организации содержимого веб-страницы. В нашем проекте HTML используется для создания структуры страницы, включая разметку каталога товаров, форму заказа и другие важные элементы интерфейса. Он определяет, как содержимое будет отображаться и взаимодействовать с пользователем.

**CSS:**

CSS служит для стилизации веб-страницы и придания ей привлекательного внешнего вида. В нашем проекте CSS используется для определения визуального оформления элементов страницы. Это позволяет создать единый и согласованный дизайн, что улучшает пользовательский опыт и делает сайт более привлекательным.

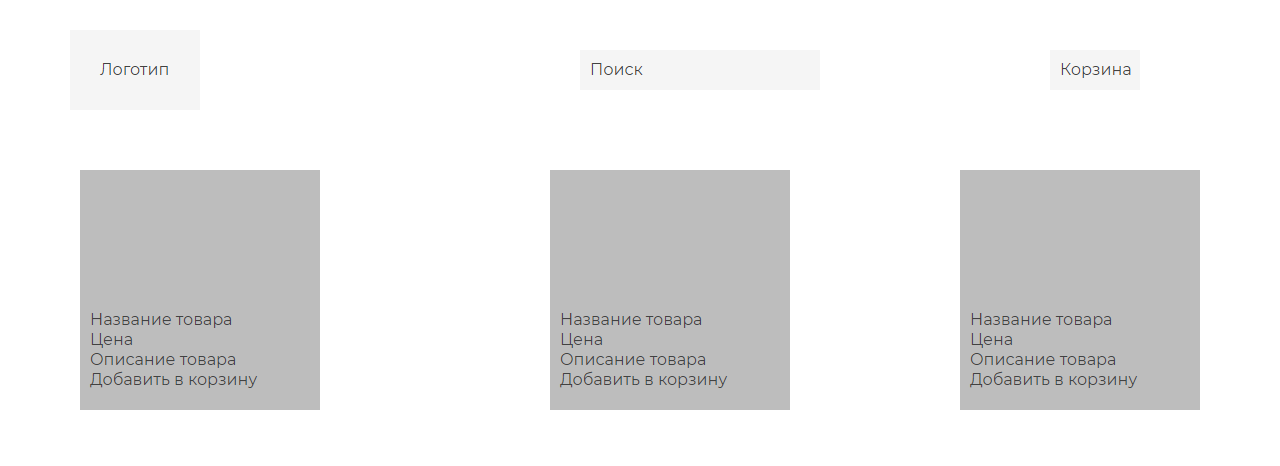
**JavaScript:**

JavaScript добавляет интерактивность и динамичность нашему веб-сайту. В проекте JavaScript применяется для создания интерактивных элементов. Это делает сайт более функциональным и удобным для пользователей, улучшая их взаимодействие с сайтом.

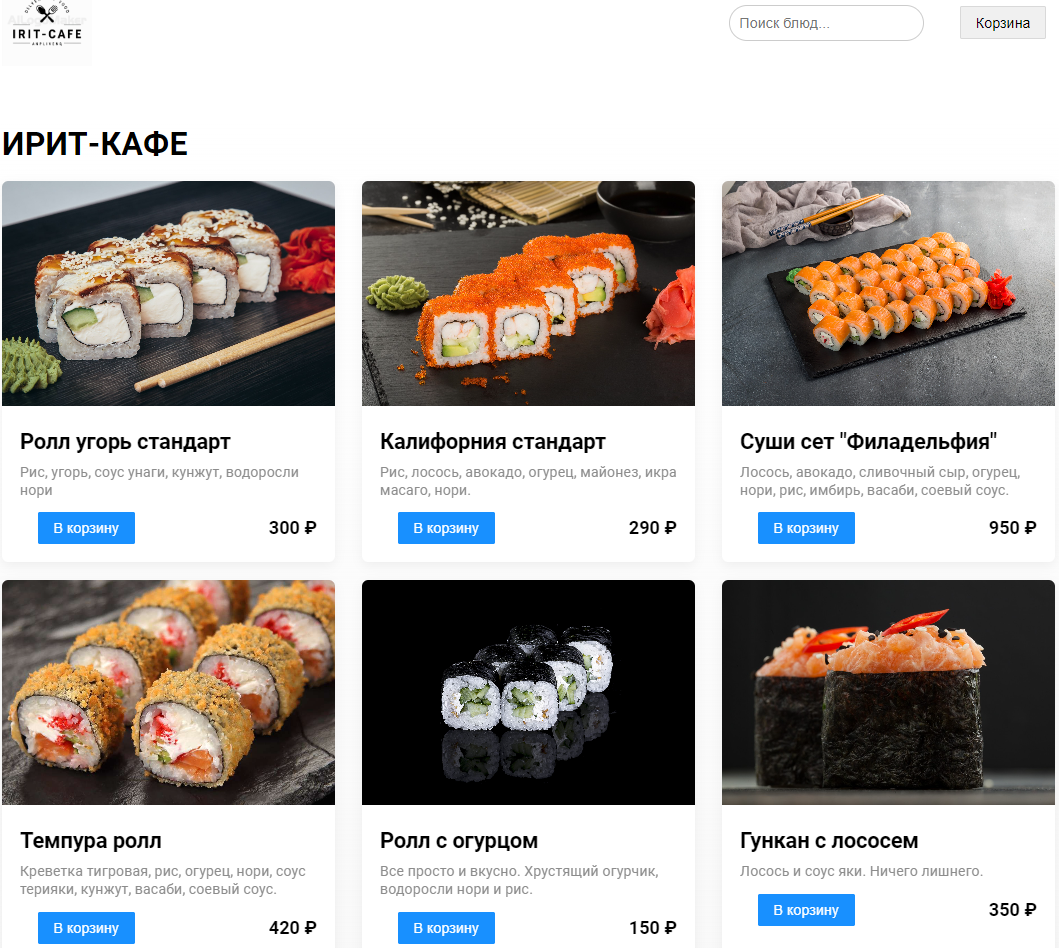
**PlantUML**:

