МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 6-05-0612-01 Программная инженерия

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

Тема «Веб-сайт «Сервис по доставке еды»

**Исполнитель**

студент 1 курса 6 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Жук

подпись, дата

**Руководитель**

старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. В. Барковский

должность, учен. степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ Е. В. Барковский

подпись дата инициалы и фамилия

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра информационных систем и технологий

Утверждаю

И.о. заведующего кафедрой

\_\_\_\_\_\_\_\_\_\_\_ Е.А. Блинова

подпись инициалы и фамилия

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 г.

**ЗАДАНИЕ**

**на курсовое проектирование**

по дисциплине«Компьютерные языки разметки»

Курс 1 Группа: 6 Специальность: 6-05-0612-01 Программная инженерия

Студент: Жук Александр Сергеевич

**1. Тема:** Веб-сайт«Сервис по доставке еды»

**2. Срок выполнения курсового проекта** : с 10 февраля 2025 г. по 05 мая 2025 г.

**3. Технические требования :**

3.1 Прототип веб-сайта должен быть разработан с использованием графических редакторов Figma/Adobe XD/Sketch.

3.2 Для хранения данных должен быть использован XML-формат.

3.3 Разметка содержания сайта должна быть выполнена с применением HTML5 и XML.

3.4 Для описания внешнего вида веб-страниц использовать SCSS и CSS3.

3.5 Веб-сайт должен содержать:

– семантические теги HTML5;

– графические элементы в форме SVG;

– несколько веб-страниц;

– JavaScript для управления элементами DOM.

3.6 Верстка сайта должна быть адаптивной и кроссбраузерной;

3.7 Для тестирования использовать The W3C Markup Validation Service/Git Super Linter

3.8 Проект и пояснения к проекту должны быть размещены на GitHub.

**4. Содержание пояснительной записки**

1. Титульный лист;

2. Задание на курсовое проектирование;

3. Введение;

4. Постановка задачи;

5. Проектирование веб-сайта

6. Реализация структуры веб-сайта

7. Тестирование веб-сайта

8. Заключение

9. Список использованных источников

10. Приложения (полный исходный текст программы разработанного приложения с подробными комментариями)

**5. Форма представления на GitHub выполненного курсового проекта:**

– Теоретическая часть пояснительной записки курсового проекта должна быть представлена в формате MS Word.

– Оформление записки должно быть согласно правилам.

– Листинги представляются в приложении.

#### Календарный план

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта |
| 1 | Задание на курсовое проектирование. Титульный лист | 10.02.2025 |
| 2 | Введение. Постановка задачи | 03.03-09.03.2025 |
| 3 | Проектирование веб-сайта | 10.03-23.03.2025 |
| 4 | Реализация структуры веб-сайта на HTML5 и внешнего оформления на SCSS и СSS3 | 24.03–16.04.2025 |
| 5 | Тестирование веб-сайта | 17.04–24.04.2025 |
| 6 | Заключение | 25.04.2025 |
| 7 | Список использованных литературных источников. Приложения | 26.04.2025 |
| 6 | Подготовка и оформление пояснительной записки курсового проекта | 28.04–04.05.2025 |
| 7 | Сдача на допуск к защите курсового проекта | 05.05.2025 |
| 8 | Защита курсового проекта | 19.05-31.05.2025 |

**5. Дата выдачи задания** «10» февраля 2025 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.В. Барковский

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата и подпись студента)

**Содержание**

|  |  |
| --- | --- |
| Введение | 6 |
| 1 Постановка задачи | 6 |
| 1.1 Обзор аналогичных решений | 6 |
| 1.2 Техническое задание | 8 |
| 1.3 Выбор средств реализации программного продукта | 8 |
| 1.4 Выводы | 9 |
| 2 Разработка макета веб-сайта | 9 |
| 2.1 Выбор способа верстки | 9 |
| 2.2 Выбор стилевого оформления | 9 |
| 2.3 Выбор шрифтового оформления | 9 |
| 2.4 Разработка логотипа | 10 |
| 2.5 Разработка пользовательских элементов | 11 |
| 2.6 Разработка спецэффектов | 11 |
| 2.7 Выводы | 12 |
| 3 Реализация структуры веб-сайта | 12 |
| 3.1 Структура HTML-документа | 12 |
| 3.2 Добавление таблиц стилей SCSS и CSS | 14 |
| 3.3 Использование стандартов XML(SVG) | 15 |
| 3.4 Управление элементами DOM | 16 |
| 3.5 Выводы | 17 |
| 4 Тестирование веб-сайта | 17 |
| 4.1 Адаптивный дизайн веб-сайта | 17 |
| 4.2 Кроссбраузерность веб-сайта | 18 |
| 4.3 Руководство пользователя | 18 |
| 4.4 Выводы | 19 |
| Заключение | 19 |
| Список использованных источников | 20 |
| Приложение А. Прототипы веб-страниц | 20 |
| Приложение Б. Макеты веб-страниц | 22 |

