

Министерство науки и высшего образования Российской Федерации
Федеральное государственное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет среднего профессионального образования
Специальность: 09.02.03 Программирование в компьютерных системах

ДОПУСТИТЬ К ЗАЩИТЕ
Заместитель декана по УР
_____ Н.В. Дрёмина
«_____» _____ 2023г.

ДИПЛОМНЫЙ ПРОЕКТ

на тему: Разработка сайта для магазина быстрого питания

Исполнитель

студент группы ПКС204

(подпись, дата)

Н.К. Зайцев

Руководитель

(подпись, дата)

О.В. Жилиндина

Нормоконтроль

(подпись, дата)

Д.М. Яковлева

Благовещенск 2023

Министерство науки и высшего образования Российской Федерации
Федеральное государственное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

УТВЕРЖДАЮ
Заместитель декана по УР
_____ Н.В. Дрёмина
«_____» _____ 2023г.

ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ

Студенту: Зайцеву Никите Константиновичу

Специальность: 09.02.03 Программирование в компьютерных системах

Руководитель ВКР: Жилиндина Ольга Викторовна

1. Тема ВКР: Разработка сайта для магазина быстрого питания

Утверждена приказом №735-уч от 27.03.2023 г.

2. Срок сдачи ВКР: 25.05.2023г.

3. ВКР предоставляется с:

А) пояснительной запиской

4. Содержание пояснительной записки

– Были изучены технологии создания веб-сайтов и разработан сайт для магазина быстрого питания;

Дата выдачи задания: 01 апреля 2023 года

Руководитель ВКР _____

О.В. Жилиндина

Задание принял и исполнил _____

Н.К. Зайцев

РЕФЕРАТ

Дипломный проект «Разработка сайта для магазина быстрого питания» содержит 47 страниц, 15 рисунков, 26 источников информации.

САЙТ, РАЗРАБОТКА, ВЁРСТКА, HTML, CSS, JAVASCRIPT, PHP, PHPMYADMIN, SQL, MYSQL, FIGMA, МАГАЗИН, КОРЗИНА, СКРИПТ

В работе исследованы актуальные технологии разработки веб-сайтов.

Цель работы: Разработка сайта для магазина быстрого питания.

В ходе выполнения выпускной квалификационной работы были проанализированы и применены технологии создания веб-сайтов с использованием языков HTML, CSS, JavaScript, PHP, SQL.

С помощью указанных языков программирования был разработан многостраничный веб-сайт, который даёт возможность ознакомиться с меню и оформить заказ.

Результатом выпускной квалификационной работы является разработанный сайт для магазина быстрого питания.

СОДЕРЖАНИЕ

Введение	7
1 Теоретическая часть	9
1.1 Обоснование потребности в web-сайте	10
1.2 Понятие web-сайта	10
1.3 Этапы разработки web-сайта	11
1.4 Средства разработки	12
1.4.1 HTML	13
1.4.2 CSS	14
1.4.3 JavaScript	15
1.4.4 Back-end языки. Обоснование выбора PHP	16
1.4.5 MySQL	18
1.4.6 PHPMyAdmin	19
1.4.7 Редакторы кода. Обоснование выбора Sublime Text	20
1.4.8 Figma	21
2 Проектная часть	23
2.1 Общая структура сайта	23
2.2 Процесс разработки клиентской части сайта	24
2.3 Адаптивность сайта	31
2.4 Процесс создания и подключения базы данных	34
2.5 Процесс разработки скриптов для сайта	38
Заключение	44
Библиографический список	46

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

БД (база данных) – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных;

СУБД (Система управления базами данных) – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных (БД);

Вёрстка веб-страниц – создание структуры гипертекстового документа на основе HTML-разметки, как правило, при использовании таблиц стилей и клиентских сценариев, таким образом, чтобы элементы дизайна выглядели аналогично макету;

Интернет-хранилище или DOM-хранилище – это программные методы и протоколы веб-приложения, используемые для хранения данных в веб-браузере. Интернет-хранилище представляет собой постоянное хранилище данных, похожее на COOKIE, но со значительно расширенной ёмкостью и без хранения информации в заголовке запроса HTTP;

Скрипт – это понятие в программировании, которое обозначает последовательность команд для выполнения конкретных операций. По сути, это небольшая программа, заточенная под определенное действие. Скрипты используются в веб-разработке, в частности, чтобы автоматизировать операции;

CSS (Cascading Style Sheets) – формальный язык декорирования и описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML). Также может применяться к любым XML-документам, например, к SVG или XUL;

DOM (Document Object Model) – это стандартный способ представления HTML и XML документов в виде иерархической структуры, которая позволяет программно манипулировать содержимым и стилями элементов веб-страницы. DOM объединяет все элементы HTML/XML в дерево, в котором каждый элемент является узлом, а его атрибуты и текст являются его потомками. DOM также

предоставляет методы для доступа к каждому элементу документа и его атрибутам, а также для добавления, изменения и удаления элементов и их содержимого;

HTML (от англ. HyperText Markup Language – «язык гипертекстовой разметки») – стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора;

JavaScript (JS) – язык программирования, который используется для создания динамических веб-страниц и интерактивных веб-приложений;

PHP (Hypertext Preprocessor) – С-подобный скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов;

PHP Data Objects – расширение для PHP, предоставляющее разработчику универсальный интерфейс для доступа к различным базам данных. PDO предлагает единые методы для работы с различными базами данных, хотя текст запросов может немного отличаться;

PHPMyAdmin – веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных;

SQL (Structured Query Language) – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

ВВЕДЕНИЕ

Сегодня интернет является неотъемлемой частью нашей жизни. Благодаря быстрому развитию технологий и всеобщей доступности сети, люди могут получать информацию, общаться, работать, делать покупки и многое другое в онлайн-режиме. Строительство бизнеса без участия интернета теперь практически невозможно. Компании, которые отказываются от создания своего сайта или онлайн-магазина, лишают себя возможности привлечь большее количество потенциальных клиентов и увеличить прибыль.

Сайт для магазина быстрого питания – это мощное маркетинговое средство. Интернет-пользователи предпочитают решать свои вопросы в онлайн-режиме, без необходимости посещать место физический. Сайт сделает услуги магазина быстрого питания более удобными и доступными для большинства клиентов. Люди смогут просматривать меню и составлять заказы онлайн, а затем получать готовую еду на вынос или с доставкой на дом или в офис.

Наличие своего сайта для магазина быстрого питания позволит увеличить число клиентов, повысить узнаваемость бренда и расширить географию продаж. Также владельцы магазина смогут более эффективно управлять заказами, обрабатывать данные клиентов и запускать рекламную кампанию в интернете. Сайт для магазина быстрого питания – это не просто удобство для клиентов, но и выгодное вложение в развитие бизнеса.

Этот сайт будет являться удобным и быстрым ресурсом для заказа еды, который позволит увеличить число клиентов и удовлетворить их потребности.

В работе изучены и применены основные принципы и подходы к созданию функционального и привлекательного сайта. Акцент сделан на использование современных технологий и инструментов для разработки сайта, таких как HTML, CSS, JavaScript, PHP, MySQL.

Цель выпускной квалификационной работы: верстка сайта, создание и подключение базы данных, разработка алгоритма работы сайта.

Предмет исследования: проектирование и разработка сайта с учётом потребностей деятельности магазина.

Задачи исследования:

- Анализ и выбор методов разработки сайтов;
- Создание макета с учётом основных принципов веб-дизайна;
- Разработка базы данных сайта;
- Разработка алгоритма отображения контента из базы данных;
- Разработка алгоритмов заполнения и редактирования корзины;
- Разработка алгоритма формирования заказа.

Объектом исследования является процесс создания функционального и привлекательного веб-ресурса, который является удобным и быстрым способом заказа еды для широкой аудитории потенциальных клиентов.

В работе будет рассмотрен весь процесс создания сайта – от выбора дизайна и создания макета до программирования функциональных возможностей, таких как корзина заказов, фильтры по категории, обработка заказов и другие функции, которых ожидают от современного веб-ресурса.

Основными методами исследования объекта в работе являются анализ литературы, обзор существующих сайтов для магазинов быстрого питания, анализ тенденций в веб-разработке, программирование веб-страниц, написание сценариев для веб-приложений, статистический анализ полученных данных.

Результаты проекта в целом могут внести вклад в развитие бизнеса в области фаст-фуда и сферы онлайн-торговли в целом. Разработанный веб-ресурс будет полезным не только для магазина быстрого питания, но и для других компаний, которые стремятся создать эффективный и функциональный веб-сайт.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Обоснование потребности в web-сайте

Современный ритм жизни требует от людей мобильности и быстроты во всех сферах. Одной из сфер, которые особенно затронула эта тенденция, стала сфера быстрого питания. Быстрое питание – это прием пищи, который не требует затрат времени на ее приготовление и сервировку. Магазины быстрого питания предоставляют такие услуги, предлагая клиентам широкий ассортимент блюд, которые можно заказать, оплатить и получить за короткий промежуток времени.

Многие кафе и рестораны имеют свой web-сайт, зайдя на который клиент может ознакомиться с меню, почитать информацию о блюдах, о самом ресторане, забронировать столик или заказать доставку еды. Свой web-сайт является мощным маркетинговым средством: он позволяет не только удерживать постоянных клиентов, но и получить новых. Если у ресторана есть свой сайт – потенциальным покупателям совсем не обязательно приходить лично, чтобы ознакомиться с меню и оформить заказ, это можно сделать из любой точки города.

Через сайт магазина быстрого питания можно предложить клиентам удобный способ заказа еды на доставку или самовывоз. Возможность изучить онлайн-меню со всеми блюдами, ценами и описанием поможет клиентам выбрать те блюда, которые им больше всего нравятся.

Наличие сайта для магазина быстрого питания является важной частью эффективного управления бизнесом. Он помогает владельцам увеличить прибыль и расширить клиентскую базу за счет более широкой доступности и удобства предоставления услуг [14].

В данном дипломном проекте будет рассмотрена разработка сайта для магазина быстрого питания “Burger City”, который будет позволять клиентам выбрать блюда и заказать доставку. Разрабатываемый сайт должен удовлетворять требованиям быстроты и доступности, чтобы клиенты магазина могли быстро и легко заказать необходимое блюдо в удобное для них время.

Сайт позволит клиентам быстро и удобно узнать меню, цены, график работы и местоположение магазина, а также сделать заказ. Это особенно удобно для людей, которые хотят заказать еду на вынос или доставку, но не хотят тратить время на посещение магазина.

1.2 Понятие web-сайта

Веб-сайт – это множество взаимосвязанных между собой веб-страниц (HTML-документов), которые связаны между собой гиперссылками и доступны пользователю через Интернет. Сайты играют достаточно важную роль в информационной эпохе и очень широко используются в бизнесе, коммерции, государственном управлении, образовании и других сферах деятельности.