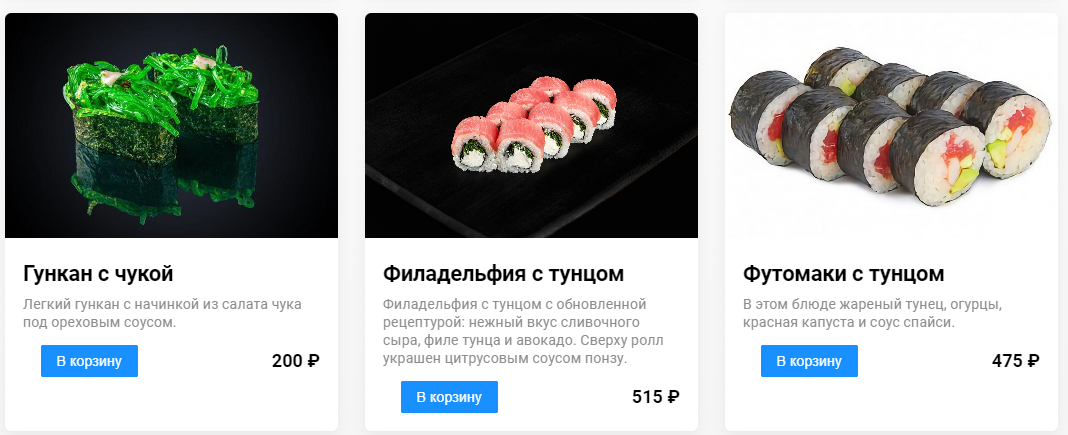
В нашем проекте мы использовали PlantUML для создания блок-схемы, которая помогла нам визуализировать взаимодействие между различными компонентами в системе разработки веб-приложения.

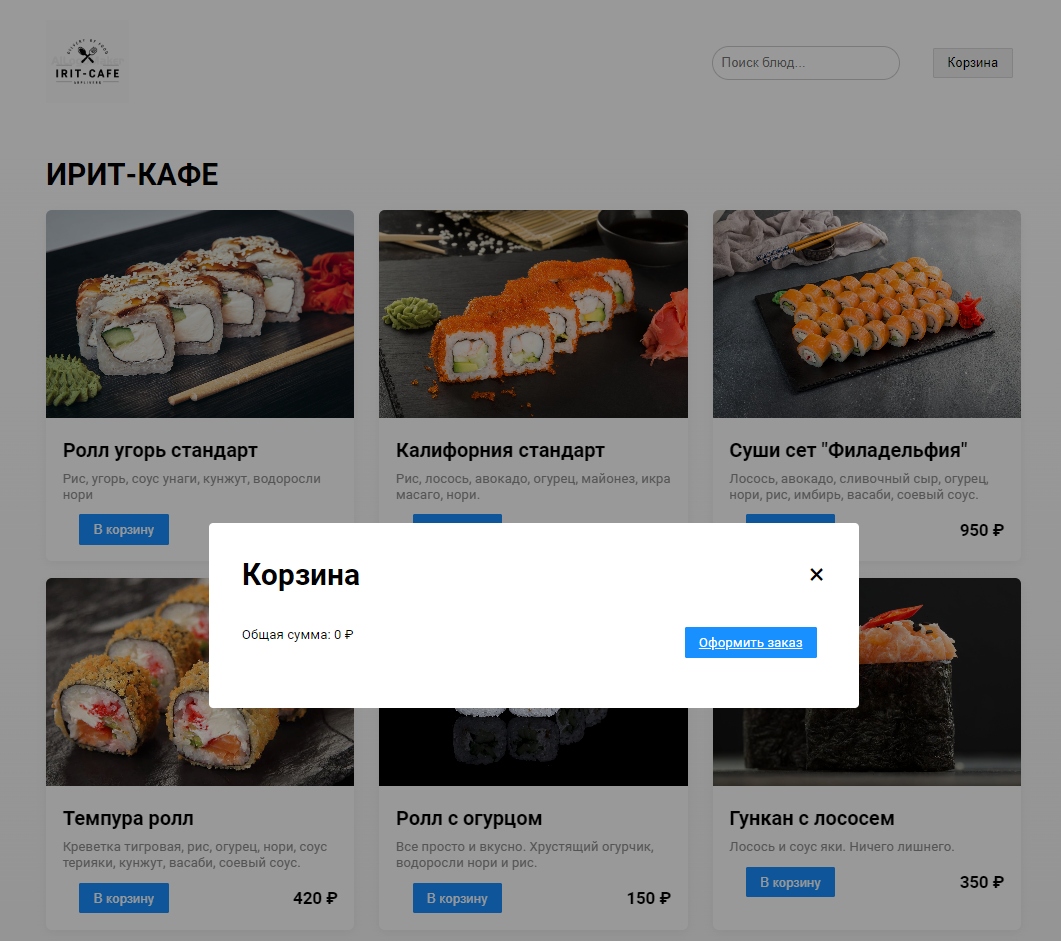
# ДИЗАЙН-ШАБЛОНЫ И ПРОТОТИП СТРАНИЦ ПРОЕКТА