|  |  |
| --- | --- |
| Приложение В. Листинг JavaScript | 24 |

**Введение**

В современном мире сервисы, предлагающие возможность онлайн-покупки товаров (без посещения магазина), являются актуальными и популярными: в 2023 году 45.8% жителей Беларуси покупали товары в Интернете, а к 2027 году, по прогнозам экспертов, покупать товары в интернете будут уже 49.1% белорусов, как указано в [1]. В Беларуси сервисы доставки еды пользуются спросом, о чем свидетельствует присутствие большого количества таких сервисов на рынке: по данным [2],в 2021 году заказать доставку еды в Беларуси можно было в 8 службах доставки.

В связи с тем, что сервисы доставки еды предлагают возможность заказа еды без выхода из дома, веб-сайт является очень важной частью любого такого сервиса, так как именно с помощью веб-сайта клиент может ознакомиться с видами еды, доступными для заказа через сервис, узнать условия и срок доставки, а также оплатить доставку еды с помощью банковской карты. Принятие заказов на доставку еды с помощью телефонной связи имеет ряд недостатков, которых лишены сервисы, оказывающие услуги по доставке еды через веб-сайт: при телефонном разговоре нельзя увидеть сервировку блюда, а также невозможно оплатить еду при ее заказе, что может доставить неудобства клиентам и привести к снижению прибыли.

Целью данного курсового проекта является разработка веб-сайта для сервиса доставки еды с использованием языков разметки HTML, XML, SCSS, языка программирования JavaScript.

К задачам курсового проекта относятся:

1.Обзор аналогичных решений.

2.Составление технического задания.

3.Выбор средств реализации программного продукта.

4.Проектирование страниц веб-сайта.

5.Реализация структуры веб-сайта.

6.Тестирование веб-сайта.

**1 Постановка задачи**

**1.1 Обзор аналогичных решений**

Для обзора решений, аналогичных по тематике веб-сайту, разрабатываемому в рамках курсового проекта, будут использованы два веб-сайта, связанных с доставкой еды: веб-сайт сервиса “Яндекс.Еда” и веб-сайт сервиса “Just-Eat”.

Дизайн аналога веб-сайта курсового проекта (веб-сайта сервиса “Яндекс. Еда” представлен на рисунке 1.1.

Изображение выглядит как текст, снимок экрана, веб-страница, еда

Контент, сгенерированный ИИ, может содержать ошибки. Изображение выглядит как текст, снимок экрана, цветок

Контент, сгенерированный ИИ, может содержать ошибки.

а б

а – десктопная версия, б – мобильная версия

Рисунок 1.1 – Веб-сайт eda.yandex.by

Преимущества данного веб-сайта:

1.Веб-сайт является кроссплатформенным.

2.На веб-сайте предусмотрена возможность фильтрации блод по магазину/ресторану и виду блюда.

3.На веб-сайте возможно изменить алгоритм рекомендации заведений в соответствии с собственными предпочтениями (в меню «Сортировка» доступны варианты «Доверюсь вам», «С высоким рейтингом», «Быстрые»).

4.Имеется возможность указать свой адрес для получения более точной информации о сроках доставки из различных заведений.

5.При просмотре разделы “Магазины” и “Рестораны” визуально разделены.

Недостатки данного веб-сайта:

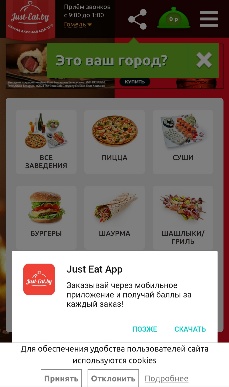
1.В мобильной версии всплывающее окно с информацией о возможности указания адреса закрывает важные элементы интерфейса (кнопку для входа в аккаунт, поисковую строку).

2.На главной странице нет информации о блюдах и продуктах, доступных для заказа.

3.При прокрутке страницы появляется информация о других заведениях, из которых можно заказать еду с помощью сервиса, в результате чего сложно получить доступ к условиям использования и политике конфиденциальности сервиса.

Дизайн аналога веб-сайта курсового проекта (веб-сайта сервиса “Just-Eat”) представлен на рисунке 1.2.

Изображение выглядит как текст, снимок экрана, веб-страница, Веб-сайт

Контент, сгенерированный ИИ, может содержать ошибки. 