Как правило, веб-сайты разрабатываются с использованием языков программирования, таких как HTML, CSS и JavaScript. Эти языки позволяют создавать динамические и интерактивные страницы, которые могут реагировать на взаимодействие пользователя [12].

Веб-сайты могут иметь различные типы содержимого, такие как текст, изображения, видео, аудио, формы и другие элементы. Некоторые сайты также могут включать базы данных и серверные скрипты для обработки запросов пользователя.

Существует множество различных типов веб-сайтов, таких как личные блоги, корпоративные сайты, интернет-магазины, новостные порталы, социальные сети и многое другое. Каждый тип сайта имеет свои уникальные особенности и цели.

Веб-сайты также могут быть разработаны с использованием различных платформ и инструментов, таких как WordPress, Joomla, Drupal, Tilda и т.д. Эти платформы облегчают создание веб-сайтов и уменьшают количество необходимого программирования.

Веб-сайты играют ключевую роль в современном интернете, предоставляя пользователю доступ к информации, продуктам и услугам. Они являются важным инструментом для бизнеса, маркетинга и коммуникации, и их значение только увеличивается с каждым годом [20].

1.3 Этапы разработки web-сайта

Разработка web-сайта включает несколько этапов, каждый из которых имеет свои особенности и задачи. Ниже приведены основные этапы создания web-сайта:

- Анализ требований заказчика. На этом этапе необходимо определить требования к сайту, его цели и задачи, целевую аудиторию, конкурентов, особенности контента и технические требования к сайту.
- Разработка технического задания. На основе анализа требований заказчика создается техническое задание, которое включает описание функциональных и нефункциональных требований к сайту.
- Проектирование структуры и дизайна сайта. На этом этапе начинается создание концепции сайта: проектируется структура сайта, определяется внешний вид, цветовая гамма, шрифты и другие элементы дизайна.
- Создание контента. Создание контента для сайта начинается на этапе проектирования. Контент может включать текстовую информацию, изображения, видео и другой мультимедийный контент.
- Разработка и программирование сайта. На этом этапе происходит реализация дизайна и функциональности сайта. Разработчики используют языки программирования (HTML, CSS, JavaScript, PHP и др.) и специализированные CMS, фреймворки и библиотеки для создания определенного функционала.
- Тестирование сайта. После завершения разработки сайта происходит тестирование, в ходе которого проверяется работоспособность сайта, корректность отображения контента на различных устройствах и в различных браузерах.
- Запуск сайта и его сопровождение. После успешного прохождения тестирования сайт запускается. В дальнейшем сайт требует регулярного сопровождения, обновления контента, поправку ошибок и обеспечения безопасности.

Разработка веб-сайта – длительный и сложный процесс, который требует тщательного планирования и внимательного следования каждому из этапов.

Важно держать в уме основные цели сайта, аудиторию, которой он будет предназначен, а также обеспечить его удобство и безопасность для пользователей. Развитие и поддержка сайта также являются неотъемлемыми этапами, необходимыми для его успешного функционирования. Команда разработчиков должна иметь необходимые ресурсы и знания, чтобы создать и поддерживать высококачественный веб-сайт.

1.4 Средства разработки

Веб-сайт – это интерактивный процесс, который состоит из множества компонентов, таких как изображения, текст, видео, музыка и другие элементы. При создании веб-сайта, разработчики обычно делят работу на две части: Front-end и Back-end [15]. Front-end разработка – это процесс создания пользовательского интерфейса веб-сайта или приложения. В этой области программирования используются различные языки и технологии.

Основным языком программирования во Front-end разработке является JavaScript. Он используется для создания динамических и интерактивных элементов на веб-страницах. На JavaScript пишутся скрипты для логики работы каких-либо компонентов. Кроме того, JavaScript используется вместе с HTML и CSS для создания анимации и визуальных эффектов [16].

Back-end – это часть веб-приложения, которая занимается обработкой и хранением данных, а также управлением логикой, необходимой для функционирования приложения. Она состоит из серверной части, которая представляет собой набор программ и баз данных, которые обеспечивают обработку запросов, сохранение информации и её передачу клиентской части приложения.

Основным языком программирования для Back-end являются такие языки программирования, как Python, PHP, Java или Node.js, и они используются для написания кода серверной части. Back-end-разработчики должны понимать принципы веб-протоколов, баз данных, а также иметь опыт работы с управлением серверами и сетями. Без Back-end веб-приложение не может

функционировать, так как клиентская часть приложения должна быть связана с серверной для получения необходимой информации.

Для работы с реляционными базами данных на серверной стороне сайта используется язык SQL (Structured Query Language). Его применяют для создания, изменения, удаления и извлечения данных из базы данных на сервере.

Кроме того, для создания веб-сайта также понадобится выбрать редактор кода, который будет использоваться для написания и отладки кода серверной и клиентской частей веб-приложения.

Для работы с графикой используются различные инструменты, включая Adobe Photoshop, Sketch, Adobe Illustrator, Figma, InVision и т.д. Эти инструменты позволяют создавать красивый дизайн для веб-сайта и разрабатывать макеты пользовательского интерфейса.

Далее будет подробно описан стек используемых технологий.

1.4.1 HTML

HTML (HyperText Markup Language) – это один из первых языков, который был разработан для создания веб-страниц. Он является одним из ключевых элементов веб-разработки, и без него невозможно создать современный сайт.

HTML используется для создания структуры и содержания веб-страницы. Это позволяет разработчикам определить, какие элементы будут присутствовать на странице, и как они будут выглядеть. С помощью HTML можно создавать заголовки, параграфы, списки, таблицы, формы для ввода данных, ссылки, изображения и многое другое.

Кроме того, HTML является языком, который позволяет создавать не только статические, но и динамические веб-страницы. Он позволяет разработчикам создавать интерактивные и многофункциональные веб-страницы, которые удовлетворяют потребности пользователей и предоставляют им удобный и понятный интерфейс.

HTML является одним из самых распространенных языков программирования в мире. Большинство сайтов в интернете используют HTML как основу для своих веб-страниц. Сегодня, с развитием технологий, HTML

продолжает развиваться и становится более мощным и удобным для создания веб-страниц.

В целом, HTML является необходимым языком для создания веб-страниц. Он позволяет создавать удобные и функциональные пользовательские интерфейсы для веб-сайтов и приложений, что делает его очень важным языком для веб-разработки.

1.4.2 CSS

CSS (Cascading Style Sheets) – это язык стилей, который играет важную роль в создании визуального впечатления веб-страниц. CSS позволяет разработчикам определять все аспекты дизайна, включая цвета, шрифты, размеры, отступы и многое другое, что позволяет создавать совершенно разные и уникальные внешние образы веб-страниц. Без CSS веб-страницы могли бы выглядеть скучно и неинтересно, а с помощью этого языка стилей, разработчики могут превратить веб-страницы в настоящие произведения искусства.

CSS позволяет создавать не только простые, но и сложные макеты и дизайны, что делает веб-страницы более функциональными и удобными для пользователей. Например, с помощью CSS можно создавать адаптивный дизайн, что позволяет веб-страницам корректно отображаться на разных устройствах и экранах, включая мобильные устройства.

Кроме того, CSS позволяет создавать интересные и креативные анимации, переходы и эффекты, которые делают веб-страницы более живыми и интерактивными для пользователей. CSS также обеспечивает кросс-браузерную совместимость, что означает, что веб-страницы, созданные с помощью CSS, могут отображаться одинаково на разных браузерах и устройствах.

В CSS существуют технологии Grid и Flex. Это два различных подхода для создания адаптивных макетов в CSS.

Flexbox (или просто Flex) это гибкий способ расположения элементов внутри контейнера. Он используется для создания одномерных макетов, то есть расположения элементов либо в строку, либо в колонку. С помощью Flexbox

можно легко управлять направлением, порядком, выравниванием и отступами элементов.

Grid является более продвинутым методом для создания двумерных макетов, то есть расположения элементов как в строку, так и в колонку одновременно. Grid используется для создания сложных сеток, где можно задавать размеры и позиции элементов на точном уровне. Grid позволяет создавать сложные дизайны с несколькими столбцами и рядами, а также легко добавлять или удалять ячейки в сетке. Использование Grid и Flex вместе позволяет создавать более сложные и адаптивные макеты, которые будут отлично работать на любых устройствах.

Media queries позволяют менять стили элементов в зависимости от ширины экрана, разрешения устройства, ориентации, пиксельной плотности и других критериев. Таким образом, можно создавать адаптивные веб-страницы, которые будут хорошо выглядеть на любых устройствах и в любых условиях. Например, можно задать различные стили для мобильных устройств с разными размерами экранов или для печати в случае, если пользователь захочет распечатать инструкцию, рецепт или другой документ. Использование media queries является важной частью разработки современного сайта и позволяет улучшать его визуальную и функциональную часть в разных сценариях использования.

1.4.3 JavaScript

JavaScript является одним из самых популярных языков программирования в мире. Этот скриптовый язык разработки создан в 1995 году Бренданом Айком компанией Netscape Communications Corporation. Он был создан для обратной совместимости с браузерами, но со временем стал широко используемым на сервере и даже на уровне операционных систем [21].

JavaScript используется для создания интерактивных сайтов, динамических пользовательских интерфейсов и для решения различных задач на стороне клиента и на стороне сервера. Кроме того, JavaScript используется в различных сферах, таких как разработка мобильных приложений, игры, аналитика и другое.

С помощью JavaScript можно создавать анимации, изменять содержимое страницы, работать с формами и многое другое. Он может использоваться для обработки данных, разработки веб-приложений, построения API и многого другого.

Одной из ключевых особенностей JavaScript является его возможность асинхронного выполнения, что позволяет выполнять несколько задач одновременно без блокировки последующих процессов. Это значительно повышает производительность веб-приложений [13].

Существует множество библиотек и фреймворков, которые расширяют возможности JavaScript и упрощают его использование. Например, React, Angular, Vue.js, Node.js и многие другие.

JavaScript продолжает развиваться, появляются новые функциональные возможности и стандарты, такие как ECMAScript 6 (ES6), который включает в себя новые конструкции языка, такие как стрелочные функции, классы и т.д.

В целом, JavaScript – это универсальный язык программирования, который используется во многих областях и продолжает развиваться, что делает его эффективным инструментом для создания веб-приложений и не только [9].

1.4.4 Back-end языки. Обоснование выбора PHP

Когда речь идет о выборе языка для серверной части веб-приложения, существует несколько вариантов. Python, Java, NodeJS и PHP – одни из самых популярных языков, используемых для этих целей. Каждый язык имеет свои преимущества и недостатки, и выбор зависит от нескольких факторов, таких как требования проекта, квалификация команды разработчиков, требования к производительности и масштабируемости.