Прототип главной страницы сайта





Изображение выглядит как текст, снимок экрана, еда

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

На главной странице сайта пользователи встречаются с каталогом товаров, представленных в удобном формате карточек. Эти карточки содержат изображения товаров, их названия, описание и цены. Пользователи могут просматривать товары, использовать функцию поиска для быстрого доступа к нужным позициям, а также добавлять выбранные товары в корзину.

Окно корзины позволяет пользователям просматривать содержимое и управлять товарами, которые они добавили. Они могут изменять количество товаров, удалять позиции, а также просматривать общую стоимость заказа. Также на этой странице есть кнопка оформления заказа.

Страница оформления заказа включает форму, в которую пользователи вводят контактные данные и информацию о доставке. Они могут указать свое имя, адрес, контактный номер телефона и другие детали. После заполнения формы пользователи могут подтвердить заказ и отправить его на обработку.

# РАЗРАБОТКА КЛИЕНТСКОЙ И СЕРВЕРНОЙ ЧАСТИ ПРИЛОЖЕНИЯ

1. HTML «**restaurant.html**»

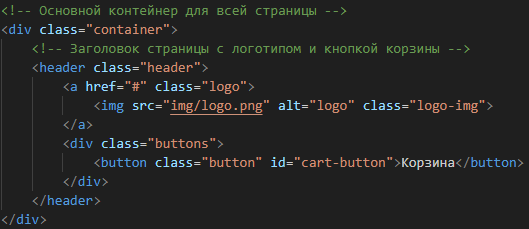


Рис. 1 - Шапка страницы

Этот код - как чертёж для создания веб-страницы. В нём есть область, где будет находиться вся страница, и внутри этой области есть заголовок. В заголовке есть картинка с логотипом и кнопка для перехода в корзину покупок. Кнопка будет отображать количество товаров в корзине. Также есть место для ввода текста, чтобы искать разные блюда на странице.

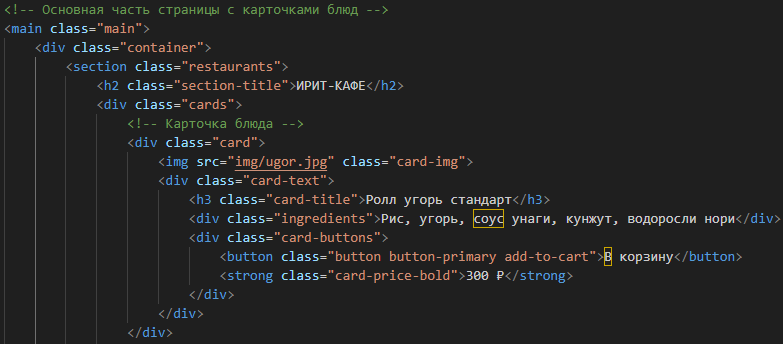


Рис. 2 - Основная часть

Данный блок представляет собой раздел основной части страницы сайта. Он содержит информацию о ресторане или кафе, название которого указано как «ИРИТ-КАФЕ». Здесь представлены карточки блюд, в которых есть изображение каждого блюда, его название, список ингредиентов и цена. Также есть кнопка «В корзину», которую можно нажать, чтобы добавить блюдо в корзину для покупок. В примере показана одна карточка с информацией о ролле с угрем и его стандартной ценой в 300 рублей.