а б

а – десктопная версия, б – мобильная версия

Рисунок 1.2 – Веб-сайт just-eat.by

Преимущества данного веб-сайта:

1.Веб-сайт является кроссплатформенным.

2.На веб-сайте предусмотрена возможность фильтрации блод по магазину/ресторану и виду блюда.

3.Имеется возможность указать свой город для получения более точной информации о сроках доставки из различных заведений.

4.На веб-сайте указан номер телефона сервиса как альтернативный вариант для оформления заказа.

Недостатки данного веб-сайта:

1.В мобильной версии при посещении веб-сайта всплывает несколько окон, которые препятствуют использованию веб-сайта до реакции на них.

2.Наличие разноцветного отвлекающего рекламного баннера на фоне веб-сайта, отвлекающих пользователя от оформления заказа.

3.Неиспользование технологий автоматического определения местонахождения пользователя и автоматическое определение местоположения пользователя как г.Гомель в случае, если местоположение пользователя неизвестно.

4.Отсутствие единой стилистики веб-сайта, выполнение различных частей веб-сайта в разных стилях.

**1.2. Техническое задание**

По условиям задания на курсовое проектирование c учетом достоинств и недостатков веб-сайтов, аналогичных разрабатываемому, были определены следующие технические требования к созданию веб-сайта:

1.Веб-сайт должен содержать главную веб-страницу, а также веб-страницы с информацией о заведениях и доступных для заказа в сервисе блюдах.

2.На главной странице веб-сайта должна быть возможность ознакомиться со списком заведений, из которых можно заказать еду с помощью сервиса доставки еды.

3.На главной странице веб-сайта должна быть возможность ознакомиться со списком блюд, которые можно заказать с помощью сервиса.

4.На веб-странице с информацией о заведении должна быть указана оценка заведения пользователями, размещена информация о скидках в заведении и доступных для заказа блюдах.

5.На веб-странице с информацией о заведении должна быть указана оценка заведения пользователями, размещена информация о скидках в заведении и доступных для заказа блюдах, информация о продавце.

5.На веб-странице с информацией о заведении должна быть возможность перехода к блюдам, доступным для заказа из этого заведения.

6. На веб-странице с информацией о блюдах должна быть размещена информация о скидках в заведении, калорийности и составе блюд.

**1.3. Выбор средств реализации программного продукта**

Принимая во внимание результаты обзора аналогичных решений, а также техническое задание, для реализации программного продукта были выбраны следующие средства:

1.Графический редактор Figma для разработки прототипа веб-сайта.

2.Язык разметки XML для хранения данных.

3.Языки разметки HTML5 и XML для разметки содержания веб-сайта.

4.Язык описания внешнего вида документа CSS3 и метаязык SCSS для описания внешнего вида веб-страниц.

5.Язык разметки SVG для описания внешнего вида графических элементов.

6.Язык программирования JavaScript для управления элементами DOM.

7.Сервис The W3C Markup Validation Service для тестирования веб-сайта.

8.Сервис GitHub для размещения курсового проекта и пояснений к нему.

**1.4 Выводы**

В данном разделе был проведен обзор существующих решений, выявлены достоинства и недостатки двух веб-сайтов, аналогичных по тематике проектируемому, определено техническое задание и выбраны средства реализации программного продукта. Выбранные средства реализации веб-сайта позволяют создать веб-сайт с адаптивной и кроссбраузерной версткой, удобный для использования клиентами сервиса по доставке еды.

**2 Разработка макета веб-сайта**

**2.1 Выбор способа верстки**

Согласно техническому заданию верстка разрабатываемого веб-сайта должна быть адаптивной и кросс-браузерной. В связи с этим при выборе технологии верстки было принято решение использовать блочную верстку, так как она является адаптивной и позволяет создавать кроссплатформенные веб-сайты c использованием единого стиля оформления на разных устройствах.

Блочная верстка – стиль верстки, при котором все элементы на веб-странице являются блочными элементами или размещаются внутри них, а их позиционирование определяется значениями, заданными в таблицах стилей SCSS и CSS. При использовании блочной верстки удобно работать с семантическими тегами языка разметки HTML.

**2.2 Выбор стилевого оформления**

При выборе стилевого оформления веб-сайта было принято решение о минималистичном дизайне веб-сайта, что позволит сделать интерфейс удобным для пользователей, а также поможет обеспечить идентичность восприятия как десктопной, так и мобильной версии веб-сайта в связи с тем, что принципы минималистичного дизайна хорошо реализуемы при всех размерах веб-страницы и экрана устройства.

