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

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 6-05-0611-01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

Тема: Веб-сайт «Пиццерия»

**Исполнитель**

студент 1 курса 1 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т. А. Туцкая

подпись, дата

**Руководитель**

Ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. В. Сазонова

должность, учен. степень, ученое звание подпись, дата

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

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

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ Д. В. Сазонова

подпись дата инициалы и фамилия

Содержание

[1. Обзор технических методов. 6](file:///D:\yes\2sem\course-1\).docx#_Toc134695540)

[1.1. Обзор аналогичных решений 6](file:///D:\yes\2sem\course-1\).docx#_Toc134695541)

[1.2 Техническое задание 16](file:///D:\yes\2sem\course-1\).docx#_Toc134695542)

[1.3. Постановка задач программного продукта 17](file:///D:\yes\2sem\course-1\).docx#_Toc134695543)

[1.4. Выбор средств реализации программного продукта 17](file:///D:\yes\2sem\course-1\).docx#_Toc134695544)

[1.5. Вывод 18](file:///D:\yes\2sem\course-1\).docx#_Toc134695545)

[2. Проектирование страниц веб-сайта. 18](file:///D:\yes\2sem\course-1\).docx#_Toc134695546)

[2.1. Выбор способа верстки 18](file:///D:\yes\2sem\course-1\).docx#_Toc134695547)

[2.2. Выбор стилевого оформления 19](file:///D:\yes\2sem\course-1\).docx#_Toc134695548)

[2.3 Выбор шрифтового оформления 19](file:///D:\yes\2sem\course-1\).docx#_Toc134695549)

[2.4. Разработка пользовательских элементов 19](file:///D:\yes\2sem\course-1\).docx#_Toc134695550)

[2.5. Разработка спецэффектов 21](file:///D:\yes\2sem\course-1\).docx#_Toc134695551)

[2.6. Выводы 21](file:///D:\yes\2sem\course-1\).docx#_Toc134695552)

[3. Реализация структуры веб-сайта. 21](file:///D:\yes\2sem\course-1\).docx#_Toc134695553)

[3.1. Структура HTML-документа 21](file:///D:\yes\2sem\course-1\).docx#_Toc134695554)

[3.2. Добавление таблиц стилей SCSS и CSS 22](file:///D:\yes\2sem\course-1\).docx#_Toc134695555)

[3.3. Использование стандартов XML (SVG) 24](file:///D:\yes\2sem\course-1\).docx#_Toc134695556)

[3.4. Выводы 26](file:///D:\yes\2sem\course-1\).docx#_Toc134695557)

[4. Тестирование веб-сайта. 26](file:///D:\yes\2sem\course-1\).docx#_Toc134695558)

[4.1. Адаптивный дизайн веб-сайта 26](file:///D:\yes\2sem\course-1\).docx#_Toc134695559)

[4.2.Кроссбраузерность веб-сайта 28](file:///D:\yes\2sem\course-1\).docx#_Toc134695560)

[4.3. Руководство пользователя 31](file:///D:\yes\2sem\course-1\).docx#_Toc134695561)

[4.4. Выводы 31](file:///D:\yes\2sem\course-1\).docx#_Toc134695562)

[5. Заключение 32](file:///D:\yes\2sem\course-1\).docx#_Toc134695563)

[6. Список использованных литературных источников 33](file:///D:\yes\2sem\course-1\).docx#_Toc134695564)

# Введение

В нашем современном информационном обществе Интернет стал неотъемлемой частью нашей повседневной жизни. Миллиарды людей всего мира используют Интернет для общения, поиска информации, развлечений и многого другого. В этом контексте создание и наличие собственного сайта становится не просто полезным, а часто даже необходимым условием для успешного существования в онлайн-мире.

Один из ключевых аспектов, по которому сайты являются необходимостью, заключается в их роли в представлении и продвижении бизнеса. Веб-сайт позволяет компаниям, предпринимателям и фрилансерам эффективно представлять свои товары и услуги онлайн. Компании могут использовать свой сайт для создания уникального бренда, установления контакта с клиентами и предоставления им информации о своих продуктах или услугах. Отсутствие веб-присутствия может означать упущение возможностей привлечения новых клиентов и уступку конкурентам, которые активно используют преимущества Интернета.

Цель курсовой работы: разработать веб-сайт для пиццерии с использованием HTML5, а также с применением SCSS/CSS3 и XML.

Задачи курсовой работы:

* проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта;
* разработать оригинальный макет сайта;
* наполнить сайт информацией по теме;
* разработать структуру веб-сайта;
* создать контент сайта на XML;
* протестировать веб-сайт;
* разработать руководство пользователя.

Целевой аудиторией являются пользователи, которые заинтересованы в покупке пиццы, а также в поиске новых вкусовых сочетаний.

# Постановка задачи

1.1 Обзор аналогичных решений

Перед выполнением курсового проекта следует провести анализ веб-сайтов такой же тематики. Анализ помогает нам выяснить, что сейчас актуально и в приоритете у пользователей, а также изучить разные подходы передачи информации на веб-сайте, стили и грамотный подбор палитры цветов.