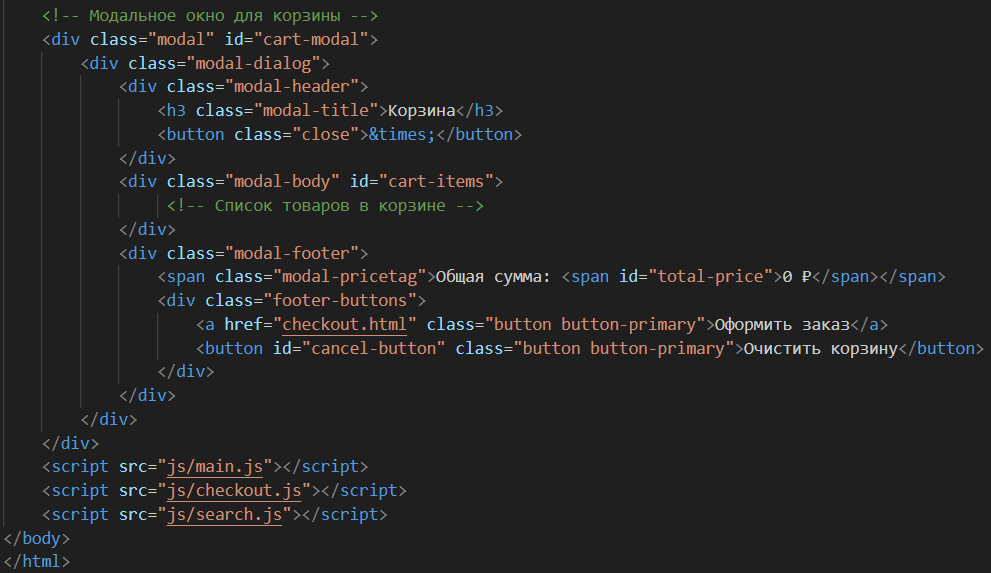


Рис. 3 - Модальное окно корзины, скрипт для работы с корзиной

Фрагмент кода создает модальное окно на веб-странице, которое отображает содержимое корзины покупок. Окно содержит заголовок «Корзина» и кнопку закрытия в верхней части. В середине окна предполагается отображение списка товаров, добавленных в корзину (в настоящее время эта область пуста). В нижней части окна показывается общая сумма стоимости всех товаров в корзине, а также кнопки для оформления заказа и очистки корзины.

1. HTML «**checkout.html**»

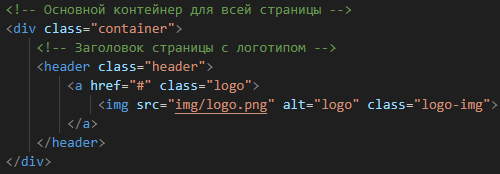


Рис. 4 – Шапка страницы

Код создает часть веб-страницы, где у нас есть основной контейнер для всего содержимого страницы, включая заголовок с логотипом. Логотип является кликабельной ссылкой, которая перенаправляет пользователя на страницу с основным меню.

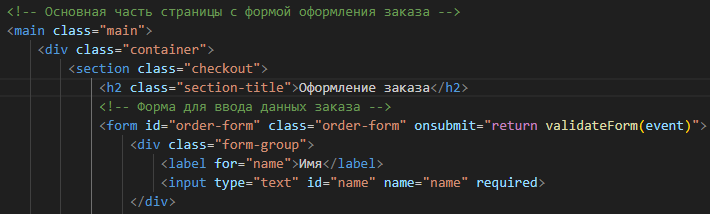


Рис. 5 - Форма оформления заказа

Код добавляет функциональность на страницу, где пользователь может заполнить форму для заказа. В форме есть поля, где пользователь может ввести своё имя, номер телефона, адрес доставки, выбрать способ оплаты и написать комментарий. После заполнения формы пользователь может нажать на кнопку для подтверждения заказа.



Рис. 6 - Скрипт для валидации формы

Подключается внешний JavaScript файл для валидации формы при отправке.

1. CSS «**style.css**» - формирует внешний вид веб-страницы, включая кнопки, формы, заголовки и другие элементы. Он определяет их стили, такие как цвета, размеры и расположение на странице. Также он содержит адаптацию, чтобы обеспечивалось хорошее отображение страницы на мобильных устройствах.

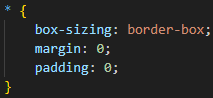


Рис. 7 - Сброс стилей и базовые настройки для всех элементов

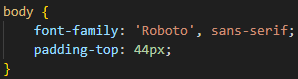


Рис. 8 - Стили для body

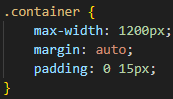


Рис. 9 - Основной контейнер

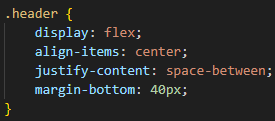


Рис. 10 - Шапка

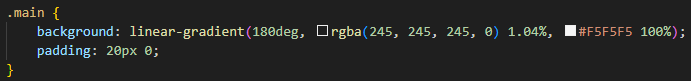


Рис. 11 - Основной контент страницы

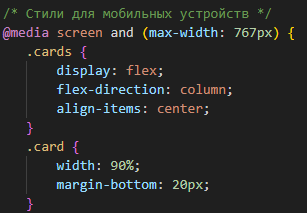


Рис. 12 - Адаптивные стили для мобильных устройств

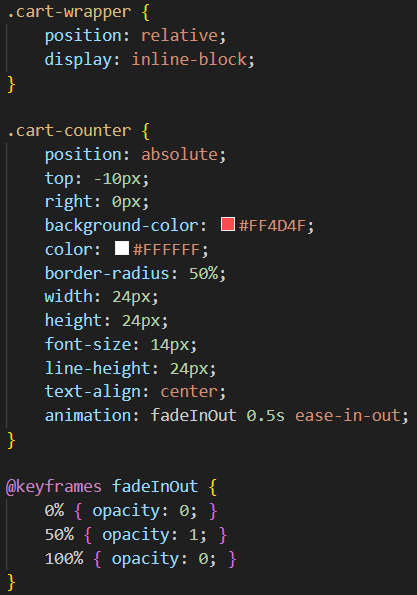


Рис. 13 – Контейнер для счётчика позиций в заказе, создан для объединения корзины и счётчика в один объект для адаптивности страницы