**2.3 Выбор шрифтового оформления**

При выборе шрифтового оформления веб-сайта учитывались удобство чтения используемого шрифта и соответствие шрифта стилевому оформлению веб-сайта, а также поддержка кириллических символов. В результате поиска шрифтов было принято решение использовать шрифт Vollkorn для основного текста веб-страниц в связи с его соответствием стилевому оформлению веб-сайта и наличием насечек, что позволит сделать текст более удобочитаемым, и шрифт Montserrat для заголовков, так как он не содержит насечек, что позволит сделать текст заголовка выделяющимся по сравнению с основным текстом.

**2.4 Разработка логотипа**

При разработке логотипа использовался графический редактор Inkscape,поддерживающий работу с векторной графикой в формате SVG.

В результате процесса разработки был создан логотип, представленный на рисунке 2.1. Логотип состоит из двух элементов: стрелки и изображения огня, выполненных в черном цвете. Огонь на логотипе символизирует преимущество сервиса доставки – быструю доставку свежей еды. Стрелка на логотипе отражает то, что сервис, которому принадлежит веб-сайт, является сервисом доставки еды.

Изображение выглядит как символ

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2.1 – Логотип разрабатываемого веб-сайта

**2.5 Разработка пользовательских элементов**

При разработке пользовательских элементов использовались принципы минималистичного дизайна. Элементы пользовательского интерфейса с учетом применяющейся в разрабатываемом веб-сайте блочной верстки расположены структурированно, их расположение позволяет пользователю быстро получить необходимую информацию об интересующем его кафе, ресторане или блюде. Прототипы веб-страниц размещены в приложении А.

**2.6 Разработка спецэффектов**

В связи с минималистичным дизайном веб-сайта на сайте применяются как спецэффекты:

1.Тени и границы около интерактивных элементов пользовательского интерфейса (кнопок просмотра ресторанов, смены информации о блюдах).

2.Зеленая рамка, появляющаяся вокруг названия блюда, на которое действует акция, при наведении курсора на текст с информацией о скидке на блюдо.

3.Информационный текст о недоступности заведения в сервисе в данный момент, появляющийся с анимацией при попытке перейти по кнопке на страницу заведения, которое в данный момент не представлено в сервисе.

4.Информационный текст об отсутствии других доступных для заказа блюд, появляющийся при нажатии на кнопку “Просмотреть другие блюда” и отсутствии иных блюд, доступных для заказа.

**2.7 Выводы**

В данном разделе был выбран способ верстки веб-сайта, выбрано его шрифтовое и стилевое оформление. В процессе создания прототипов веб-страниц был разработан логотип сайта, а также применяющиеся на нем спецэффекты и пользовательские элементы. Выбранные для разработки веб-сайта технологии и инструменты позволяют реализовать принципы минималистичного дизайна в проектируемых веб-страницах и обеспечить соответствие разработанных веб-страниц их прототипам.

**3 Реализация структуры веб-сайта**

**3.1. Структура HTML-документа**

На созданном веб-сайте внутри тега head размещается тег meta, определяющий кодировку используемых на веб-сайте символов, тег link, позволяющий использовать в документе таблицу стилей CSS из соответствующей папки, полученную путем компиляции препроцессором Sass таблицы стилей SCSS, а также тег title, который определяет название веб-страницы, что помогает пользователю быстрее найти веб-сайт среди открытых вкладок в веб-браузере. Структура тега head на главной веб-странице разработанного веб-сайта приведена в листинге 3.1.

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Сервис по доставке еды</title>

<link rel="stylesheet" href="css/page1.css" />

</head>

Листинг 3.1 – Структура тега head

При верстке сайта использовались семантические теги header, article, aside. Внутри тега header, обычно использующегося для размещения вводного контента (логотипа веб-сайта, имени автора, формы поиска и др.), на созданном веб-сайте был размещен логотип, номер телефона для оформления заказа, форма для ввода города, в котором пользователь хочет оформить заказ, а также название заведения, на веб-странице которого находится пользователь. Структура тега header на главной веб-странице разработанного веб-сайта приведена в листинге 3.2.

<header id="header1">

<img src="images/logo.svg" class="logo" />

<p class="P1">Доставка еды в городе:</p>

<p class="hideo" id="phonehtml"></p>

<form class="form " id="form">

<input class="input" id="inp1" type="text">

</form><aside>

<p class="P2">Режим работы:9:00-22:00</p>

</aside>

</header>

Листинг 3.2 – Структура тега header

В представленном фрагменте кода теги form и input использовались для создания формы, позволяющей пользователю ввести город, в котором он желает оформить заказ, и получить номер телефона, при звонке на который можно оформить заказ с помощью сервиса, а информация о времени работы и назначении сервиса оформлена с помощью тега p. Для отображения логотипа был использован тег img. Атрибуты тегов class служат для подключения стилей через таблицы стилей CSS,а по идентификаторам, заданным с помощью атрибута id, осуществляется доступ к элементам с помощью JavaScript. Семантический тег aside в представленном листинге используется для отображения того, что информация о режиме работы сервиса, размещенная в нем, не является самостоятельной частью веб-страницы и не может быть выделена в отдельную веб-страницу без потери смысла, а также на расположение элемента сбоку страницы, установленное с помощью таблиц стилей CSS и SCSS.