1.1.1 Аналог «Domino’s Pizza»

На рисунке 1.1 предоставлен фрагмент меню сайта.

|  |
| --- |
|  |

Рисунок 1.1 – Фрагмент меню сайта «Domino’s Pizza»

На данном сайте пиццы представлены визуально привлекательным образом, с видимыми начинками и аппетитной корочкой. Фоновый цвет, который используется, является светлым, что создает чистое и свежее впечатление. Это позволяет пиццам стать главным акцентом изображения. Размеры пицц также ясно указаны, что помогает посетителям сайта быстро выбрать свой вариант.

В целом, дизайн создает приятное визуальное впечатление и эффективно передает информацию о разнообразии пицц, доступных на веб-сайте.

С данного сайта также можно рассмотреть модальное окно, которое предоставляет более подробную информацию о выборе характеристик пиццы. Модальное окно представлено на рисунке 1.2.

Здесь можно отметить гармоничный дизайн и удачное расположение элементов модального окна. Оно имеет чистый и современный вид, сочетающийся с общим стилем веб-сайта; отображается в центре экрана, привлекая внимание пользователя. Расположение текстовых и графических элементов в модальном окне также удачно выполнено, что обеспечивает простоту восприятия информации. В целом, дизайн и расположение элементов в данном модальном окне создают гармоничную и удобную пользовательскую интерфейсную визуализацию.

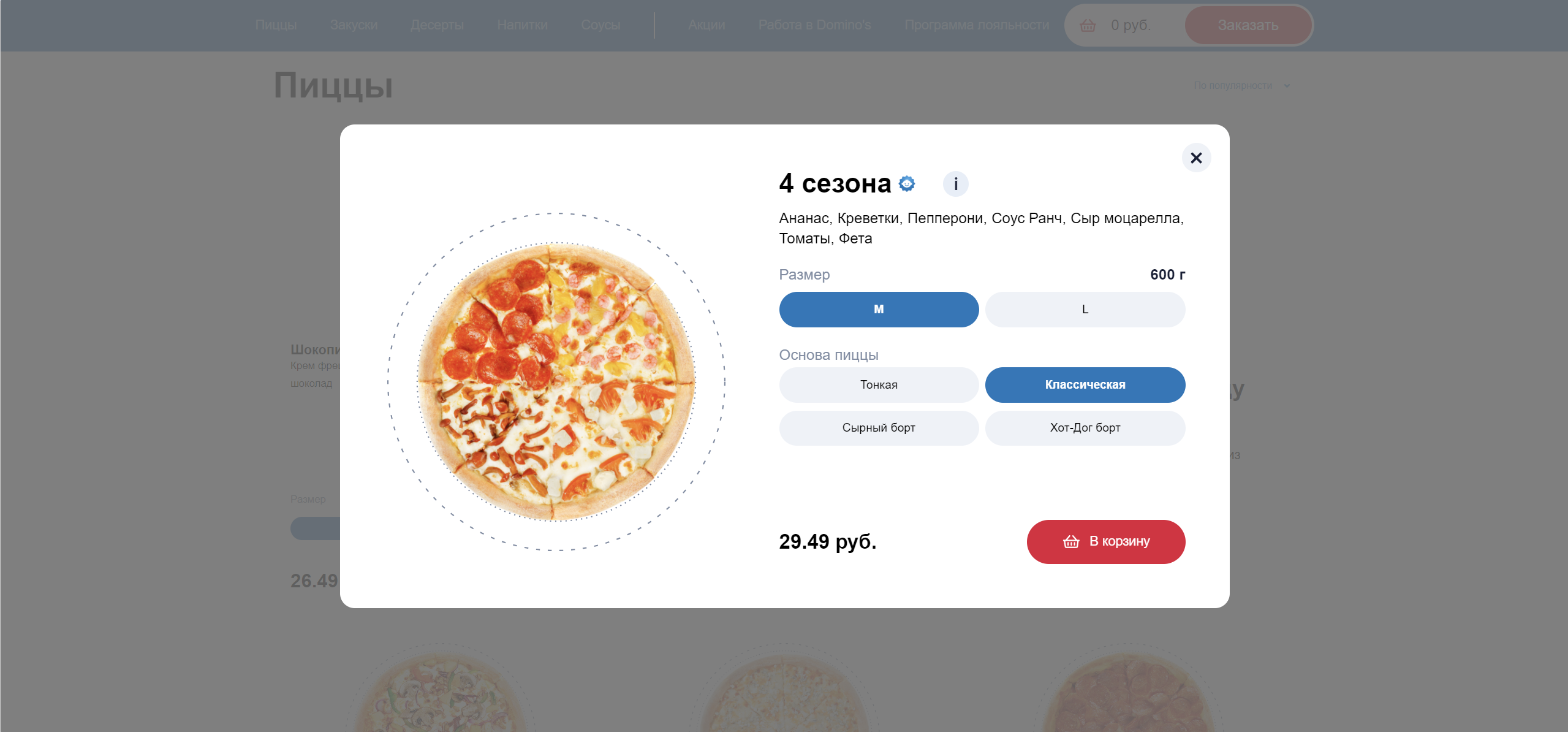


Рисунок 1.2 – Модальное окно с более подробной информацией о пицце сайта «Domino’s Pizza»

Недостатки не были обнаружены.