1. CSS «**animation.css**» - используется для добавления анимаций на веб-страницу. Он определяет, как элементы будут анимироваться при определенных действиях пользователя или событиях на странице, таких как наведение курсора или открытие модального окна.

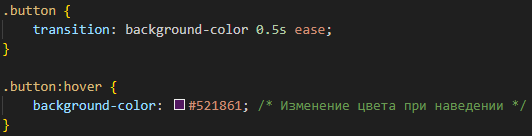


Рис. 14 - Анимация для кнопок

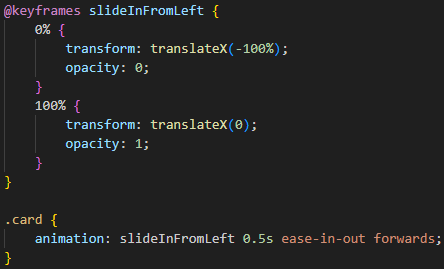


Рис. 15 - Анимация для появления карточек

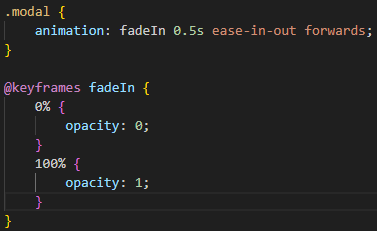


Рис. 16 - Анимация для появления модального окна

1. CSS «**checkout-style.css**» - используется для стилизации основных элементов веб-страницы, таких как шапка, форма заказа, кнопки и текстовые поля. Он задает размеры, цвета, шрифты и отступы, чтобы страница выглядела привлекательно и удобно как на компьютерах, так и на мобильных устройствах.

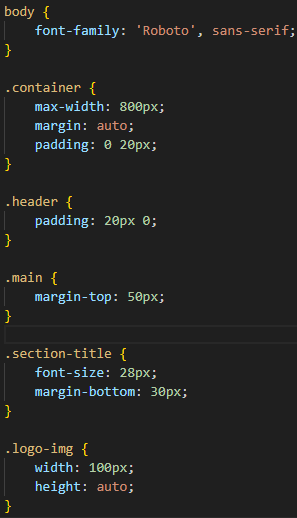


Рис. 17 – Основные стили, контейнер, шапка, основной контент



Рис. 18 - Стили для мобильных устройств

1. JS «**main.js**»

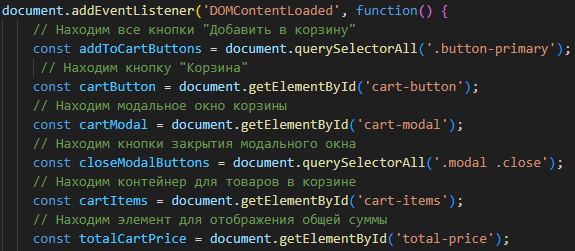


Рис. 19 - Обработчик события загрузки страницы

Скрипт предназначен для работы с элементами на веб-странице, связанными с функциональностью корзины товаров. После загрузки всего HTML-документа скрипт находит различные элементы, такие как кнопки «Добавить в корзину», кнопку «Корзина», модальное окно корзины, кнопки закрытия модального окна, элементы корзины, общую стоимость корзины и счетчик товаров в корзине. Затем скрипт выполняет необходимые действия.



Рис. 20 - Добавление товара в корзину

Данный блок кода отвечает за добавление товара в корзину при нажатии на кнопку «Добавить в корзину». Сначала она определяет, на какую именно кнопку было совершено нажатие. Затем она находит соответствующую карточку товара и извлекает из неё название и цену товара. Далее функция проверяет, есть ли уже такой товар в корзине. Если товар уже присутствует, то обновляется количество этого товара в корзине, увеличивая его счетчик. Если товара еще нет в корзине, создается новый элемент корзины с указанием названия, цены и количества товара. После этого обновляется общая стоимость товаров в корзине, сохраняется содержимое корзины, обновляется счетчик товаров в корзине.