Python – это популярный язык программирования, который известен своей простотой, читабельностью и легкостью в использовании. Он широко используется для разработки веб-приложений, научных вычислений, анализа данных и искусственного интеллекта. Python – интерпретируемый язык, что означает, что его не нужно компилировать перед запуском кода. Это облегчает быструю разработку и тестирование приложений. Python также имеет большое и

активное сообщество, а это значит, что существует множество библиотек и фреймворков, которые помогут разработчикам создавать сложные приложения.

Java – еще один популярный язык, который широко используется для разработки приложений корпоративного уровня. Он известен своей масштабируемостью, безопасностью и надежностью. Java – компилируемый язык, что означает, что перед запуском приложения код должен быть скомпилирован. Это делает разработку и тестирование приложений немного более сложным по сравнению с Python. Однако Java – это зрелый язык с большим и активным сообществом, а это значит, что существует множество инструментов, библиотек и фреймворков, которые помогут разработчикам создавать надежные и масштабируемые приложения [8].

NodeJS – это среда выполнения JavaScript, построенная на движке V8 JavaScript в Chrome. Она позволяет разработчикам создавать быстрые и масштабируемые сетевые приложения, используя JavaScript на стороне сервера. NodeJS ориентирован на события и является неблокирующим, что означает, что он может одновременно обрабатывать большое количество соединений, не блокируя выполнение других запросов. Это делает NodeJS идеальным для создания приложений реального времени, таких как чат-приложения, онлайн-игры и потоковые сервисы. NodeJS также имеет большое и активное сообщество, что означает, что существует множество модулей и фреймворков, которые помогут разработчикам создавать сложные приложения.

Выбор PHP для проекта обоснован тем несколькими факторами. Одним из главных преимуществ PHP является простота использования. PHP это простой и интуитивно понятный язык, который легко изучить даже новичкам. Это делает его отличным выбором для небольших проектов или проектов с сжатыми сроками. PHP также имеет большое и активное сообщество, а это значит, что существует множество ресурсов, которые помогут разработчикам в изучении и устранении любых проблем, с которыми они могут столкнуться.

Еще одним преимуществом PHP является его гибкость. PHP можно использовать для создания широкого спектра приложений, от простых скриптов

до сложных веб-приложений. PHP также может быть интегрирован с другими технологиями, такими как базы данных, веб-серверы и другие языки программирования. Это делает его отличным выбором для создания масштабируемых и гибких приложений, которые могут расти вместе с бизнесом.

PHP также известен своей скоростью и производительностью. PHP оптимизирован для веб-разработки и может обрабатывать большое количество запросов одновременно. Это делает его отличным выбором для создания веб-сайтов и приложений с высокой посещаемостью. PHP также имеет встроенный механизм кэширования, который может повысить производительность приложения за счет уменьшения количества запросов к серверу.

В заключение можно сказать, что PHP является отличным выбором для разработки веб-приложений, требующих динамического контента и высокой производительности. Он прост в изучении, гибок и имеет большое и активное сообщество. Хотя другие языки, такие как Python, Java и NodeJS, имеют свои преимущества, PHP является лучшим выбором для данного проекта благодаря простоте использования, гибкости и скорости.

1.4.5 MySQL

MySQL – это бесплатная и открытая реляционная система управления базами данных (СУБД), которая широко используется во всем мире. Она разработана компанией Oracle Corporation и доступна для использования под лицензией GPL [2].

MySQL поддерживает множество функций, которые делают ее одной из наиболее популярных СУБД на рынке. Она поддерживает многие языки программирования, такие как PHP, Java, C++, Python и другие. Кроме того, MySQL имеет хорошую производительность и надежность, что делает ее идеальным выбором для создания больших и сложных приложений.

Среди основных функций MySQL можно отметить:

- Поддержка транзакций и блокировок, что обеспечивает целостность данных и защиту от конкурентного доступа.

- Мощный язык запросов SQL, позволяющий выполнять множество операций с данными, таких как выборка, добавление, изменение и удаление.
- Система безопасности, которая обеспечивает контроль доступа к данным и защиту от несанкционированного доступа.
- Поддержка репликации, которая позволяет создавать копии баз данных и обеспечивать высокую доступность.
- Возможность работать с различными типами данных, такими как числа, строки, даты и времена, а также поддержка геоданных.
- MySQL также имеет множество инструментов для управления базами данных, таких как MySQL Workbench, phpMyAdmin и другие. Они позволяют создавать, изменять и удалять таблицы, выполнять запросы и отслеживать производительность баз данных.

Таким образом, MySQL – это мощная и надежная система управления базами данных, которая подходит для широкого спектра приложений и может быть использована как начинающими, так и профессиональными разработчиками [22].

1.4.6 PHPMyAdmin

PHPMyAdmin – это бесплатный веб-интерфейс для управления базами данных MySQL. Он позволяет пользователям легко и удобно управлять, и администрировать базы данных с помощью стандартных веб-браузеров.

Основным преимуществом PHPMyAdmin является его интуитивно понятный интерфейс. С его помощью можно легко создавать новые базы данных, таблицы, пользователей, а также изменять и удалять уже существующие данные. Также PHPMyAdmin позволяет выполнять различные запросы к базам данных, отслеживать их статус и получать полезную информацию.

PHPMyAdmin содержит множество функций и инструментов, таких как создание и удаление баз данных, управление пользователями и правами доступа, резервное копирование данных и восстановление, работа с триггерами и процедурами, визуализирование структуры базы данных, а также многое другое.

RНРMyAdmin является открытым программным обеспечением, которое можно бесплатно скачать и использовать. Оно поддерживается сообществом разработчиков, что позволяет обеспечить его постоянное обновление и улучшение.

В целом, RНРMyAdmin является мощным и удобным инструментом для администрирования баз данных MySQL. Его преимущества включают легкость в использовании, множество функций и возможность работы с несколькими серверами одновременно.

1.4.7 Редакторы кода. Обоснование выбора Sublime Text

Для написания кода программисты пользуются специальными текстовыми редакторами. Это самостоятельная компьютерная программа или компонент программного комплекса, предназначенная для создания и изменения текстовых данных в общем и текстовых файлов, в частности.

Для веб-разработки наиболее популярны такие редакторы кода, как Notepad++, Sublime Text, Visual Studio Code, WebStorm.

Notepad++ – бесплатный текстовый редактор с открытыми исходными кодами для операционных систем Windows. Он поддерживает более 80 языков программирования и позволяет быстро и легко редактировать текстовые файлы. Notepad++ также поддерживает множество плагинов, которые добавляют дополнительные функции и возможности редактирования текста. В совокупности все эти возможности делают Notepad++ одним из наиболее популярных выборов для программистов и разработчиков, работающих с различными языками программирования и редактирования текстовых файлов.

Visual Studio Code (VS Code) – бесплатный, открытый редактор кода, применяемый в первую очередь для разработки программного обеспечения. Он был разработан компанией Microsoft и позволяет работать с различными языками программирования, такими как C#, Python, JavaScript, TypeScript и многими другими. Большая часть функциональности VS Code доступна благодаря множеству расширений и плагинов, созданных сообществом open source.

WebStorm – это интегрированная среда разработки (IDE), разработанная компанией JetBrains, которая предназначена для разработки веб-приложений. WebStorm интегрируется с другими инструментами, такими как Git, NPM и Node.js. Он также поддерживает язык TypeScript и работу с различными фреймворками, такими как AngularJS, React и Vue.js. WebStorm является платным редактором, но предоставляется бесплатный пробный период для использования. Этот редактор определенно не для всех, но именно те, кто работают с веб-разработкой, могут ощутить все преимущества WebStorm в полной мере.

Sublime Text – это мощный текстовый редактор, который пользуется огромной популярностью среди программистов и разработчиков. Он доступен для Windows, macOS и Linux.

Основные преимущества Sublime Text заключаются в его скорости работы и гибкости настройки. Редактор быстро открывает большие файлы и обеспечивает быстрое редактирование текста. Кроме того, пользователи могут легко настроить ключевые комбинации клавиш, цветовую схему, шрифты и другие параметры интерфейса.

Sublime Text также поддерживает множество различных языков программирования и форматов файлов, включая HTML, CSS, JavaScript, Python и многие другие. В редакторе есть много полезных функций, таких как автодополнение кода, выделение синтаксиса, готовые шаблоны кода и поддержка плагинов.

Поэтому, для этого проекта был выбран Sublime Text. Это отличный вариант для программистов и разработчиков, которые ищут быстрый и гибкий текстовый редактор с множеством возможностей настройки.

1.4.8 Figma

Figma – это мощный инструмент для дизайна интерфейсов, который позволяет работать в облачном режиме. Он широко используется дизайнерами и разработчиками, которые работают с различными проектами и хотят упростить процесс создания интерфейсов.

Одной из основных причин популярности Figma является его возможность работы в онлайн режиме. Это позволяет пользователям работать над проектами удаленно, обмениваться файлами и результатами своей работы с другими членами команды. Кроме того, Figma имеет множество функций и инструментов, которые облегчают процесс дизайна интерфейсов.

Figma позволяет пользователям создавать векторные изображения, макеты и прототипы интерфейсов, а также использовать готовые компоненты и элементы дизайна, чтобы ускорить процесс работы. С помощью этого инструмента можно создавать многоуровневые системы дизайна, а также простые и сложные анимации, что делает его излюбленным инструментом для создания пользовательских интерфейсов.

Figma также позволяет пользователям легко поделиться своими проектами и работать в режиме реального времени, что делает этот инструмент идеальным выбором для работы в команде. Интерфейс Figma интуитивно понятен, что упрощает процесс освоения и использования этого инструмента для создания дизайна интерфейсов.

Одним из недостатков Figma является его стоимость. Для использования всех функций этого инструмента нужно купить подписку, которая может быть достаточно дорогой для многих пользователей. Однако, бесплатная версия Figma также имеет множество полезных функций и является отличным выбором для новичков в области дизайна интерфейсов.

В целом, Figma – это мощный инструмент для создания дизайна веб-интерфейсов, который обладает большим количеством функций и возможностей для работы в команде. Его интуитивно понятный интерфейс и возможность работы в облачном режиме делают его одним из лучших выборов для дизайнеров и разработчиков.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Общая структура сайта

На Рисунке 1 представлена структурная схема сайта, которая учитывает всю специфику разработки приложения.