Здесь можно сделать вывод, что на сайте элементы меню должны быть грамотно расположены друг относительно друга, поддержание минималистического стиля.

1.1.2 Аналог

1.2 Техническое задание

Была поставлена задача реализовать полноценный многостраничный сайт на тему «Пиццерия». На сайте должна быть предоставлена информация о ресторане, краткое описание и отдельная страница, на которой будет расположено меню со всеми товарами и их кратким описанием. Вся информация должна являться актуальной.

Основными задачами сайта являются:

1. Предоставление общей информации о пиццерии;
2. Предоставление информации о меню;
3. Предоставление подробной информации о каждой секции из меню.

На каждой странице веб-сайта будет располагаться навигационное меню, с помощью которого пользователю будет удобно перемещаться по сайту. Также снизу будет расположена контактная информация ресторана и его соц. сети.

Также для привлечения внимания целевой аудитории главная задача сделать удобный и стильный интерфейс сайта, грамотно расположить все элементы на страницах.

Все страницы веб-сайта должны быть кроссбраузерными и адаптивными для мобильной версии, планшетного устройства и ноутбука, чтобы пользователь имел возможность зайти на сайт с любого устройства.

1.3. Выбор средств реализации программного продукта

Курсовая работа выполняется в редакторе кода Visual Studio Code. Visual Studio Code — это упрощенный, но мощный редактор исходного кода, который работает на компьютере и доступен для Windows, macOS и Linux. Имеет встроенную поддержку JavaScript, TypeScript и Node.js, а также обширную экосистему расширений для других языков и сред выполнения.

При создании сайта и его функционала будут использованы HTML, CSS, SASS, JavaScript и XML.

HTML (HyperText Markup Language) — это стандартизированный язык разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML-документ от сервера по протоколам HTTP или HTTPS, или открывают его с локального диска, и затем интерпретируют код, преобразуя его в интерфейс, который отображается на экране монитора.

CSS (Cascading Style Sheets)— это язык, который используется для описания внешнего вида и форматирования документа, написанного на языке разметки HTML. CSS позволяет веб-разработчикам создавать стиль и дизайн веб-страниц, контролируя макет, цвета, шрифты и многое другое.

HTML используется для создания структуры содержимого веб-страницы, в то время как CSS применяется для стилизации и оформления этого содержимого.

JavaScript — это мощный язык программирования, который используется для создания динамического и интерактивного контента на веб-страницах. Он позволяет добавлять сложные функции, такие как анимация, обработка событий, управление мультимедиа и многое другое. JavaScript работает в браузере пользователя и является неотъемлемой частью веб-разработки, дополняя HTML и CSS.

SCSS (Sassy CSS) — это синтаксис для языка препроцессора SASS, облегчает написание и поддержку стилей, предоставляя такие функции, как переменные, вложенность, миксины, функции и операторы. Он использует синтаксис, похожий на обычный CSS, что делает его легко читаемым и понятным.

XML (eXtensible Markup Language) — это расширяемый язык разметки, который используется для описания, хранения и передачи структурированных данных. Он позволяет создавать свои собственные теги, обеспечивая тем самым гибкость в представлении данных. XML широко используется для обмена данными между приложениями и системами, включая веб-сервисы, базы данных и многие другие.

1.4. Вывод

В данном разделе были рассмотрены аналоги веб-сайтов на выбранную тему, выявлены их плюсы и минусы для предотвращения недочётов и ошибок в реализации веб-сайта. Были выбраны средства разработки программного продукта, поставлены конкретные цели и определены технические задачи.

# 2. Проектирование страниц веб-сайта

2.1 Выбор способа вёрстки

Требования курсового проекта включают в себя кроссбраузерность и адаптивность. По этой причине была выбрана flex-вёрстка.

Flex-вёрстка (Flexbox) в HTML является одним из методов организации и расположения элементов на веб-странице. Она основана на концепции гибкого контейнера и гибких элементов, которые могут быть упорядочены и выравниваться внутри контейнера.

Flexbox предоставляет набор свойств CSS, которые позволяют управлять размещением элементов в строку или столбец и распределять доступное пространство между ними. Он предоставляет более простую и эффективную альтернативу традиционным методам верстки с использованием float и position.

2.2 Выбор стилевого оформления

Выбор стилевого оформления является важным этапом разработки проекта по нескольким причинам:

1. Первое впечатление. Дизайн сайта играет ключевую роль в формировании первого впечатления у посетителей. Когда пользователь открывает веб-сайт, он сразу же оценивает его внешний вид и визуальную привлекательность. Чистый, современный и профессиональный дизайн может вызвать положительные эмоции и улучшить восприятие бренда. С другой стороны, плохо спроектированный или неэстетичный дизайн может вызвать негативные ассоциации и оттолкнуть посетителей.
2. Удобство использования. Хорошо продуманный дизайн сайта обеспечивает удобство использования и улучшает пользовательский опыт. Правильное размещение элементов, интуитивная навигация, понятные и читаемые шрифты, доступность функциональности – всё это делает сайт более удобным для посетителей. Когда пользователи могут легко находить информацию и выполнять нужные действия без лишних усилий, они чувствуют себя комфортно и вероятность их возвращения на сайт возрастает.
3. Соответствие целевой аудитории. Стилевое оформление сайта должно соответствовать предпочтениям и ожиданиям целевой аудитории. Изучение характеристик и предпочтений целевой аудитории поможет определить подходящие цветовые схемы, шрифты, компоновку и визуальные элементы.