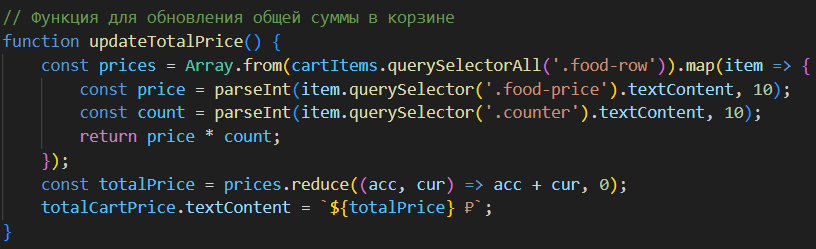


Рис. 21 - Обновление общей суммы в корзине

Функция обновляет общую сумму покупок в корзине. Она находит все строки товаров в корзине и извлекает из них цену каждого товара и количество единиц товара. Затем она вычисляет стоимость каждого товара (цена умножается на количество) и сохраняет эти значения в массив. После этого функция суммирует все стоимости, чтобы получить общую сумму всех товаров в корзине, и отображает эту общую сумму на странице.



Рис. 22 - Изменение количества товаров в корзине и удаление товаров

Код добавляет слушателя событий к элементу корзины покупок. Когда пользователь кликает по элементу в корзине, код определяет, на каком элементе был сделан клик. В зависимости от типа элемента (например, кнопка увеличения или уменьшения количества товара, или кнопка удаления товара), код соответственно изменяет количество товара, удаляет товар из корзины или ничего не делает. После каждого изменения в корзине вызываются функции для обновления общей стоимости корзины, сохранения содержимого корзины, обновления счетчика товаров в корзине.

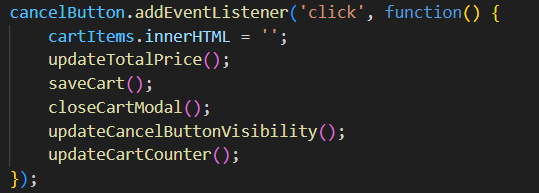


Рис. 23 - Обработчик для кнопки «Очистить корзину»

Этот код добавляет слушателя событий к кнопке "Очистить корзину". Когда пользователь кликает по этой кнопке, код очищает содержимое корзины покупок, обновляет общую стоимость корзины, сохраняет состояние корзины, закрывает модальное окно корзины, обновляет счетчик товаров в корзине.

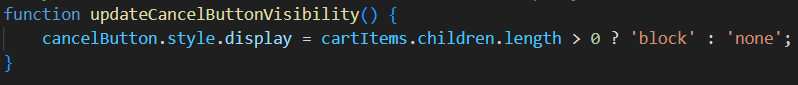


Рис. 24 - Функция для обновления видимости кнопки «Очистить корзину»

Данная функция обновляет видимость кнопки «Очистить корзину» в зависимости от содержимого корзины покупок. Если в корзине есть хотя бы один товар (длина списка дочерних элементов корзины больше нуля), то кнопка «Очистить корзину» будет отображаться, иначе кнопка будет скрыта.

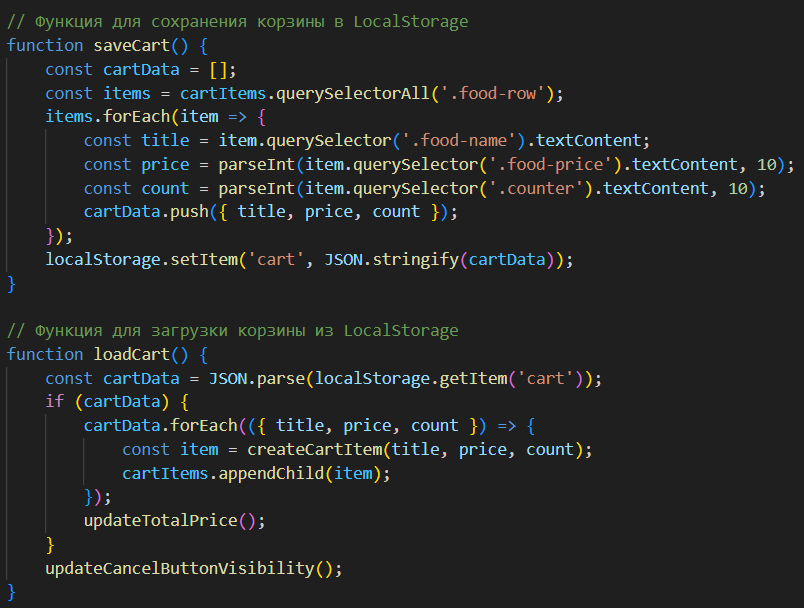


Рис. 25 - Функции для сохранения и загрузки корзины в LocalStorage

Функция «saveCart()» сохраняет все товары из корзины в специальном хранилище браузера. Она берет информацию о каждом товаре (название, цена и количество), упаковывает её в специальный формат, называемый JSON, и сохраняет в этом хранилище.

Функция «loadCart()» делает обратное действие. Она достает все товары из хранилища, которые ранее были сохранены функцией «saveCart()», и показывает их в корзине на странице. Если в корзине был товар до того, как мы закрыли страницу или перезагрузили браузер, эти товары снова появятся в корзине.