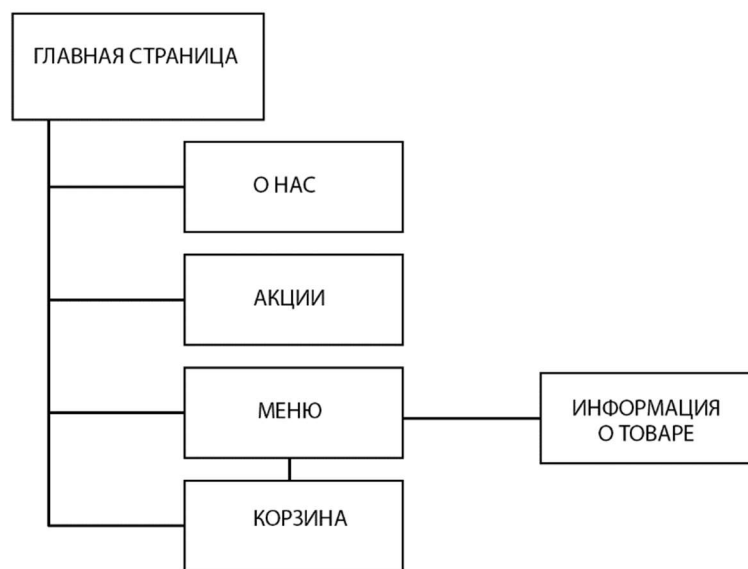


Рисунок 1 – Схематический вид структуры сайта

На главной странице отображается основная информация сайта.

Страница «О НАС» содержит в себе все данные о сайте и магазине.

Страница «АКЦИИ» содержит информацию о скидках и акциях для потенциальных покупателей.

Страница «МЕНЮ» содержит все блюда, которые предлагает магазин быстрого питания.

Страница «ИНФОРМАЦИЯ О ТОВАРЕ» содержит подробную информацию о выбранном блюде.

Страница «КОРЗИНА» отображает позиции, которые пользователь добавил в корзину и форму для оформления заказа.

Переход между страницами осуществляется с помощью гиперссылок.

2.2 Процесс разработки клиентской части сайта

Имея в арсенале знания языков HTML, CSS, JavaScript и PHP вполне возможно создать сайт без использования готовых шаблонных решений, конструкторов и CMS. Подобная работа не только интереснее для программиста, но и явно демонстрирует навыки программирования. Когда потенциальный работодатель ознакомится с портфолио разработчика, он сможет четко определить, что данные технологии и языки действительно освоены, а не поверхностно изучены.

После изучения структуры сайта можно приступить к созданию дизайна и вёрстке. Для создания макета сайта «Burger City» была использована web-версия приложения Figma (Рисунок 2). С готовым макетом гораздо проще и удобнее верстать.

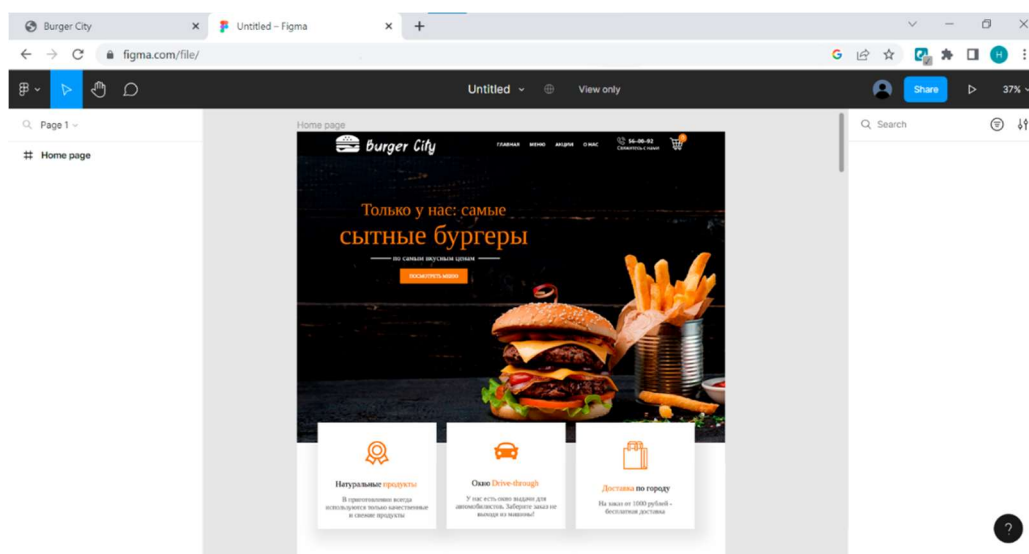


Рисунок 2 – Дизайн макет

При создании сайта нужно уделить внимание не только контенту, графике, изображениям и структуре. Важно также составить гармоничную композицию. Композиция – это грамотно организованное пространство, где все элементы смотрятся целостно, гармонично взаимодействуют друг с другом и с пользователем. Наш мозг привык моментально считывать информацию, поэтому важно преподнести ее правильно, удобно. Хорошо выстроенная логическая

компоновка элементов приятна для восприятия, а также помогает вести взгляд пользователя в нужном направлении и качественно с ним взаимодействовать.

В веб-дизайне есть несколько основных принципов для создания грамотного дизайна. Один из них: направление взгляда. При построении композиции UI-дизайнерами используются два паттерна направления взгляда пользователя: Z и F-образные. В этой работе страницы имеют F-образную композицию. В таком случае сначала считывается левая сторона вертикально, потом по горизонтали, причем верхние элементы запоминаются хорошо, а нижние уже не так значимы.

Человек сканирует страницу сверху вниз слева направо. Это связано с тем, что в большинстве стран мира люди пишут и читают любой текст сверху вниз слева направо, нам привычен такой паттерн. Если размещать контент в несколько колонок, при этом в левой колонке будут заголовки, ключевые тезисы или важные функциональные элементы (например, навигация, фильтрация) – можно существенно упростить жизнь пользователю и повысить удобство ресурса [10].

Для точек фокусировки (якорных объектов) был выбран цвет ярко-оранжевый цвет #FF7400. Он отлично гармонирует с фотографиями блюд магазина, не нарушая, а только дополняя композицию.

Верстка сайта - это процесс создания веб-страницы или целого сайта с использованием языков разметки, таких как HTML, CSS и JavaScript. Верстка включает в себя структурирование информации на странице с помощью HTML-тегов, определение визуального оформления с помощью CSS и программирование динамического поведения с помощью JavaScript [11].

Почти во всех современных многостраничных сайтах есть такие разделы, как «шапка» и «подвал». Шапка и подвал сайта – это части веб-страницы, которые находятся вверху и внизу страницы соответственно и которые отображаются на всех страницах сайта.

Шапка сайта обычно содержит логотип или название компании, навигационное меню, контактную информацию и другие элементы, которые

помогают пользователям быстро ориентироваться на сайте и переходить на нужные страницы.

Подвал сайта, с другой стороны, обычно содержит дополнительную навигацию, ссылки на социальные сети, контактную информацию, копирайты и другие элементы, которые помогают пользователям получить дополнительную информацию о сайте и связаться с его владельцами.

Для сайта этой работы были сделаны шапка и подвал, которые будут одинаково отображаться на каждой странице сайта. Для шапки было задано позиционирование "fixed". Это позволяет закрепить ее на экране и не давать ей прокручиваться вместе с остальным содержимым страницы. Это удобно, когда вы хотите, чтобы ваша шапка всегда была видна пользователю независимо от того, какую часть страницы он просматривает.

Верстать шапку и подвал для каждой странички сайта заново не приходится. HTML-документ был поделён на части, или компоненты, которые подключаются с помощью PHP-кода. В папке components есть несколько php-файлов, в том числе header.php, footer.php и main.php. Чтобы отобразилась главная страница сайта, в файле index.php следующий код:

```
require('components/header.php');  
require('components/main.php');  
require('components/footer.php');
```

Этот код возвращает браузеру header, main и footer по порядку, а в них части html-документа с php-вставками. Для страницы «меню» вместо файла main.php подключается menu.php, а ссылки на подвал и шапку остались не изменны, т.е. подключаются те же самые header.php и footer.php. Таким образом, эти компоненты достаточно сверстать один раз, а не повторять их в других файлах снова и снова. Если захочется что-то исправить: достаточно внести изменения в этот файл и они вступят в силу на всех страницах. Чтобы не ломать структуру HTML-документов, тег <head> и метаинформация хранятся в шапке, а она всегда подключается первой; подключение скриптов и закрывающие теги

</body>, </html> соответственно находятся в подвале, он подключается позже остальных элементов.

На все интерактивные объекты (кнопки, ссылки) были наложены визуальные эффекты. Например, на кнопки шапки был наложен hover, который по наведению мыши меняет цвет элемента (Рисунок 3). Hover является CSS-псевдоклассом, который применяется к элементу при наведении на него курсора мыши. Это позволяет изменять стили элемента в зависимости от того, находится ли курсор над ним или нет.



Рисунок 3 – Изменение цвета элемента по наведению курсором мыши

На главной странице находится информация о данном магазине, в том числе график работы и адрес. С помощью сервиса «Яндекс Карты», на сайт можно добавить интерактивную карту с отметкой, где находится заведение. (Рисунок 4). Это красиво выглядит и наглядно показывает потенциальным покупателям местоположение магазина быстрого питания.

Мы находимся по адресу:
г. Благовещенск,
ул.Театральная, 246а

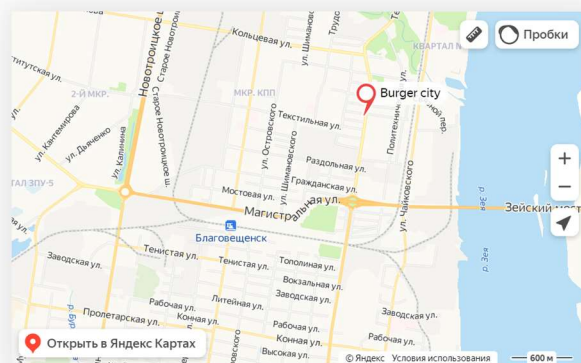


Рисунок 4 – Интерактивная карта с отметкой Burger city

На странице «Меню» отображаются все блюда, которые предлагает автокафе. Они выглядят в виде карточек товаров. Каждый товар отображается в виде отдельной карточки, которая имеет свои картинку, название и цену. Это видно на Рисунке 5. Карточки гибко позиционируются с помощью `display: flex` [6]. Справа страницы есть фильтр, выглядящий как список чек-боксов, который предлагает отметить интересующие категории блюд, чтобы отобразить только их.