Семантический тег article в созданном веб-сайте был использован для размещения основного контента веб-страниц. Структура тега article на главной веб-странице разработанного веб-сайта приведена в листинге 3.3.

<article class="art1">

<p class="sectionname">Рестораны</p>

<a class="linkto1" href="page2.html"><p class="names" id="name1html"></p></a>

<a class="linkto2" id="l2"><p class="names" id="name2html"></p></a>

<a class="linkto3"><p class="names" id="name3html"></p></a>

<p class="sectionname2">Кафе</p>

<a class="linkto4"><p class="names" id="name4html"></p></a>

<a class="linkto5"><p class="names" id="name5html"></p></a>

<a class="linkto6"><p class="names" id="name6html"></p></a>

<p class="sectionname3">Блюда</p>

<a class="linkto7"><p class="names" id="name7html"></p></a>

<a class="linkto8"><p class="names" id="name8html"></p></a>

<a class="linkto9"><p class="names" id="name9html"></p></a>

</article>

Листинг 3.3 – Структура тега article

В представленном фрагменте кода теги a используются для создания кнопок с возможностью добавления ссылок на другие веб-страницы. С помощью таблиц стилей CSS к кнопкам, размещенным на главной веб-странице с использованием тега a, добавляются фоновые изображения заведений и блюд. Тег p используется для изменения расположения подписей к кнопкам с изображениями заведений и блюд с помощью таблиц стилей CSS и SCSS. Контент, содержащийся внутри тега p (названия заведений и блюд), добавляется на веб-страницу путем переноса информации, хранимой в формате XML, с помощью JavaScript.

На других веб-страницах разработанного веб-сайта теги используются таким же образом, как и в представленных выше листингах.

**3.2. Добавление таблиц стилей SCSS и CSS**

В разработанном веб-сайте таблицы стилей SCSS и CSS используются для визуального оформления, а также установки расположения элементов на веб-странице и добавления к ним анимации.

Миксины в созданном веб-сайте используются для описания свойств повторяющихся элементов (дизайна кнопок на веб-сайте, свойств установки шрифтов, позиционирования элементов). Миксин, использующийся для создания кнопки на веб-странице заведения и аналогичного элемента с информацией о заведении, приведен в листинге 3.4.

@mixin button-style($height, $width, $left, $top, $font-family, $font-size, $border: none) {

    @include absolute-position($left, $top);

    @include font-style($font-family, $font-size);

    display: inline-block;

    height: $height;

    width: $width;

    background-color: $color-primary;

    color: black;

    text-decoration: none;

    border: $border;

}

Листинг 3.4 – Миксин для создания элементов

В представленном миксине свойства, повторяющиеся у обьектов, определены в самом миксине, а различающиеся свойства определяются посредством ввода аргументов при использовании миксина. Для включения других миксинов используется директива @include с указанием передаваемых параметров в форме переменных.

Правила @media были использованы в таблицах стилей SCSS и CSS для создания версий сайта, адаптированных под мобильные устройства и планшеты, а циклы @for использовались для придания одинаковых свойств однотипным элементам. Функции могли использоваться для определения свойств обьекта по полученным аргументам, но в большинстве случаев вместо функций и шаблонов в коде использовались миксины, так как они сочетают преимущества шаблона и функции, позволяя работать с передаваемыми аргументами и автоматически подставлять значения при использовании миксина. Использование функции, правила @media,условного оператора и оператора цикла в коде приведено в листинге 3.5.

@function getlinktomobilesize($i){

  @if($i >=1 and $i <=9){@return 1.6vh;}

}

@media screen and (max-width: 1028px) {

  @for $i from 1 through 9 {

    .linkto#{$i} { font-size: getlinktomobilesize($i) }

  }

Листинг 3.5 – Использование различных конструкций и операторов в таблице стилей SCSS

В представленном фрагменте кода функция используется для недопущения применения стиля к элементам, для которых этого не было предусмотрено, а также для указания на то, что это свойство предназначено для версии сайта, адаптированной под мобильные устройства. Правило @media используется для применения стиля только при просмотре веб-сайта с устройства с максимальной шириной до 1028 пикселей, а цикл используется для присвоения стиля элементам с однотипными селекторами.