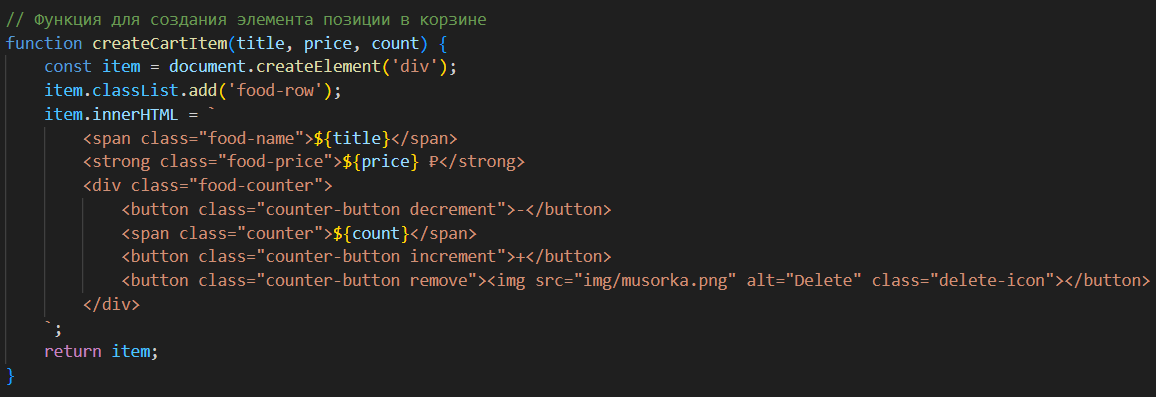


Рис. 26 - Функция для создания элемента позиции в корзине

Функция создает карточку товара для отображения в корзине. Она принимает название товара, его цену и количество. Затем она создает новый блок HTML и добавляет ему класс «food-row». В этот блок добавляются название товара, его цена и кнопки для управления количеством товара. В итоге функция возвращает этот блок, представляющий товар в корзине.

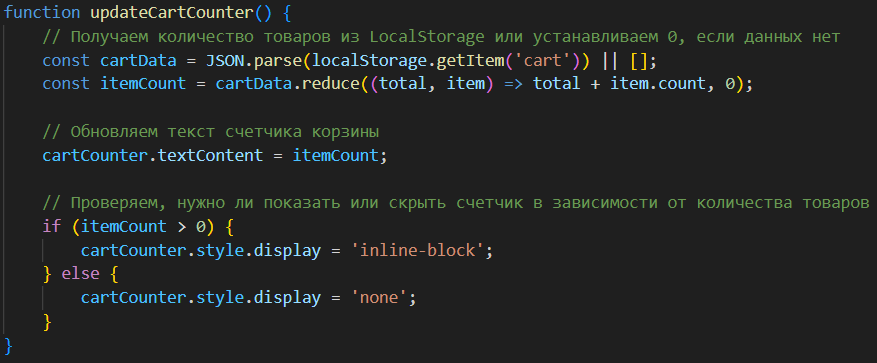


Рис. 27 - Функция для обновления счетчика товаров в корзине

Функция обновляет счетчик товаров в корзине. Она получает данные о товарах из локального хранилища браузера и считает общее количество товаров. Затем функция обновляет текст счетчика корзины на странице, отображая общее количество товаров. В зависимости от того, есть ли товары в корзине или нет, функция показывает или скрывает счетчик на странице.

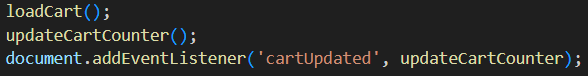


Рис. 28 – Команды для загрузки корзины при обновлении страницы

Данные команды отвечают за управление содержимым корзины покупок на веб-странице. Они загружают содержимое корзины из локального хранилища браузера при загрузке страницы, обновляют счетчик товаров в корзине и обновляют его при каждом изменении содержимого корзины.

1. JS «**checkout.js**»

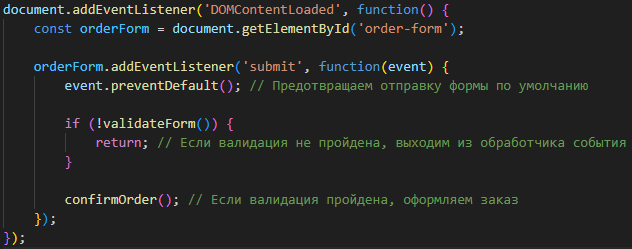


Рис. 29 - Обработчик события загрузки страницы

Код добавляет слушателя событий, который запускается, когда весь HTML-документ полностью загружен. Когда пользователь отправляет форму заказа на странице, этот обработчик событий перехватывает отправку формы по умолчанию, чтобы страница не перезагружалась. Затем он проверяет данные, введенные пользователем, с помощью функции «validateForm()». Если данные проходят проверку, вызывается функция «confirmOrder()», чтобы оформить заказ. Если данные не проходят проверку, ничего не происходит.