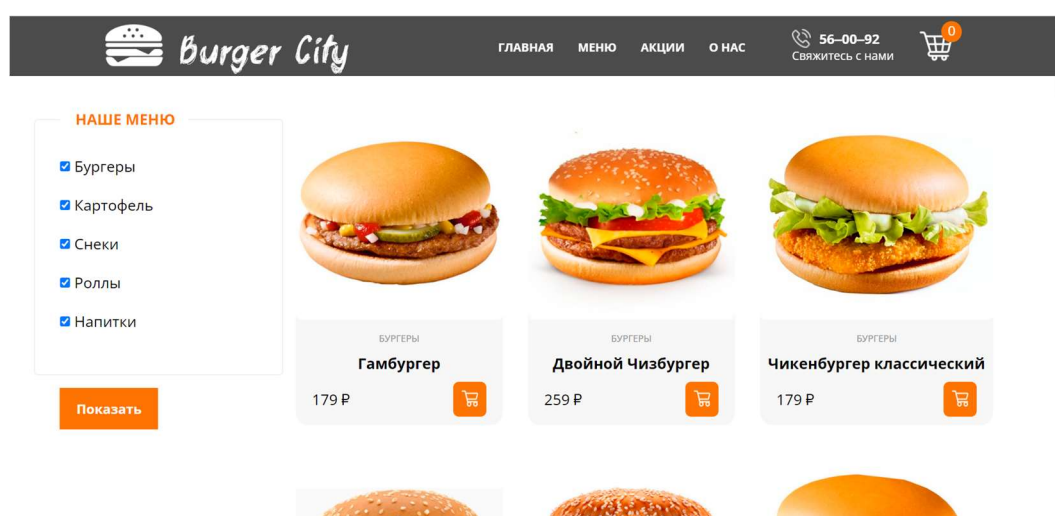


Рисунок 5 – Страница «Меню»

При клике по названию блюда открывается страничка с более подробным описанием (Рисунок 6). Тут покупатель может ознакомиться с составом и добавить товар в корзину или вернуться в меню. К каждому товару генерируется своя страничка, а данные подставляются `php`-вставками. Элемент, который ведёт пользователя на эту страничку тоже имеет в себе ссылку с динамической `php`-вставкой с информацией о `id` товара. При генерировании странички `php`-код смотрит информацию о ссылке в адресной строке, и на её основе возвращает нужные данные.

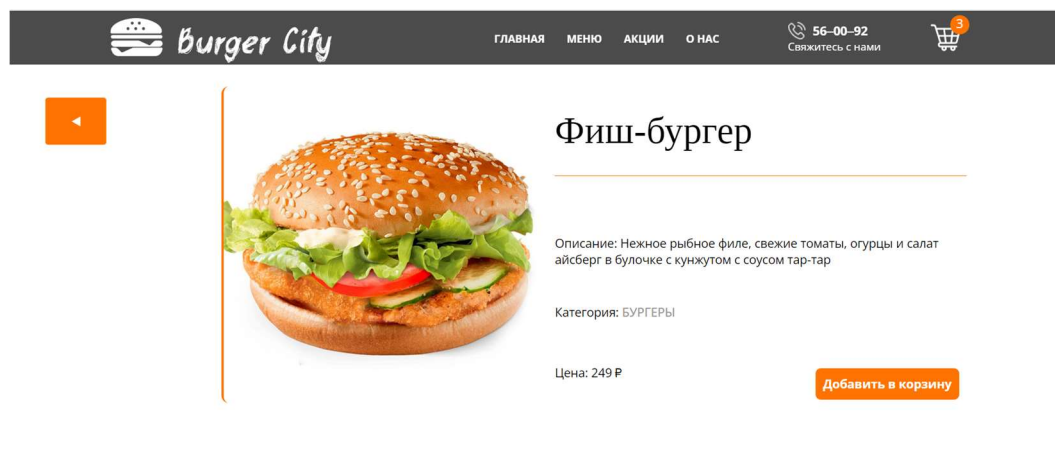


Рисунок 6 – Страничка товара

Когда покупатель добавляет товар в корзину, это сразу отражается в шапке сайта. При этом, с помощью js-скриптов позиции в корзине запоминаются, и отображаются при переходе на другие страницы. Страничку корзины, в которую добавлены несколько товаров можно посмотреть на Рисунке 7, а если корзина пуста – страничка выглядит как на Рисунке 8.

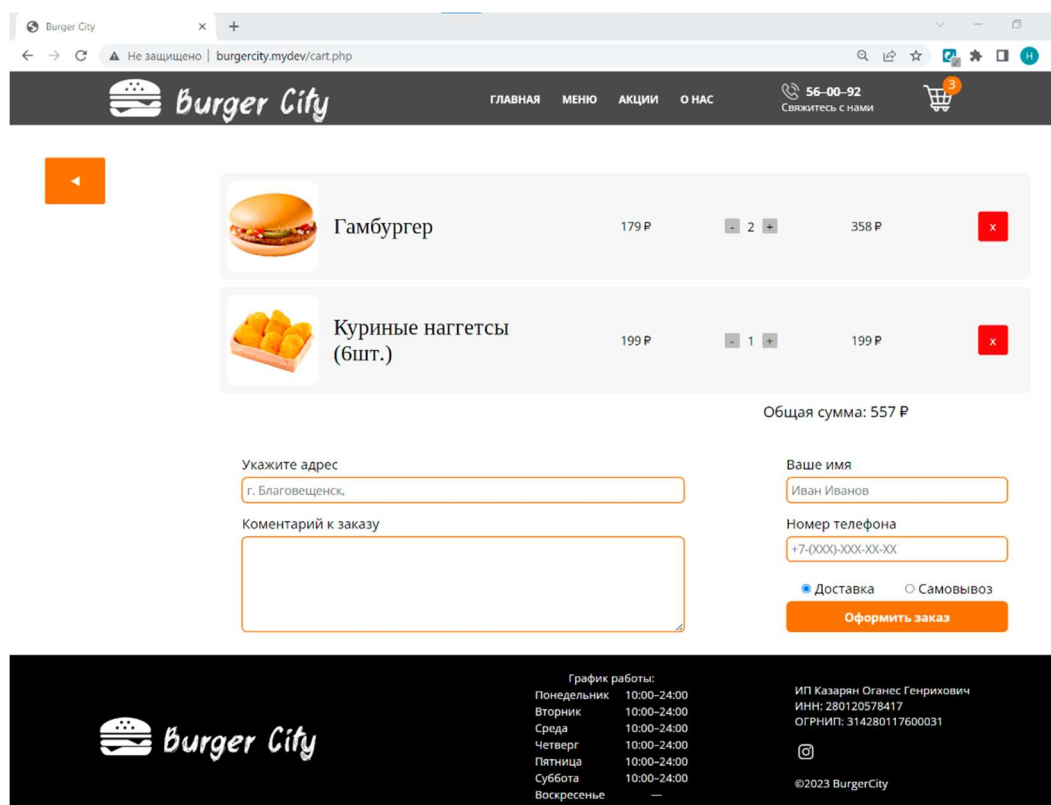


Рисунок 7 – Страничка не пустой корзины

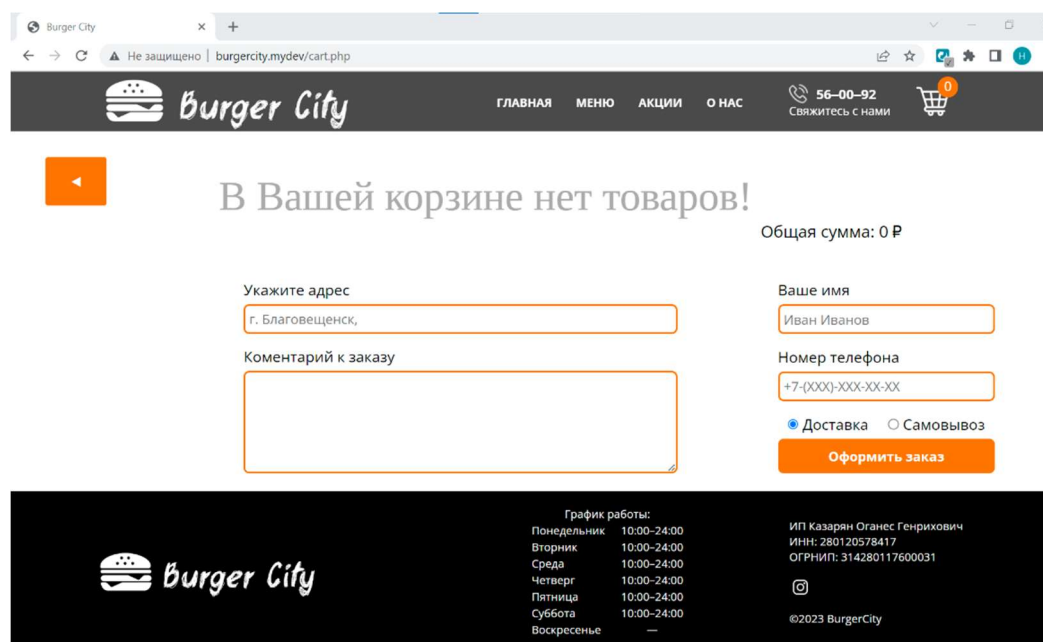


Рисунок 8 – Страничка пустой корзины

На страничке отображаются товары, добавленные в корзину (их названия, изображения, цены), а также количество и стоимость товаров одной позиции и общая сумма заказа. Если товаров нет, пользователь видит соответствующую надпись и может вернуться в меню. После списка товаров находится форма для оформления заказа: в неё необходимо указать адрес, имя, телефон.

По мимо тегов лейблов формы клиенту помогают ориентироваться плейсхолдеры. Placeholder – это вспомогательный текст, который располагают в форме ввода данных. Его главная задача заключается в том, чтобы подсказать пользователю, какую информацию нужно указать в конкретном поле. [25].

С помощью радиокнопок покупатель может указать, хочет он заказать доставку, или может быть сделать предзаказ и забрать готовые блюда самостоятельно.

Так же для заказа предусмотрен комментарий, в котором покупатель может указать любую информацию, например, «Приготовьте пожалуйста без горчицы. У меня на неё аллергия» или «Я хочу приехать за заказом к 19:00, и сделайте пожалуйста по острее». В случае каких-либо непонятных ситуаций, сотрудники могут позвонить по указанному номеру и уточнить все детали.

2.3 Адаптивность сайта

Современный мир информации требует от сайта соответствовать всем устройствам и экранам, на которых его могут просматривать пользователи. Таким образом, важной характеристикой сайта становится его отзывчивость и адаптивность [23].

Во-первых, отзывчивый и адаптивный сайт улучшает пользовательский опыт. Когда пользователь открывает сайт на своем смартфоне или планшете, он хочет видеть информацию, выровненную по центру экрана, легко найти нужную ему информацию, удобно просмотреть фото и видео материалов. Также, отзывчивая и адаптивная вёрстка позволяет быстро загружать контент, что повышает уровень удовлетворённости пользователя.

Во-вторых, отзывчивость и адаптивность веб-сайта позволяют ему эффективно удерживать свою зрительскую аудиторию и снижать показатели отказов. Если сайт не адаптирован под разные экраны и не отображается корректно на мобильных устройствах, то он теряет пользователей, которые также потеряют интерес к продукту и могут покинуть его в поисках других вариантов.

В-третьих, адаптивность и отзывчивость сайта повышают его индексацию в поисковых системах. Многие поисковые системы, такие как Google, предпочитают сайты, отвечающие требованиям мобильной экосистемы и присваивают более высокие рейтинги, что, в свою очередь, помогает сайту привлекать новых посетителей.