Для акцентирования внимания клиента на товаре при оформлении веб-сайта был выбран минимализм.

Минимализм как стиль оформления интернет-магазина имеет свои преимущества и может быть эффективным выбором по следующим причинам:

1. Фокус на товарах. Минималистический дизайн помогает сосредоточить внимание клиентов исключительно на товарах. Отсутствие избыточных декоративных элементов делает страницы более простыми и чистыми, что позволяет клиентам быстро оценить предлагаемые товары.
2. Стиль и элегантность. Минималистичный дизайн может создать ощущение стиля и элегантности, что может быть важным для привлечения пользователей и установления доверия к бренду.
3. Быстрая загрузка страниц. Из-за отсутствия избыточных декоративных элементов и сложных графических элементов, минималистический дизайн обычно приводит к быстрой загрузке страниц. Это важно для удержания внимания клиентов и снижения отказов.
4. Простота обновления и поддержки. Благодаря своей простоте и чистоте, минималистический дизайн обычно легче обновлять и поддерживать в сравнении с более сложными стилями оформления.

Таким образом, выбор стилевого оформления сайта играет ключевую роль в привлечении пользователей, создании удобства использования, установлении брендовой идентичности и отражении целей бренда. Это не только важно для первого впечатления, но и для удержания посетителей и улучшения пользовательского опыта.

2.3 Выбор шрифтового оформления

Для оформления текстового содержания сайта был выбран шрифт Epilogue.

Шрифт Epilogue – это современный и элегантный шрифт, который может быть превосходным выбором для оформления сайта по нескольким причинам:

1. Читаемость. Epilogue имеет чёткие и чистые формы символов, что делает его легко читаемым как на экране, так и на печатной продукции. Это особенно важно для веб-сайтов, где посетители ожидают быстрого доступа к информации.
2. Стиль. Этот шрифт имеет современный и стильный вид, что помогает создать современный облик сайта. Он хорошо сочетается с различными дизайнерскими стилями и может быть использован как для заголовков, так и для текстового контента.
3. Вариативность. Epilogue предлагает несколько начертаний, от обычного до полужирного и курсивного, что даёт возможность создавать разнообразные дизайнерские решения и выделять ключевые элементы на странице.

В целом, **Epilogue** представляет собой универсальный и гибкий шрифт, который может быть хорошим выбором для веб-дизайна. Его чистые линии и разнообразие весов делают его легко читаемым и адаптируемым к различным контекстам.

Шрифт был взят с сайта «Google Fonts» и подключён через конструкцию @font-face.

2.4 Разработка логотипа

Логотип – это визуальный знак, представляющий бренд или компанию, помогающий создать узнаваемость и запоминаемость.

Хороший логотип должен быть простым, запоминающимся, уникальным и легко узнаваемым. Он может включать в себя графические элементы, текст или их комбинацию, которая ассоциируется с основными атрибутами и ценностями бренда.

Логотип сайта является важным элементом брендинга и визуальной идентичности. Вот несколько причин, почему логотип сайта является важным:

1. Идентификация бренда. Логотип является визуальным символом, который помогает потребителям быстро узнать и идентифицировать ваш бренд среди множества других на рынке.
2. Привлечение внимания. Хорошо разработанный логотип привлекает внимание посетителей сайта и делает бренд более запоминающимся в их сознании.
3. Профессионализм. Профессионально созданный логотип придаёт сайту видимость и ощущение доверия. Он помогает создать первое впечатление о компании, которое может повлиять на решение пользователей остаться на сайте и изучить его дальше.
4. Дифференциация от конкурентов. Уникальный логотип помогает вашему бренду выделиться на рынке и отличиться от конкурентов. Он помогает создать своеобразное "лицо" вашей компании, которое делает её уникальной и привлекательной для целевой аудитории.
5. Эмоциональное воздействие. Логотип может вызывать определенные эмоции у аудитории, что помогает укрепить связь с вашим брендом и повысить узнаваемость. Он может создавать положительные ассоциации, вызывать доверие или вызывать интерес к вашему бренду.
6. Отражение имиджа компании. Логотип должен быть согласован с общей имиджевой стратегией компании и отражать ее ценности, миссию, атмосферу и стиль. Он является визуальным представлением имиджа компании и должен передавать желаемый образ, который вы хотите создать в глазах своей аудитории.

Таким образом, логотип играет ключевую роль в формировании впечатления о вашей компании и влияет на восприятие бренда вашей целевой аудиторией.

Опираясь на вышеперечисленные причины, наиболее подходящий логотип для сайта Kaori Mau является изображение, представленное на рисунке 2.1.