**3.3. Использование стандартов XML (SVG)**

Стандарты XML при разработке веб-сайта использовался для хранения информации о заведениях и блюдах. Структура XML-документа, в котором хранится информация о блюдах и заведениях для добавления текста к кнопкам на главной странице, представлена в листинге 3.6.

<info>

<phone id="phonexml">Номер телефона для заказа:+375111111111</phone>

<rest id="name1xml">

KFC

</rest>

<rest id="name2xml">

Mak.by

</rest>

<rest id="name3xml">

Papa Doner

</rest>

<cafe id="name4xml">

Bubble Tea

</cafe>

<cafe id="name5xml">

Klaus Cafe

</cafe>

<cafe id="name6xml">

Драники & Кава

</cafe>

<food id="name7xml">

Харчо

</food>

<food id="name8xml">

Эклер малиновый

</food>

<food id="name9xml">

Сэндвич с курочкой

</food>

</info>

Листинг 3.6 – Структура XML-документа

В представленном фрагменте кода тег info – заглавный тег, включающий в себя все теги XML-документа,а теги phone,rest,cafe и food обозначают тип обьекта, название которого приведено внутри тега.

Доступ к элементам XML для включения их содержания в элементы HTML-документа осуществлялся по их идентификатору с помощью языка программирования JavaScript. Код, осуществляющий доступ к элементам XML, приведенным в листинге 3.6, и их включение в объектную модель (DOM) главной веб-страницы, приведен в листинге В.1.

Для работы с векторной графикой в созданном веб-сайте был использован формат SVG. Создание SVG-изображений для использования на веб-страницах производилось с помощью графического редактора Inkscape, а использование их производилось с помощью тега img языка разметки HTML.

**3.4 Управление элементами DOM**

В разработанном веб-сайте для управления элементами объектной модели документа использовался язык программирования JavaScript. С помощью языка программирования JavaScript были реализованы следующие спецэффекты на веб-сайте:

1. Отображение введенного в форму выбора города на главной странице текста и номера телефона, по которому можно оформить заказ в сервисе, при нажатии клавиши Enter после ввода текста. Код, с помощью которого реализуется данный спецэффект, представлен в листинге В.2.
2. Отображение информации о том, что заведение скоро будет доступно в сервисе, при нажатии на кнопку заведения, еще не представленного в сервисе, на главной странице. Код, с помощью которого реализуется данный спецэффект, представлен в листинге В.3.
3. Появление текста “Других блюд пока нет в наличии! Но вы можете ознакомиться с блюдами, имеющимися в наличии, и действующими акциями.” при попытке ознакомиться с другими доступными блюдами на веб-странице с информацией о блюдах. Код, с помощью которого реализуется данный спецэффект, представлен в листинге В.4.
4. Появление зеленой рамки вокруг названия блюда, на которое действует акция, и изменение текста, сообщающего об акции, при наведении курсора на информацию об акции, размещенную на веб-странице с информацией о блюдах. Код, с помощью которого реализуется данный спецэффект, представлен в листинге В.5.

**3.5 Выводы**

При разработке веб-сайта использовались язык разметки HTML для создания структуры HTML-документа, таблицы стилей CSS и SCSS для изменения визуального оформления веб-сайта, стандарт XML для хранения данных, стандарт SVG для работы с векторной графикой, язык программирования JavaScript для использования данных, хранимых в формате XML, и создания спецэффектов на веб-сайте. Данные технологии позволяют обеспечить соответствие функциональности сайта техническому заданию и создать сайт, удобный для использования пользователями и дальнейшей разработки, добавления нового функционала.

**4 Тестирование веб-сайта**

* 1. **Адаптивный дизайн веб-сайта**

Для тестирования веб-сайта на адаптивность использовались веб-браузеры Microsoft Edge и Mozilla Firefox, так как они используют разные движки (Blink и Gecko),что позволяет полноценно оценить адаптивность и доступность веб-сайта для большинства пользователей.

По результатам тестирования, проведенного в 2 веб-браузерах для 3 типов устройств (смартфон Iphone 14 Pro,планшет IPad Air,ноутбук Asus FX505DT), некорректной работы веб-страниц выявлено не было, веб-сайт сохранял свою функциональность на выбранных устройствах в обоих веб-браузерах.

* 1. **Кроссбраузерность веб-сайта**

Для тестирования веб-сайта на кроссбраузерность использовались веб-браузеры Microsoft Edge и Mozilla Firefox, так как они используют разные движки (Blink и Gecko) что позволяет полноценно оценить кроссбраузерность веб-сайта и его доступность для большинства пользователей. Движок Blink используется всеми веб-браузерами на основе Chromium,поэтому тестирование работоспособности веб-сайта на этом движке позволяет определить работоспособность веб-сайта в большом количестве веб-браузеров, включая браузеры Google Chrome и Opera.