В-четвертых, создание отзывчивого и адаптивного сайта может свести к минимуму затраты на разработку. Разработка сайта на стартовых этапах может полностью определить стили и возможности будущего сайта, тем самым пресекая дополнительные затраты на дополнительную разработку или пересмотр уже готовой вёрстки.

Итак, адаптивная вёрстка позволяет создать сайт, способный динамично меняться в зависимости от устройства, на котором его открывают, и размера его экрана. С помощью адаптивных сеток возможно сделать дизайн, который легко перестраивается при необходимости.

Один из самых удобных методов – создание адаптивных сеток с помощью медиавыражений в CSS.

Медиавыражения – это способ, который помогает определить в стилях характеристики браузера или предпочтения пользователя, а затем применить стили или запустить другой код на основе этих параметров.

То есть на экране смартфона сайт будет выглядеть иначе, чем на экране ноутбука, чтобы пользователю было комфортно взаимодействовать с интерфейсом. Медиавыражения могут быть основаны на ширине экрана, ориентации устройства, настройки операционной системы и многом другом.

Любое медиавыражение начинается директивы `@media`, её указание необходимо, чтобы выражение заработало.

В скобках указывают характеристики – условия, при которых данное медиавыражение меняет вид сайта соответственно указанным свойствам.

```
@media (min-width: 768px){  
  .user-list {  
    /* стили для разрешения от 768px */  
  }  
}
```

В данном примере указано, что если минимальная ширина экрана браузера равна или больше 768px, то нужно применить определенные стили к элементам с классом "user-list". Таким образом можно настроить отображение сайта для разных устройств или положений экрана.

Директива `@media (orientation: portrait) {...}` указывает на то, что следующий блок CSS-кода будет применяться только при определенных условиях - в данном случае, когда ориентация экрана является портретной. Весь CSS-код, написанный внутри фигурных скобок {...}, будет применяться только в случае, если экран находится в вертикальной ориентации.

Такое условие может быть полезным для адаптивности макета сайта для мобильных устройств, когда некоторые блоки или элементы могут отображаться более оптимально в портретной ориентации, чем в ландшафтной.

На Рисунке 9 показано, как откроется страничка «корзина» на экране планшета.

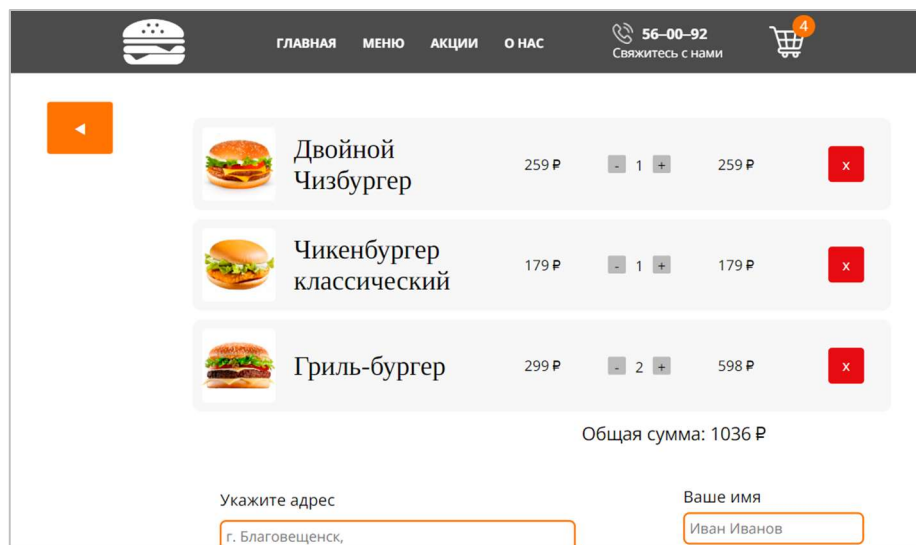


Рисунок 9 – Страничка «Корзина» на экране планшета

На Рисунке 10 показано, как откроется страничка «корзина» на экране смартфона.

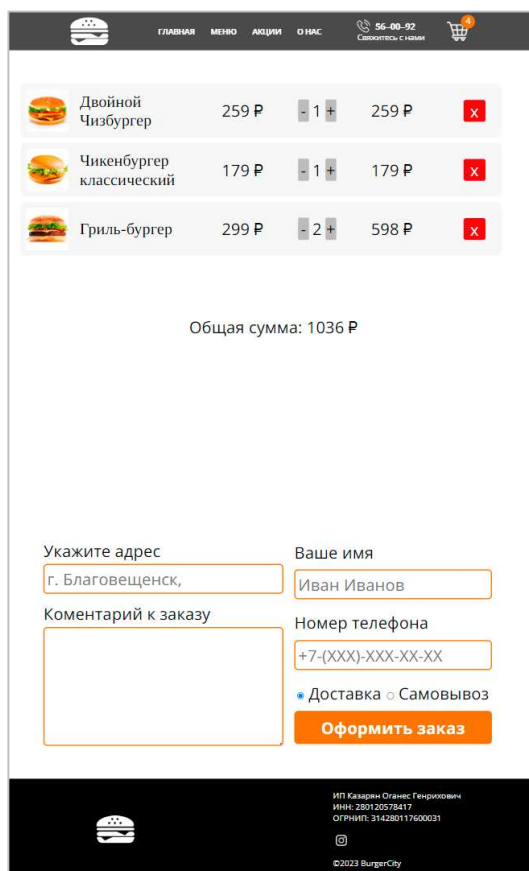


Рисунок 10 – Страничка «Корзина» на экране смартфона

2.4 Процесс создания и подключения базы данных

Чтобы с товарами можно было легко работать, не меняя вёрстку сайта, вся информация берётся из базы данных MySQL. Это даёт возможность добавлять, убирать и редактировать позиции меню как с помощью языка SQL, так и с помощью веб-приложения RHPMyAdmin. База данных «burgercity» содержит в себе таблицу «goods» (с английского – товары). В этой таблице каждый товар имеет свой неповторяющийся id и предусмотренную информацию. Скриншот части этой таблицы представлен на Рисунке 11. Столбец 'category' указывает категорию блюда, 'name' – название, 'price' – цену, 'image' – путь к картинке, а 'description' – описание.

	id	category	name	price	image	description
<input type="checkbox"/>  	1	Бургеры	Гамбургер	179	img/menu/1.jpg	Котлета из натуральной цельной говядины, заправлен...
<input type="checkbox"/>  	2	Бургеры	Двойной Чизбургер	259	img/menu/2.jpg	Две говяжьи котлетки, двойная порция сыра, салатны...
<input type="checkbox"/>  	3	Бургеры	Чикенбургер классический	179	img/menu/3.jpg	Куриная котлетка, Салатный лист, соусы "Томатный" ...
<input type="checkbox"/>  	4	Бургеры	Таёжный	279	img/menu/4.jpg	Сочная говяжья котлета, двойная порция сыра, лук, ...
<input type="checkbox"/>  	5	Бургеры	Гриль-бургер	299	img/menu/5.jpg	Говяжья котлета, двойная порция помидоров, тройная...
<input type="checkbox"/>  	6	Бургеры	Двойной Гамбургер	259	img/menu/6.jpg	Две говяжьи котлеты, двойная порция сыра, солёные ...
<input type="checkbox"/>  	7	Бургеры	Чикенбургер Большой	229	img/menu/7.jpg	Большая куриная котлетка, салатный лист, соусы "То...
<input type="checkbox"/>  	8	Бургеры	Фиш-бургер	249	img/menu/8.jpg	Нежное рыбное филе, свежие томаты, огурцы и салат ...
<input type="checkbox"/>  	9	Бургеры	Сытный	299	img/menu/9.jpg	Два говяжьих рубленых бифштекса, приправленные пря...
<input type="checkbox"/>  	10	Картофель	Картофель-Фри маленький	159	img/menu/10.jpg	Классический подсолённый картофель-фри 140 гр.
<input type="checkbox"/>  	11	Картофель	Картофель-Фри средний	179	img/menu/11.jpg	Классический подсолённый картофель-фри 180 гр.
<input type="checkbox"/>  	12	Картофель	Картофель-Фри большой	219	img/menu/12.jpg	Классический подсолённый картофель-фри 240 гр.
<input type="checkbox"/>  	13	Картофель	Картофель по-деревенски	229	img/menu/13.jpg	Хрустящие золотистые дольки картофеля в коже с а...
<input type="checkbox"/>  	14	Картофель	Картофельные дипперы	229	img/menu/14.jpg	Картофельные дольки продолговатой формы, обжаренны...
<input type="checkbox"/>  	15	Снеки	Куриные наггетсы (6шт.)	199	img/menu/15.jpg	Нежнейшие кусочки мяса птицы в золотистой панировк...
<input type="checkbox"/>  	16	Снеки	Куриные наггетсы (9шт.)	249	img/menu/16.jpg	Нежнейшие кусочки мяса птицы в золотистой панировк...
<input type="checkbox"/>  	17	Снеки	Сырные палочки	199	img/menu/17.jpg	Хрустящие палочки из Моцареллы, обжаренные во фрит...

Рисунок 11 – часть таблицы «goods»

База данных была создана с помощью SQL. Вот код, который создаёт таблицу «goods»:

```
CREATE TABLE IF NOT EXISTS goods
(
    id int NOT NULL AUTO_INCREMENT,
    category varchar(255) NOT NULL,
```

```
name varchar(255) NOT NULL,  
price varchar(255) NOT NULL,  
image varchar(255) NOT NULL,  
description varchar(255) NOT NULL,  
PRIMARY KEY(id)  
) CHARSET=utf8
```

Таблица goods будет иметь шесть столбцов:

- id – уникальный идентификатор (ключ), который будет автоматически генерироваться для каждой новой записи в таблице.
- category – категория товара. Это строковый столбец с максимальной длиной 255 символов.
- name – наименование товара. Это строковый столбец с максимальной длиной 255 символов.
- price – цена товара. Это строковый столбец с максимальной длиной 255 символов.
- image – локальная ссылка на изображение товара, путь к файлу. Это строковый столбец с максимальной длиной 255 символов.
- description – описание товара. Это строковый столбец с максимальной длиной 255 символов.

Ключевым столбцом является id, который объявлен как первичный ключ таблицы. Он генерируется автоматически для каждой новой записи в таблице и представляет собой уникальный идентификатор товара. Целость данных в таблице гарантирована благодаря уникальным значениям ключа.

Так же указывается кодировка (CHARSET) для таблицы, в данном случае – utf8. Это обеспечивает правильное отображение символов на разных языках, таких как русский, китайский и т.д [5].