Логотип выполнен под концепцию сайта. По центру располагается название пиццерии “Kaori Mau”. Чтобы вызвать у пользователя ассоциации с пиццей, сверху изображена печь. Также логотип содержит окружность оранжево-коричневого цвета, который напоминает форму пиццы, и надпись (слоган) “Crave The Taste” (в пер. “жаждать этого вкуса”) в нижней части логотипа.

Общий дизайн логотипа простой и современный, с основным акцентом на названии ресторана.

|  |
| --- |
|  |

Рисунок 2.1 – Логотип сайта «Kaori Mau»

2.5. Разработка пользовательских элементов

Концепция проекта включает в себя различные элементы пользовательского интерфейса, которые обеспечат взаимодействие с пользователем. Среди них: навигационное меню для удобного перемещения по сайту, блоки с информацией о товарах, и футер, содержащий основные данные и ссылки.

Навигационное меню – это не только обязательный элемент каждого веб-сайта, но и его кардинальная часть, определяющая удобство использования и навигацию по контенту. Это своего рода дорожная карта сайта, которая обеспечивает пользователей ссылками на различные страницы, обеспечивая им простой и интуитивно понятный способ перемещения по всем разделам и контенту сайта.

На сайте, навигационное меню (см. рис. 2.2) включает в себя логотип сайта и ссылки на три основные страницы: "Главная", "Меню" и "О нас". Фон навигационного меню выполнен в белом цвете с приятным персиковым оттенком, а снизу отделён светло-серой полосой.

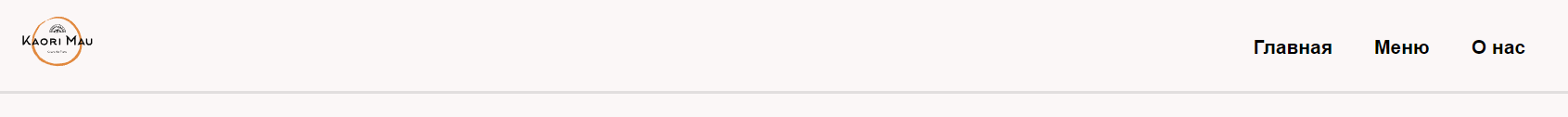


Рисунок 2.2 – Навигационное меню

Блоки с информацией о товарах являются основным компонентом веб-сайта. Они должны быть структурированы таким образом, чтобы пользователь мог легко ориентироваться и при этом получал удовольствие от визуального оформления и расположения элементов. Блоки с информацией о товарах должны включать следующую информацию:

1. Изображение товара. Следует использовать качественное изображение товара, которое ясно показывает его внешний вид.
2. Название товара. Предоставление краткого, но информативное названия, которое ясно описывает товар.
3. Цена. Расположение цены должно быть рядом с изображением и названием товара, чтобы пользователь мог легко увидеть основную информацию о цене.

Блок с информацией о секциях меню предоставлена на рисунке 2.3.



Рисунок 2.3 – Фрагмент меню

Каждый блок имеет следующую информацию: чёткое изображение товара, без лишних деталей, его наименование и цена.

Также, если пользователь нажмёт на любую секцию из меню, на экране появится модальное окно (см. рис. 2.4) с более подробной информацией о товаре.

Модальное окно – это всплывающее окно на веб-странице, которое появляется поверх всего остального и блокирует способность взаимодействовать с остальной частью страницы, пока оно открыто.

Это удобное решение, поскольку предоставляет возможность показать дополнительные детали о товаре без перегруженности основной страницы меню.

На данном модальном окне присутствуют то же изображение пиццы, но в более крупном размере, его наименование, цена и подробное описание, которое помогает пользователю принять решение о приобретении товара в зависимости от его предпочтений.