При тестировании кроссбраузерности веб-сайта непредвиденного поведения веб-страниц (изменение шрифтов, некорректная работа анимации) выявлено не было, сайт не отличался по функциональности в обоих браузерах, использовавшихся для тестирования веб-сайта на кроссбразуерность. Так как при разработке веб-сайта не использовалось программных конструкций языка разметки HTML или таблиц стилей CSS, не входящих в актуальные стандарты, веб-сайт должен корректно работать во всех веб-браузерах, поддерживающих стандарты HTML5,CSS3 и JavaScript.

* 1. **Руководство пользователя**

При посещении главной веб-страницы пользователь веб-сайта может совершить 3 действия:

1.Ввести город, где пользователь планирует оформлять заказ, в соответствующую форму. После завершения ввода поле для ввода города исчезнет и будет заменено на название города, введенное пользователем, а также на веб-странице появится информация о номере телефона для заказа блюд с помощью сервиса по доставке еды.

2.Нажать на кнопку с названием заведения, еще не представленного в сервисе. После этого действия кнопка заменится на информацию о невозможности заказа блюда из выбранного заведения через сервис.

3.Нажать на кнопку с названием заведения, представленного в сервисе. После этого действия произойдет переход на веб-страницу с информацией о заведении.

Также на главной веб-странице пользователь может ознакомиться с логотипом и информацией о времени работы сервиса.

При посещении веб-страницы с информацией о заведении пользователь веб-сайта может совершить 2 действия:

1.Нажать на логотип сервиса по доставке еды в левом верхнем углу веб-страницы. После этого произойдет переход на главную веб-страницу.

2.Нажать на кнопку “Просмотреть блюда” в нижней части экрана. После этого произойдет переход на веб-страницу с информацией о блюдах.

Также на веб-странице с информацией заведения пользователь может ознакомиться с фотографией заведения, информацией об отзывах, акциях, наличии новых блюд и организации, оказывающей услуги.

При посещении веб-страницы с информацией о блюдах пользователь веб-сайта может совершить 3 действия:

1.Ознакомиться с информацией об блюде, на которое в данный момент действует акция, и его ценой без акции, наведя курсор мыши на зеленый текст с информацией об акции.

2. Нажать на кнопку “Просмотреть блюда” в нижней части экрана. После этого текст кнопки заменится на информацию об отсутствии других блюд в данный момент.

3.Нажать на кнопку c изображением стрелки вверху экрана. После этого произойдет переход на веб-страницу с информацией о заведении, осуществляющем приготовление блюд.

Также на веб-странице с информацией о блюдах пользователь может ознакомиться с названиями и изображениями блюд, доступных к заказу в сервисе, информацией об их составе и энергетической ценности, названием заведения, осуществляющего приготовление блюд.

* 1. **Выводы**

В процессе тестирования веб-сайта использовались 2 браузера (Google Chrome и Mozilla Firefox), работающих на разных движках, что позволило качественно оценить адаптивность и кроссбраузерность веб-сайта. В результате тестирования не было выявлено непредвиденного поведения веб-сайта, что позволяет утверждать о корректной работе веб-страниц на различных устройствах и в различных браузерах.

**Заключение**

В процессе работы над курсовым проектом были выполнены следующие задачи:

1.Произведен обзор существующих решений, определены их преимущества и недостатки.

2.Составлено техническое задание, определены технические требования к созданию веб-сайта.

3.Выбраны средства реализации программного продукта (графический редактор Figma, языки разметки HTML, XML, SVG, таблицы стилей CSS и SCSS, сервисы The W3C Markup Validation Service и GitHub).

4.Спроектированы веб-страницы, созданы их макеты и прототипы.

5.Реализована структура веб-сайта, состоящая из 3 веб-страниц.

6.Проведено тестирование веб-сайта на кроссбраузерность и адаптивность.

При работе над проектом технологии, встречающиеся в учебном курсе, позволили полностью реализовать необходимую функциональность веб-сайта, технологии, не описанные в учебном курсе, не использовались.

Результат работы над курсовым проектом – кроссбраузерный и адаптивный веб-сайт для сервиса по доставке еды, размещенный на сервисе GitHub по адресу: https://github.com/00ux/KYAR\_Dostavka\_Edy/ и позволяющий пользователю получить всю необходимую информацию для выбора и заказа блюд.

Среди направлений развития проекта можно выделить:

1.Динамическое создание контента о блюдах и заведениях в зависимости от источника, с которого пользователь перешел на страницу.

2.Реализация интерфейса для заказа еды через сервис по доставке еды с помощью веб-сайта.