Таблица была заполнена с помощью таких конструкций:

```
INSERT INTO goods ('category', 'name', 'price', 'image', 'description') VALUES
```

('Бургеры', 'Гамбургер', '179', 'img/menu/1.jpg', 'Котлета из натуральной цельной говядины, заправленная горчицей и кетчупом, луком, маринованным огурчиком в карамелизованной булочке.'),

('Бургеры', 'Двойной Чизбургер', '259', 'img/menu/2.jpg', 'Две говяжьи котлетки, двойная порция сыра, салатный лист, сочный помидор и соус \"Цезарь\" на булочке с кунжутом.')

Далее с помощью PHP-кода, а именно расширения PDO можно подключиться к базе данных и записать информацию в переменную [3].

PDO или PHP Data Objects – это расширение PHP, которое предоставляет унифицированный интерфейс для доступа к различным базам данных. Оно позволяет использовать одни и те же методы и функции для работы с различными СУБД (MySQL, PostgreSQL, SQLite и др.). PDO обеспечивает безопасность и защиту от SQL-инъекций, а также уменьшает объем кода благодаря использованию подготовленных запросов [7].

Для заказов была создана ещё одна таблица в базе данных. Она называется «orders» и хранит в себе id заказа, имя и телефон покупателя, указанный покупателем адрес, способ получения заказа и комментарий к заказу. Информация в неё загружается не статично, а динамически. Данные передаются в файл sendmail.php методом POST, а оттуда с помощью расширения PDO добавляются в базу данных [4].

Благодаря этому коду создаётся и запускается SQL-запрос с актуальной информацией о заказе:

```
$PDO = PDOConnect::getInstance();  
$json = json_decode($_POST['cart']);  
$sqlAdd = "  
INSERT INTO orders  
SET buyer = :buyer,tel = :tel,address = :address,delivery = :delivery,note =  
:note,card = :json  
";  
$set = $PDO->PDO->prepare($sqlAdd);
```

```

$set->bindValue(':buyer', $_POST['buyer']);
$set->bindValue(':tel', $_POST['tel']);
$set->bindValue(':address', $_POST['address']);
$set->bindValue(':delivery', $_POST['delivery']);
$set->bindValue(':note', $_POST['note']);
$set->bindValue(':json', json_encode($json));
$set->execute();

```

Таким образом, в базе данных «burgercity» есть две таблицы: «goods» – для хранения информации о товарах, предлагаемых пользователям на сайте, и «orders» – для хранения информации о заказе.

Вот код, который выполнит запрос «SELECT * from `goods`» и поместит результат в массив \$products. Таким образом получится двумерный массив, т.е. каждая ячейка (товар) будет хранить в себе другой ассоциативный массив (информацию о товаре):

```

$PDO = PDOConnect::getInstance();
$result = $PDO->PDO->query(" SELECT * from `goods` ");
$products = array();
while ($productInfo = $result->fetch()){
    $products[] = $productInfo;
}

```

Такой подход позволяет сделать странички сайта не статическими, а динамическими: т.е. карточки не вписаны в html код, а генерируются. Все товары хранятся в базе данных, если добавить товар в базу данных – он отобразится на сайте. Это происходит с помощью PHP-вставок в HTML-разметку, например:

```

<div class="cards">
    <?foreach ($products as $product):?>
        <div class="card" data-id="<?=$product['id']?>">
            <div>content</div>
            <p class='name'><?=$product['name']?>"></p>
        </div>
    </foreach>
</div>

```

```
<?endforeach?>  
</div>
```

Этот код отображает набор карточек (div с классом "card") для каждого продукта из массива \$products. Каждая карточка имеет атрибут "data-id" со значением id продукта и блок-элемент с классом "name", отображающий имя продукта.

Цикл foreach проходит по каждому элементу массива \$products и создает разметку HTML для каждого элемента. Это достигается путем помещения отдельных элементов разметки HTML внутри цикла, с использованием тегов для интеграции PHP-кода [26].

В результате, для каждого продукта из массива \$products выводится карточка с его именем и идентификатором. Подобный код, только расширенный, с большим количеством php-вставок, используется для отображения списка продуктов на веб-странице.

По такому же принципу генерируются и пункты фильтра категории. Если в базе данных появится какой-то товар с новой категорией, будет автоматически создан новый чек-бокс, такой же функциональный как остальные.

2.5 Процесс разработки скриптов для сайта

Скрипты для сайта разрабатываются для добавления функциональности сайту, создания интерактивных элементов, повышения производительности и удобства использования сайта для пользователей [24].

Для начала нужно сделать корзину, чтобы пользователь мог добавить в неё товары. Существует несколько способов реализовать корзину на сайте. Если сайт одностраничный – достаточно будет поверхностных знаний JavaScript. Однако, этот сайт является многостраничным, и конечно подразумевается, что при перезагрузке странички или при переходе из страницы меню в страницу корзины данные не будут утеряны. Соответственно, их нужно где-то хранить, причём не на сервере, а на клиентской стороне. Это можно сделать разными способами. Один из самых простых из них – хранить информацию в LocalStorage.

LocalStorage – это механизм хранения данных прямо в браузере, который позволяет сохранять информацию на стороне клиента в формате ключ-значение. Хранилище доступно только для текущего домена, то есть другие сайты не могут получить доступ к этим данным. Это позволяет использовать LocalStorage для хранения конфиденциальных данных, таких как логины, пароли, персональная информация пользователя. Данные, сохраненные в LocalStorage, сохраняются после закрытия браузера и даже после перезагрузки компьютера, что делает их полезными для сохранения предпочтений пользователя, настроек или результатов работы приложений на стороне клиента [18].

Чтобы получить данные cart из LocalStorage используется конструкция `window.localStorage.getItem('cart')`, а чтобы записать данные – конструкция `window.localStorage.setItem('cart')`. При этом нужно учитывать, что данные в LocalStorage хранятся в виде ключ-значение, а значение в виде строки. Поэтому нужно не забывать про необходимость преобразовывать объект cart в строку, и из строки в объект. В это помогут функции `JSON.parse()` и `JSON.stringify()` соответственно, аргументом в которые мы передаём необходимую переменную [19].

Когда открывается страничка сайта, первым делом проверяется, есть ли что-то в корзине, и, если есть – помещается в объект cart для дальнейшей работы. Cart хранит в себе id добавленного в корзину товара в качестве ключа, и количество в качестве значения. Функция `sumCount()` считает сумму значений всех ключей, т. е. общее количество товаров в корзине и с помощью функции `changeCount()` эта информация изменяется в шапке сайта.

На каждой кнопке «добавить в корзину» с помощью цикла был установлен слушатель события, который по щелчку на кнопку добавит значение data-id кнопки в cart и установит значение 1, если товара не было или увеличит значение на 1 если товар уже был. Data-id генерируется PHP-кодом с помощью id из базы данных, соответственно каждая кнопка «добавить в корзину» отличается своим значением data-id, и в корзину будет добавлен товар с нужным id [1].

Страница «Корзина» отображает товары, которые были добавлены и скрывает остальные. Если в cart нет позиций, будет показан текст «В Вашей корзине нет товаров!», а если какие-то позиции есть – этот текст будет скрыт.

Это происходит путём добавления или удаления класса у элемента с помощью JavaScript-кода. В CSS заранее прописаны свойства для класса hidden { display: none }, который полностью скрывает элемент со страницы. С помощью JavaScript можно добавить или убрать класс hidden указанным элементам сайта.

Если же товары в корзине есть, на странице «Корзина» каждый из них отображается и имеет кнопки для увеличения или уменьшения количества, и кнопку товара из корзины. На каждой кнопке свой слушатель событий, который по нажатию выполнит соответствующий код для соответствующего товара. Если количество товара становится меньше 1, он автоматически удаляется из корзины.

Так же с помощью JavaScript-кода можно умножить цену товара на количество и вывести это значение, и посчитать общую сумму заказа. Если элементы уже были сгенерированы в DOM-дереве, получить доступ к их текстовому содержимому можно с помощью метода textContent.

Внизу странички «Корзина» есть форма, которую покупатель должен заполнить чтобы оформить заказ. Поле «Комментарий к заказу» не является обязательным. Однако получить номер телефона клиента всё-таки важно. Поэтому в HTML-разметке указан атрибут required. Он указывает на то, что данный элемент HTML формы должен быть обязательно заполнен. Атрибут следует применять к тегам input или textarea. При попытке отправки формы в том случае, если поле с этим атрибутом не заполнено – браузер не даст отправить форму и выведет ошибку в виде всплывающей подсказки.

Когда покупатель заполнил форму и нажимает кнопку «Оформить заказ» - все данные формы и корзина из localStorage отправляются скриптом order.js в файл sendmail.php методом POST, а оттуда с помощью расширения PDO добавляются в таблицу «orders» базы данных «burgercity» и на электронную почту функцией mail().

Вот наглядный пример, как это всё работает: допустим покупатель определился с блюдами и заполнил форму, как на Рисунке 12.

Burger City ГЛАВНАЯ МЕНЮ АКЦИИ О НАС 56-00-92 Связаться с нами

Гамбургер 179 Р 2 358 Р

Куриные наггетсы (бшт.) 199 Р 1 199 Р

Латте 189 Р 2 378 Р

Общая сумма: 935 Р

Укажите адрес: г. Благовещенск, ул. Калинина 75, 408

Ваше имя: Зайцев Никита

Коментарий к заказу: Пусть курьер позвонит, я выйду и заберу заказ.

Номер телефона: +7 914 382 04 25

☒ Доставка ☐ Самовывоз

Оформить заказ

Рисунок 12 – Заполненная форма

Покупатель нажмёт кнопку «оформить заказ», если всё успешно – на экране появится уведомление о том, что заказ отправлен. Тем временем на почте появится письмо, которое будет выглядеть как на Рисунке 13.

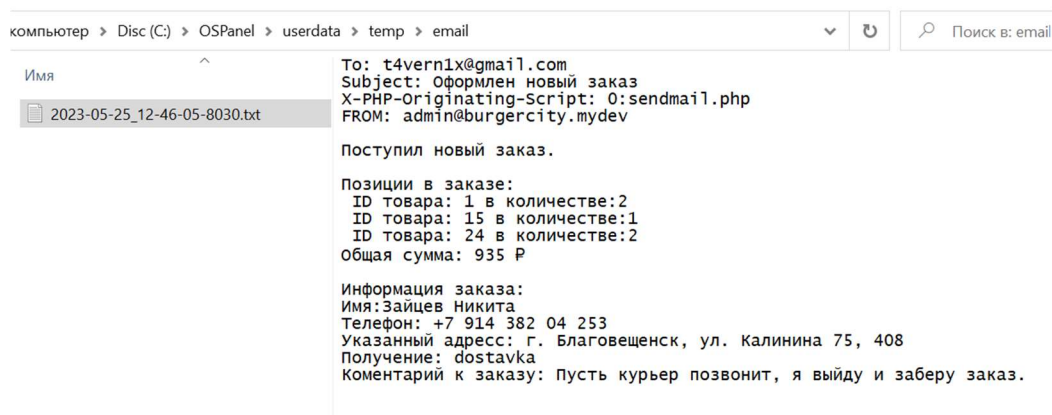


Рисунок 13 – Запись о заказе на почте

Если всё прошло без ошибок, то покупатель увидит уведомление alert() о том, что заказ отправлен. В таблице «orders» базы данных «burgercity» появится новая запись, как показано на Рисунке 14.

	id	buyer	tel	address	delivery	note	cart
<input type="checkbox"/>	3	Зайцев Никита	+79143820425	г. Благовещенск, ул. Калинина 75, 408	dostavka	Пусть курьер позвонит, я выйду и заберу заказ.	["1":2,"15":1,"24":2]

Рисунок 14 – Запись о заказе в базе данных

С помощью LocaleStorage происходит запоминание выбранных фильтров на странице меню. Пользователь может отметить, какие категории он хочет посмотреть и нажать кнопку «Показать», на которой висит слушатель событий. По щелчку на кнопку выбранные флажки будут записаны в LocaleStorage со значением True, а не выбранные со значением False.