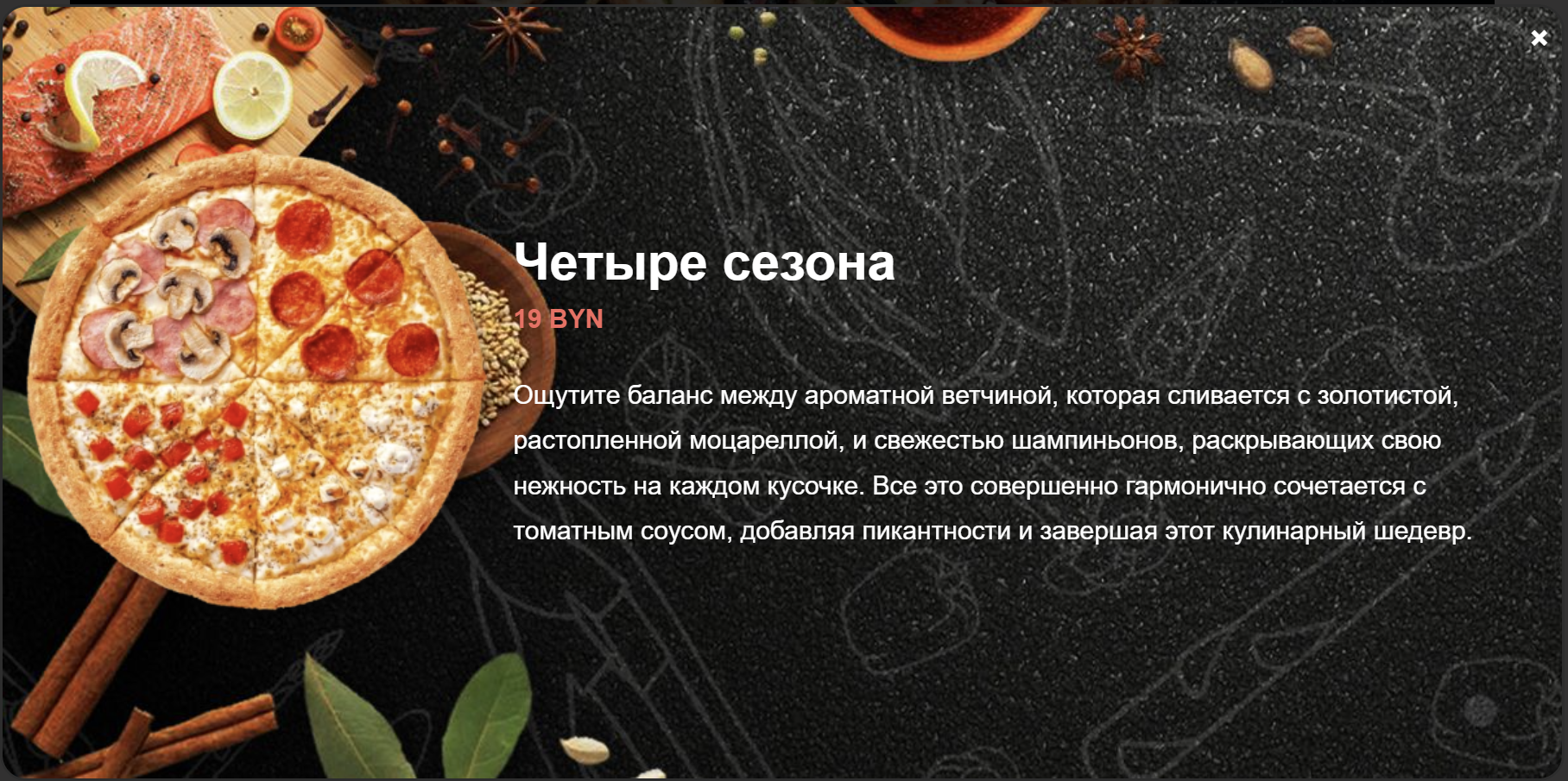


Рисунок 2.4 – Модальное окно

Футер – это нижняя часть веб-страницы, который обычно содержит информацию о контактах компании, ссылки на соцсети и т. п.

Футер, который представлен на рисунке 2.5, содержит в себе следующую информацию:

1. Информация для связи с компанией, включая адрес, номер телефона и электронную почту.
2. Ссылки на другие страницы веб-сайта, обеспечивающие удобную навигацию.
3. Ссылки на профили в социальных сетях, где пользователи могут получать актуальные новости и информацию о компании;
4. Ссылки на политику конфиденциальности и условия использования.

Футер также ограждён сверху светло-серой линией.

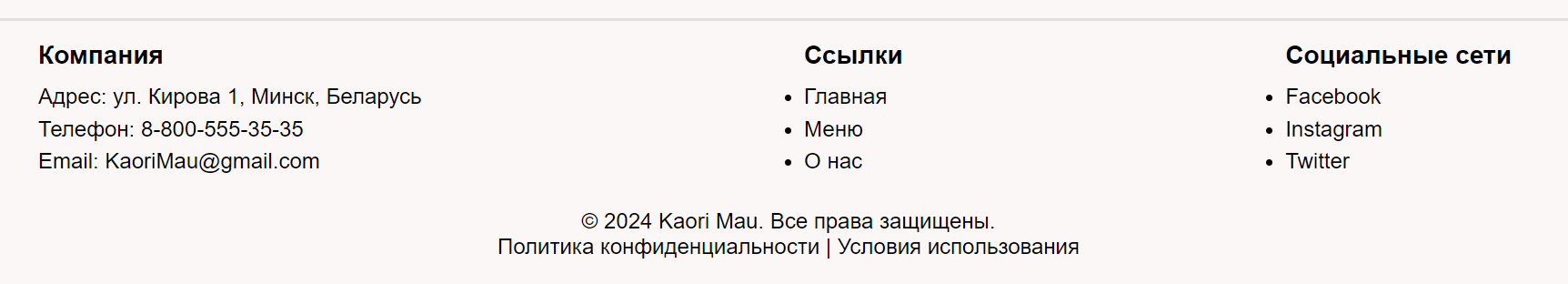


Рисунок 2.5 – Футер сайта

2.6 Разработка спецэффектов

На сайте присутствуют динамические эффекты, которые придадут сайту интерактивность и сделают его более привлекательным для пользователя.

В навигационном меню, футере, а также в меню сайта (при наведении на название товара) был реализован плавный переход цвета текста из одного в другой (см. рис. 2.6, 2.7 и 2.8).

Также была реализована плавная анимация появления и закрытия модального окна на странице.