3.Использование города, введенного пользователем в форму на главной веб-странице, для персонализации контента на веб-сайте.

**Список использованных источников**

1.Банк БелВЭБ [Электронный ресурс] – 2023. – Режим доступа: <https://www.belveb.by/blog/business-1068-s/e-commerce-perspektivy-otrasli-v-2023-godu-39861-p/> – Дата доступа: 04.03.2025.

2. Аргументы и факты в Беларуси [Электронный ресурс] – 2021. – № 11. Режим доступа: <https://aif.by/social/priyatnogo_appetita_kak_v_belarusi_razvita_sistema_dostavki_edy> – Дата доступа: 04.03.2025.

Приложение А

**Прототипы веб-страниц**

Изображение выглядит как текст, диаграмма, оригами, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок А.1 – Прототип главной веб-страницы

Изображение выглядит как текст, линия, диаграмма, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок А.2 – Прототип веб-страницы с информацией о заведении

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок А.3 – Прототип веб-страницы с информацией о доступных блюдах

Приложение Б

**Макеты веб-страниц**

**Изображение выглядит как текст, еда, блюдо, Быстрое питание

Контент, сгенерированный ИИ, может содержать ошибки.**

Рисунок Б.1 –Макет главной веб-страницы

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок Б.2 –Макет веб-страницы с информацией о заведении

Изображение выглядит как текст, меню, снимок экрана, фрукт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок Б.3 –Макет веб-страницы с информацией о доступных блюдах

Приложение В

**Листинг JavaScript**

document.getElementById("phonehtml").innerHTML =

document.getElementById("phonexml").firstChild.nodeValue;

document.getElementById("name1html").innerHTML =

document.getElementById("name1xml").firstChild.nodeValue;

document.getElementById("name2html").innerHTML =

document.getElementById("name2xml").firstChild.nodeValue;

document.getElementById("name3html").innerHTML =

document.getElementById("name3xml").firstChild.nodeValue;

document.getElementById("name4html").innerHTML =

document.getElementById("name4xml").firstChild.nodeValue;

document.getElementById("name5html").innerHTML =

document.getElementById("name5xml").firstChild.nodeValue;

document.getElementById("name6html").innerHTML =

document.getElementById("name6xml").firstChild.nodeValue;

document.getElementById("name7html").innerHTML =

document.getElementById("name7xml").firstChild.nodeValue;

document.getElementById("name8html").innerHTML =

document.getElementById("name8xml").firstChild.nodeValue;

document.getElementById("name9html").innerHTML =

document.getElementById("name9xml").firstChild.nodeValue;

Листинг В.1 – Использование JavaScript для работы с данными в формате XML

document.getElementById('form').addEventListener('submit', function (event) {

event.preventDefault();

const inputField = document.getElementById('inp1');

const inputValue = inputField.value.trim();

if (inputValue) {

const newParagraph = document.createElement('p');

newParagraph.textContent = inputValue;

newParagraph.classList.add('fornewpgph');

document.getElementById('phonehtml').classList.remove("hideo");

inputField.replaceWith(newParagraph);

}

});

Листинг В.2 – Использование JavaScript для отображения информации при заполнении формы

document.querySelectorAll('.linkto1').forEach(element => {

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

const newElement = document.createElement('span');

newElement.className = 'fornewpph1';

newElement.textContent = "Заведение скоро будет в нашем сервисе!"

this.style.opacity = '0';

this.addEventListener('transitionend', () => {

this.replaceWith(newElement);

//

requestAnimationFrame(() => {

newElement.style.opacity = '1';

});

}, { once: true });

});

});

Листинг В.3 – Использование JavaScript для отображения информации при нажатии на элемент

document.getElementById('refresh').addEventListener('click', function (e) {

e.preventDefault();

const element = e.currentTarget;

element.textContent = "Других блюд пока нет в наличии! Но вы можете ознакомиться с блюдами, имеющимися в наличии, и действующими акциями.";

element.classList.add("fornewtext");

element.style.pointerEvents = 'none';

});

Листинг В.4 – Использование JavaScript для изменения информации при нажатии на элемент

document.getElementById("sale2").addEventListener('mouseover', (e) => {

document.getElementById("sale2").innerHTML = 'Цена без скидки:10 руб.90 коп.';

document.getElementById("name2html").style.border = "3px solid green";

});

document.getElementById("sale2").addEventListener('mouseout', (e) => {

document.getElementById("sale2").innerHTML = 'Скидка 1 руб. на Гриль Бургер';

document.getElementById("name2html").style.border = "0";

});

Листинг В.5 – Использование JavaScript для изменения информации и применения эффекта к другому элементу при нажатии на элемент