Фильтры меню выглядят как список чек-боксов. Флажок, или флаговая кнопка, или чекбокс (от англ. check box), галочка – это элемент графического пользовательского интерфейса, позволяющий пользователю управлять параметром с двумя состояниями: ☒ включено и ☐ отключено [17].

При нажатии на кнопку «Показать» происходит перезагрузка страницы. У каждой карточки товара проверяется категория. Если данная категория в localeStorage имеет значение «false», то карточка получает класс hidden и соответственно скрывается. На самом деле её код остаётся на странице, только он не отображается.

При загрузке страницы сразу проверяется, были ли установлены какие-то фильтры. Если информации о фильтрах в LocaleStorage нет, то она создаётся со значениями True на каждую категорию, соответственно пользователь увидит все товары магазина.

На каждой карточке товара и на страничке товара есть ссылка на категорию данного блюда. Она имеет отличающийся от основного текста цвет, чтобы показать пользователю, что он может на неё нажать, и соответственно

посмотреть товары этой категории. При нажатии на неё, в LocaleStorage будет изменена информация о фильтрах: именно эта категория будет включена, а все остальные выключены. Затем откроется страница меню с применёнными фильтрами, чтобы пользователь мог посмотреть товары данной категории.

На Рисунке 15 изображена страница меню с установленными и применёнными фильтрами.

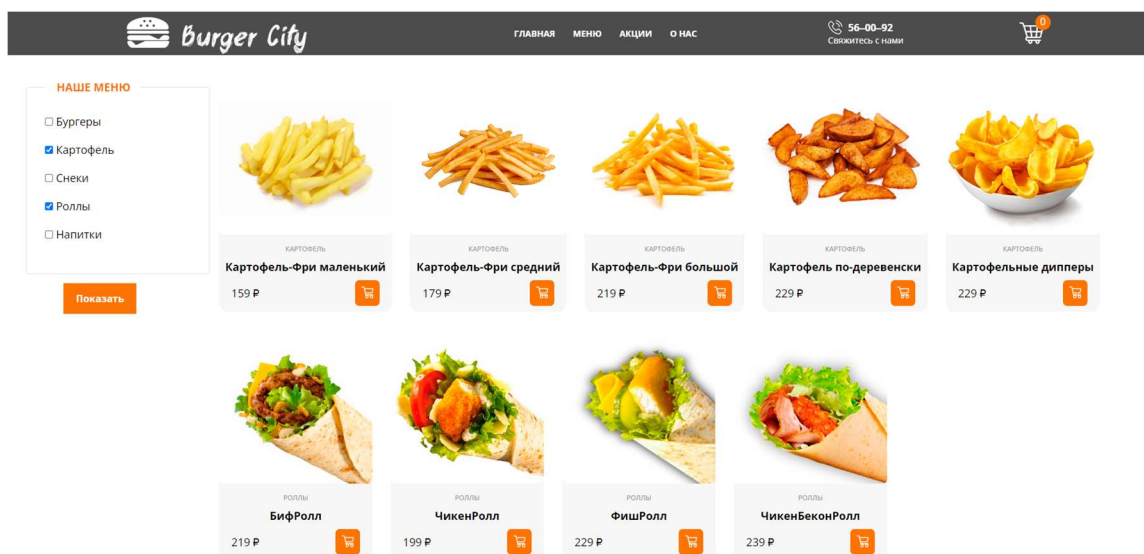


Рисунок 15 – Меню с фильтрами «Картофель» и «Роллы»

ЗАКЛЮЧЕНИЕ

Результатом проведённой работы является разработанный веб-сайт, предназначенный для автоматизации процесса ознакомления с меню и оформления заказа.

В ходе исследования были изучены основные принципы и подходы к созданию функционального и привлекательного сайта. Проанализированы современные технологии и инструменты веб-разработки, из них применены наиболее подходящие для данного проекта.

Следующим этапом стало проектирование структуры сайта и создание макета. Были изучены основные принципы и важные аспекты проектирования композиции веб-страниц. Дизайн сайта учитывает современные требования отображения контента. Навигация по сайту проста для пользователя. Наиболее оптимальная цветовая гамма и красивые кнопки взаимодействия улучшают процесс просмотра, навигации и общения с интерфейсом.

На следующем этапе создавалась вёрстка сайта по готовому макету. Была написана семантически правильная разметка веб-страниц. Были добавлены стили, чтобы сайт соответствовал дизайн-макету. Применены технологии, делающие внешний вид интерактивных элементов отзывчивыми на поведение пользователя.

Была создана база данных для сайта. В неё были добавлены позиции меню магазина. Реализованы алгоритмы генерации HTML-документов с информацией, получаемой из базы данных.

Следующим этапом стало создание скриптов сайта. Был разработан код для сохранения позиций меню в виртуальную корзину, их редактирования и грамотного отображения. Так же был создан скрипт для отображения товаров по заданным фильтрам на странице меню.

Реализован алгоритм обработки заказа: данные о помещённых в виртуальную корзину товарах и информация, введённая пользователем, сохраняются в базу данных сайта и отправляются на электронную почту.

Программный продукт имеет большие перспективы для будущего развития. Можно дорабатывать текущие функции веб-сайта или же добавлять новые. Актуальность отображаемого контента зависит от актуальности данных, добавленных в базу данных.

Цель, которая была сформирована на начальном этапе разработки веб-сайта, была полностью достигнута. Все задачи, при помощи которых решалась поставленная цель так же были выполнены.

Разработанный веб-ресурс позволит увеличить число клиентов, повысить узнаваемость бренда и расширить географию продаж. Также это даст возможность более эффективно управлять заказами, обрабатывать данные клиентов и запускать рекламную кампанию в интернете. Сайт для магазина быстрого питания – это не просто удобство для клиентов, но и мощное средство для развития бизнеса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Бхаргава, А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих / А. Бхаргава. – Санкт-Петербург : Издательский Дом ПИТЕР, 2022. – 288 с.
- 2 Грофф Д. Р. SQL. Полное руководство / Д. Р. Грофф, П. Н. Вайнберг, Э. Д. Опель. – 3. – СПб. : Диалектика, 2019. – 560 с.
- 3 Документация по языку PHP [Электронный ресурс]. URL: <https://www.php.net/docs.php>
- 4 Дэвид Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов / Д. Дэвид. – Отдельное издание. – Москва : Диалектика-Вильямс, 2020. – 464 с.
- 5 Дэвид, Т. Программист-прагматик / Т. Дэвид, Х. Эндрю. – Москва : Диалектика, 2020. – 368 с.
- 6 Журнал «Доктайп» HTML Academy [Электронный ресурс]. URL: <https://htmlacademy.ru/blog>
- 7 Зандстра М. С. PHP. Объекты, шаблоны и методики программирования / М. С. Зандстра. – Москва : Вильямс, 2020. – 576 с.
- 8 КОД Журнал Яндекс Практикума [Электронный ресурс]. URL: <https://thecode.media>
- 9 Крокфорд Д. Как устроен JavaScript / Д. Крокфорд. – Санкт-Петербург : Питер, 2019. – 304 с.
- 10 Круг, С. Не заставляйте меня думать. Веб-юзабилити и здравый смысл / С. Круг. – Москва : Эксмо, 2021. – 256 с.
- 11 Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл. – СПб : БХВ, 2022. – 896 с.
- 12 Мартин Р. Идеальный программист. Как стать профессионалом разработки ПО / Р. Мартин. – Санкт-Петербург : Питер СПб, 2019. – 224 с.
- 13 Мартин, Р. Чистый код. Создание, анализ и рефакторинг / Р. Мартин. – СПб. : Питер, 2018. – 464 с.

- 14 Правдина, Е. Говорят, в IT много платят / Е. Правдина. – Москва : БОМБОРА, 2022. – 384 с.
- 15 Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера, 5 изд / Н.А. Прохоренок. – СПб. : БХВ-Петербург, 2019. – 912 с.
- 16 Роббинс Д.Н. "HTML5, CSS3 и JavaScript. Исчерпывающее руководство" / Д.Н. Роббинс. – Москва : Эксмо, 2020. – 528 с.
- 17 Робсон, Фримен, Э., Э. Изучаем программирование на JavaScript / Э., Э. Робсон, Фримен. – Санкт-Петербург : Питер, 2022. – 640 с.
- 18 Симпсон, К. {Вы пока еще не знаете JS} Познакомьтесь, JavaScript / К. Симпсон. – Санкт-Петербург : Питер, 2022. – 190 с.
- 19 Современный учебник JavaScript [Электронный ресурс]. URL: <https://learn.javascript.ru>
- 20 Фаулер М. Шаблоны корпоративных приложений / М. Фаулер. – Санкт-Петербург : Диалектика, 2020. – 544 с.
- 21 Флэнаган Д. JavaScript. Полное руководство / Д. Флэнаган. – Москва : Диалектика-Вильямс, 2021. – 720 с.
- 22 Форта Б. Освой самостоятельно SQL / Б. Форта, П. Н. Вайнберг, Э. Д. Опель. – 4-е изд. – М. : И.Д. Вильямс, 2019. – 264 с.
- 23 Фрэйн Б. Отзывчивый дизайн на HTML5 и CSS3 для любых устройств / Б. Фрэйн. – Санкт-Петербург : Питер, 2022. – 336 с. – ISBN 978-5-4461-1495-5
- 24 Хавербеке М. Выразительный JavaScript. Современное веб-программирование / М. Хавербеке. – Санкт-Петербург : Питер, 2019. – 480 с.
- 25 Шмитт К. HTML5. Рецепты программирования. / К. Шмитт, К. Симпсон. – СПб : Питер, 2012. – 288 с.
- 26 HTML Academy: интерактивные онлайн-курсы [Электронный ресурс]. URL: <https://htmlacademy.ru>