Анимация кнопок при наведении также имеет некоторые плюсы: повышение привлекательности интерфейса, подчёркивание важности элемента, улучшенная визуальная обратная связь.

Пример реализации такой кнопки представлен на рисунке 2.9.

|  |
| --- |
| а б |

а – обычное состояние; б – состояние при наведении

Рисунок 2.6 – элемент навигационного меню

|  |
| --- |
| а б |

а – обычное состояние; б – состояние при наведении

Рисунок 2.7 – элемент футера

|  |
| --- |
| а б |

а – обычное состояние; б – состояние при наведении

Рисунок 2.8 – элемент меню

|  |
| --- |
| а б |

а – обычное состояние; б – состояние при наведении

Рисунок 2.9 – кнопка перехода к странице меню

2.7 Выводы

В данном разделе был разработан прототип веб-сайта. Были определены расположение элементов, дизайн навигационного меню, основной части веб-страниц и футера. Определены подходы к проектированию, а также стили, шрифты, главная цветовая схема страниц и спецэффекты для улучшения визуального восприятия.

# 3. Реализация структуры веб-сайта

3.1. Структура HTML-документа

Структура веб-сайта представляет собой организацию его контента и функциональности для обеспечения удобной навигации и понимания пользователем. Это включает в себя разделение сайта на страницы, определение их иерархии и взаимосвязей, а также организацию элементов на каждой странице для достижения оптимальной пользовательской доступности и удовлетворения потребностей аудитории.

Хорошая структура HTML-документа имеет большое значение для эффективной работы сайта. Вот несколько причин, почему это так важно:

1. Улучшение доступности. Хорошо структурированный HTML облегчает доступ к контенту сайта для людей с ограниченными возможностями и поисковых систем. Правильное использование заголовков, списков, атрибутов для изображений и других элементов помогает понять содержание страницы.
2. Улучшение производительности: Правильное использование семантических элементов HTML и размещение контента в соответствии с его значимостью позволяет браузерам эффективнее отображать страницы, что ускоряет загрузку и улучшает пользовательский опыт.
3. Облегчение обслуживания: Чистая и структурированная разметка делает код более читаемым и поддерживаемым для разработчиков. Это упрощает внесение изменений, исправление ошибок и добавление нового контента.

Структура главной страницы представлена в листинге 3.1.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>…</head>  <body>  </body>  <header></header>  <main>  <section class="first"> … </section>  <section class="second"> … </section>  <section class="third"> … </section>  </main>  <footer> … </footer>  </html> |

Листинг 3.1 – Структура HTML-документа

С использованием тега <header> была создана "шапка" сайта, которая также содержит навигационное меню. Тег <footer>, в свою очередь, используется для реализации "подвала" страницы, а тег <main>, включающий в себя теги <section>, служит для создания "тела" страницы, где хранится основная её часть.

3.2. Добавление таблиц стилей SCSS и CSS

В процессе разработки проекта были использованы разные методы подключения таблиц стилей, а именно внешнее, внутреннее и встраиваемое подключение.

|  |
| --- |
| <head>  <link rel="stylesheet" href="styles/header-footer.css">  <link rel="stylesheet" href="styles/main.css">  </head> |

Листинг 3.2 – Пример внешнего подключения CSS

|  |
| --- |
| <head>  <link rel="stylesheet" href="../styles/style-about-us.css">  <link rel="stylesheet" href="../styles/style-about-us.css.map">  </head> |

Листинг 3.3 – Пример внешнего подключения SCSS

Для стилизации повторяющихся элементов и основной разметки страниц было использовано внешнее подключение стилей (см. рис. 3.2, 3.3), где стили размещаются в отдельном файле и могут быть применены на всех страницах сайта. Внутреннее подключение (см. рис. 3.4) было применено для внесения изменений в контент на второстепенных страницах с минимальным объёмом кода.

Строковое подключение стилей (см. рис. 3.4) может быть полезным для стилизации уникальных элементов, которым не нужно повторно использовать стили на других страницах.

|  |
| --- |
| <style>  main {  display: flex;  background-color: #FCF7F7;  flex-direction: column;  align-items: center;  }  </style> |

Листинг 3.4 – пример внутреннего подключения

|  |
| --- |
| <figcaption style="font-size:36px;" class="btf">Четыре сезона</figcaption> |

Листинг 3.5 – Пример строкового подключения

Таблицы стилей были преимущественно реализованы с использованием селекторов по классу, потому что это позволяет более гибко и эффективно применять стили к определенным элементам на веб-странице. Использование классов в селекторах позволяет создавать множество стилей, которые могут быть применены к различным элементам с одинаковым классом.