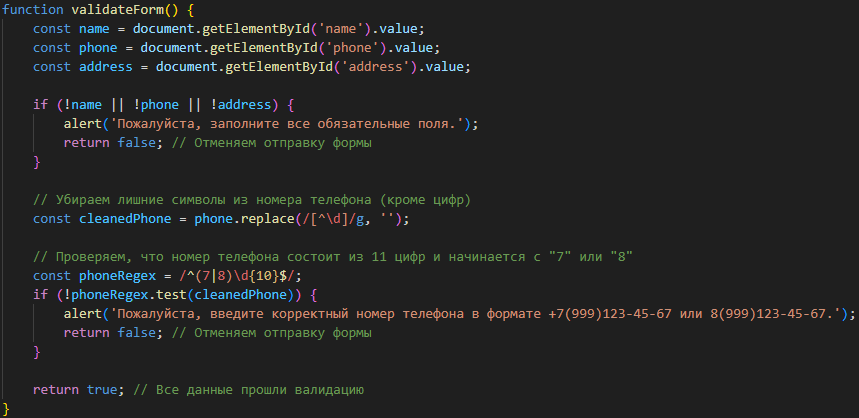


Рис. 30 – Валидация

Функция «validateForm()» проверяет, что пользователь заполнил все обязательные поля в форме заказа и ввел корректный номер телефона. Если какие-то поля не заполнены или номер телефона не соответствует ожидаемому формату, функция выводит предупреждение и предотвращает отправку формы. Если все данные введены правильно, функция позволяет отправить форму.

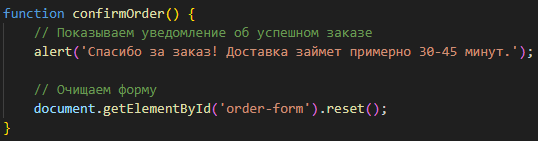


Рис. 31 – Функция подтверждения заказа

Функция «confirmOrder()» используется для подтверждения успешного размещения заказа. Она выводит всплывающее уведомление с сообщением о том, что заказ принят, и информацией о времени доставки. Затем функция очищает все поля ввода формы заказа.

1. JS «**search.js**»



Рис. 32 – Функция поиска

Код делает интерактивный поиск на веб-странице. Когда пользователь вводит текст в поле поиска, код проверяет каждую карточку на странице. Если введенный текст соответствует заголовку карточки, она остается видимой, иначе она скрывается. Таким образом, пользователь может искать карточки по их заголовкам.

**Связь с клиентской частью:**

В нашем проекте для хранения информации о корзине покупок мы решили использовать «localStorage», встроенное хранилище данных веб-браузера. Это позволяет нам сохранять данные на стороне клиента между различными сеансами работы пользователя.

«localStorage» связан с клиентской частью проекта через JS. Когда пользователь добавляет товар в корзину, JS сохраняет информацию о товаре в «localStorage». При просмотре содержимого корзины страница извлекает данные из «localStorage» и отображает их пользователю. Таким образом, «localStorage» играет роль промежуточного хранилища данных между клиентской и серверной частями проекта.

# ЗАКЛЮЧЕНИЕ

***Общие выводы***

В ходе работы была разработана веб-платформа для онлайн-заказа еды, которая включает все основные функциональные элементы, необходимые для эффективного использования как пользователями, так и администраторами системы. В процессе разработки были определены основные требования к системе, выбраны подходящие технологии, созданы дизайн-шаблоны, а также реализована клиентская часть приложения.

***Решение поставленных задач***

Все поставленные задачи были успешно решены:

1. ***Проведен анализ предметной области.***

Было определено, что система предназначена для ресторанов и кафе, предоставляющих услуги по доставке еды. На основе этого анализа были выявлены основные объекты и процессы, участвующие в работе системы, такие как меню, корзина, оформление заказа и подтверждение заказа.

1. ***Определены и выбраны инструменты для разработки.***

В качестве основных инструментов разработки были выбраны:

**«HTML»** и «**CSS»** для создания структуры и стилей веб-страниц.

**«JavaScript»** для реализации интерактивного поведения и динамического контента.

**«Google Fonts (Roboto)»** для улучшения визуальной привлекательности и читабельности текста.

1. ***Разработаны и утверждены прототипы и дизайн-шаблоны.***

Были созданы схемы расположения элементов на страницах и дизайн-шаблоны, включающие стилизацию элементов и их расположение. Это помогло визуализировать и утвердить структуру и внешний вид интерфейса до начала разработки.

1. ***Реализована клиентская часть веб-приложения.***

Клиентская часть приложения была разработана с использованием HTML, CSS и JavaScript.

1. ***Проведено тестирование и внесены необходимые исправления.***

Приложение было протестировано на различных устройствах и браузерах для обеспечения корректной работы и адаптивности. Были внесены исправления для улучшения пользовательского опыта и устранения выявленных ошибок.

***Итоговый результат***

Итогом работы является функциональное веб-приложение для онлайн-заказа еды, которое отвечает современным требованиям по удобству и функциональности. Приложение обладает интуитивно понятным интерфейсом, обеспечивает корректную работу на различных устройствах и платформах, а также включает основные функции для успешного выполнения заказов пользователями.

В дальнейшем можно рассмотреть возможности для расширения функциональности приложения, включая интеграцию с платежными системами, добавление функции отслеживания заказа в реальном времени и улучшение интерфейса администратора для более удобного управления заказами.