В папке "styles" находятся все файлы, связанные с визуальным оформлением страницы, такие как файлы с расширением .css или .sass. Листинги представлены в приложении Г.

3.3 Использование стандартов XML (SVG)

На веб-сайте SVG используется для логотипа в блоке доставки, для изображения социальных сетей в теге footer. Пример кода SVG указан на рисунке 3.3.

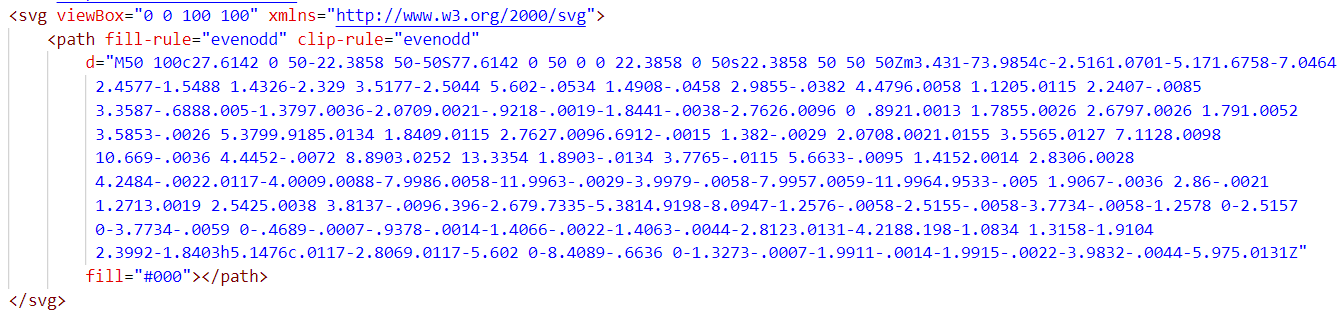


Рисунок 3.3 – Иконка Facebook

XML на веб-сайте используется для подгрузки разных данных на одну страницу при переходе по изображениям-ссылкам в слайдере. Листинги приведены в приложениях Д и Е. Пример использования XML представлен в листинге 3.3.

<info id="11">

        <toimg>АКЦИЯ НА БЛЮДА С МОРЕПРОДУКАТМИ</toimg>

        <pic>/images/11.jpg</pic>

        <description>Каждый понедельник вы можете приобрести любое блюдо с морепродуктами</description>

        <price>со скидкой в 19%</price>

        <code>промокод MORE19</code></info>

Листинг 3.3 – Пример блока информации в XML-документе

3.4 Управление элементами DOM

Управление DOM элементами означает изменение свойств, атрибутов и содержимого HTML-элементов с помощью языка программирования JavaScript. Управление DOM элементами позволяет создавать динамические и интерактивные веб-страницы. Пример использования JavaScript на странице формы представлен в листинге 3.4.

<script>

 const button = document.querySelector('.btn.btn-primary')

 button.addEventListener('click', function(){

  alert("Заказ оформлен успешно!");

  document.location('main.html');})</script>

Листинг 3.4 – Скрипт для страницы с формой

JavaScript был использован на веб-сайте так же для реализации слайдера, корзины и заполнения страницы информацией из XML документа. Листинги приведены в приложении Ж.

3.5 Выводы

На данном этапе была реализована структура веб-сайта на HTML, а также созданы таблицы стилей CSS/Sass, благодаря которым был реализован дизайн веб-сайта. Так же было показано применение SVG и XML для упрощения работы передачи информации и заполнения страниц веб-сайта. Был использован язык программирования JavaScript для создания динамичных и интерактивных элементов веб-сайта.

4 Тестирование веб-сайта

4.1 Адаптивный дизайн веб-сайта

Адаптивный дизайн веб-сайта означает создание веб-сайта, который может адаптироваться под различные размеры экранов устройств, на которых пользователи могут просматривать веб-сайт. Это достигается благодаря использованию гибких элементов и медиа-запросов, которые позволяют определять размер экрана и соответственно изменять расположение и размеры элементов на странице. Адаптивный дизайн обеспечивает оптимальную работу веб-сайта на всех устройствах, включая настольные компьютеры, планшеты и мобильные телефоны, и позволяет сохранять важный контент и функциональность.

Для осуществления на веб-сайте адаптивности, было принято решение использовать Flexbox-вёрстку, которая значительно облегчает работу. Была убрана горизонтальная полоса прокрутки и использованы значения для стилей зависящие от viewport и ширины экрана. Так же были использованы медиа-запросы, пример использования которых представлен на рисунке 4.1.

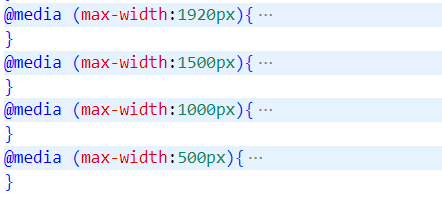


Рисунок 4.1 – Медиа-запросы на главной странице

4.2 Кроссбраузерность веб-сайта

Кроссбраузерность веб-сайта — это возможность корректного отображения и работы веб-страницы на различных веб-браузерах. Проверка кроссбраузерности веб-сайта осуществлялась при помощи Google Chrome, Microsoft Edge, Mozilla Firefox, Opera и Yandex.

В ходе тестирования кроссбраузерности было выяснено, что веб-сайт корректно отображается во всех, взятых для рассмотрения браузерах, поэтому редактирование кода не потребовалось.

4.3 Руководство пользователя

На каждой странице веб-сайта есть шапка, в которой присутствует навигационное меню для перемещения по веб-сайту. Логотип по щелчку мыши ведёт на главную страницу, а ссылки на соответствующие части веб-сайта. На главной станице ссылки «Доставка», «О нас», «Контакты» ведут на блоки с

Заключение

Поставленной целью курсового проекта было создание адаптивного и кроссбраузерного веб-сайта ресторана китайской кухни, котором будет содержать всю полезную и нужную информацию. Кроме главной страницы были созданы страницы «Меню», «Категории», «Заказ» и «Акции».

В ходе выполнения курсового проекта были рассмотрены аналогичные решения, их плюсы и минусы. Были спроектированы прототипы и макеты, на основе которых был создан веб-сайт. Для проектирования прототипов и макетов был использован графический онлайн-редактор Figma. Так же были разработаны логотип, элементы пользовательского интерфейса и спецэффекты.

Были использованы SVG изображения для отображения качественных картинок, которые не будут зависеть от разрешения экрана. Был создан XML-документ для хранения данных. Так же было разработано руководство пользователя для использования веб-сайта.

Была описана реализация поставленных задач для курсового проекта:

1) Анализ существующих решений;

2) Выбор средств для реализации;

3) Разработка макетов и прототипов;

4) Разработка структуры сайта;

5) Наполнение сайта соответствующей информацией;

6) Тестирование веб-сайта;

7) Разработка руководства пользователя.

Был разработан веб-сайт, который соответствует поставленным задачам,

благодаря ранее изученному материалу. Можно заключить, что в ходе выполнения курсового проекта были достигнуты поставленные цели и задачи проекта. Курсовой документ размещён на хостинге Github <https://github.com/yaravaz/Course-project.git>