Далее расположены примеры таблицы стилей CSS (см. рис. 3.6) и SCSS3 (см. рис. 3.7).

|  |
| --- |
| .centered-heading {  align-items: center;  text-align: center;  color: #ffffff;  margin-top: 200px;  font-size: 50px;  }  .subheading {  text-align: center;  color: #ffffff;  font-size: 16px;  margin-top: 15px;  text-decoration: none;  } |

Листинг 3.3 – Пример таблицы стилей CSS

|  |
| --- |
| @mixin img-style {  border-radius: 12px;  height: 225px;  width: 400px;  }  .our-information {  display: flex;  flex-direction: column;  img {  @include img-style;  }  } |

Листинг 3.3 – Пример таблицы стилей SCSS

SCSS имеет несколько преимуществ перед CSS, например:

1. Переменные. SCSS позволяет использовать переменные для хранения значений, таких как цвета, размеры и шрифты. Это упрощает и улучшает поддержку кода, поскольку значения могут быть легко изменены в одном месте, и эти изменения автоматически применятся ко всем местам, где используются переменные. Это также делает код более читабельным и удобным для обслуживания.
2. Вложенность. SCSS позволяет вкладывать стили внутри других стилей, что улучшает структурирование и читаемость кода. Вложенные стили отражают иерархию HTML-разметки и делают код более логичным и понятным. Кроме того, это упрощает написание и поддержку сложных селекторов.
3. Миксины. Миксины в SCSS позволяют определять и повторно использовать группы стилей. Это полезно, когда требуется применить один и тот же набор стилей к нескольким элементам.
   1. Использование стандартов XML (SVG)

XML является удобным форматом данных благодаря своей читаемости для человека, расширяемости, переносимости и широкому применению. XML использует простую синтаксическую структуру, позволяющую легко читать и понимать данные. Он также может быть настроен и расширен в соответствии с конкретными потребностями проекта. XML независим от платформы и может быть интерпретирован на разных системах, обеспечивая переносимость данных. Пример использования XML предоставлен на листинге 3.4.

|  |
| --- |
| <articles>  <title>  Почему стоит выбрать Kaori Mau?  </title>  <content>  Мы стремимся предложить самые аутентичные и вкусные блюда итальянской  кухни. Все ингредиенты импортируются из Италии, а наш шеф-повар имеет более чем  20-летний опыт работы. Мы предлагаем разнообразные пиццы, пасты и салаты, чтобы  удовлетворить ваши потребности.  </content>  </articles> |

Листинг 3.4 – Пример использования стандартов XML

SVG-код представлен в приложении X.

SVG плюсы

~~Введение (указывается цель курсового проекта, задачи курсового проекта, целевая аудитория веб-сайта и актуальность использования веб-сайта по выбранной теме)~~

1. Постановка задачи

1.1. Обзор аналогичных решений

~~1.2. Техническое задание (информация о проекте (количество веб-страниц и что они будут содержать), формирование требований к программному продукту, формулировка задач программного продукта, то есть то, что должен реализовывать ваш продукт)~~

~~1.3. Выбор средств реализации программного продукта~~

~~В данном подразделе описать выбор языка разметки, инструментов для создания веб-сайта и библиотек, которые будут подключаться~~

~~1.4. Вывод~~

~~В конце каждого раздела должен быть вывод. В данном выводе можете написать про выявленный основной функционал вашего программного продукта и про стек технологий, который будете использовать~~

~~2. Проектирование страниц веб-сайта~~

~~В данном разделе составить макет используя Figma или другой графический редактор, обосновать выбор оформления веб-сайта, представить в графическом виде расположение элементов на веб-сайте и какие стили (эффекты) будут применены к этим элементам~~

~~2.1. Выбор способа верстки~~

~~2.2. Выбор стилевого оформления~~

~~2.3. Выбор шрифтового оформления~~

~~2.4. Разработка логотипа (описание лого)~~

~~2.5. Разработка пользовательских элементов~~

~~2.6. Разработка спецэффектов~~

~~2.7. Выводы~~

3. Реализация структуры веб-сайта

Представить листинги структуры веб-сайта. Обосновать выбор элементов в коде.

Если это большой важный кусок кода, то покажите его в приложение в виде листинга, но укажите ссылку на приложение. Например, скрипт для создания таблиц представлен в приложении Б. В листингах интервалов после абзацев быть не должно.

~~3.1. Структура HTML-документа~~

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

3.3. Использование стандартов XML (SVG)

3.4 Управление элементами DOM

3.4. Выводы

4. Тестирование веб-сайта

4.1. Адаптивный дизайн веб-сайта

Представить как осуществляться будет адаптивность вашего сайта под разные устройста в различных браузерах

4.2. Кроссбраузерность веб-сайта

В кроссбраузерное тестирование входит общий вид вашего проекта в других браузерах (сохранились ли шрифты, не съехала ли анимация и так далее)

4.3. Руководство пользователя

Описание того, как пользователю пользоваться вашим сайтом со скриншотами, а также как он может взаимодействовать с основными ключевыми функциями вашего продукта.

4.4. Выводы

Заключение (подытожить что было сделано вами, что было вами использовано дополнительно не встречающееся в курсе, ссылка на репозитарий Github с веб-сайтом курсового проекта)

Список использованных источников.

Приложение (полный исходный текст программы разработанного приложения с подробными комментариями в виде листинга).

Приложение А Прототипы веб-страниц

Приложение Б Макет структуры веб-сайта

Приложение В Листинг НТML-документа

Приложение Г Листинг SCSS и CSS

Приложение Д Листинг XML-файлов

Приложение Е Листинг SVG

Приложение Ж Листинг JavaScript

Куда фигму пихать