

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ИНСТИТУТ БИЗНЕСА»
БЕЛОРУССКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА**

**Факультет бизнеса
Кафедра менеджмента технологий**

Дипломная работа

**РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ ПРЕДПРИЯТИЯ
ОБЩЕСТВЕННОГО ПИТАНИЯ**

ОСТАПЕНКО Алексей Александрович

Научный руководитель:
Кандидат технических наук, доцент
_____И.П. Стацук

Допущен к защите
«_____» _____ 2018 г.

Заведующий кафедрой менеджмента технологий
Кандидат технических наук
_____Ю.Н. Силкович

Минск, 2018

**ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ИНСТИТУТ БИЗНЕСА И МЕНЕДЖМЕНТА ТЕХНОЛОГИЙ»
БЕЛОРУССКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА**

Факультет бизнеса
Кафедра менеджмента технологий

УТВЕРЖДАЮ
Заведующий кафедрой
к. т. н., доцент
_____ Ю. Н. Силкович
« ____ » _____ 2018 г.

ЗАДАНИЕ
на дипломную работу

Студенту Остапенко Алексею Александровичу группа 452 УИР

1. Тема дипломной работы Разработка WEB-приложения для предприятия общественного питания

Утверждена приказом ректора БГУ от 16 января 2018 г. № 25 - ПС

2. Исходные данные к дипломной работе

- а) материалы преддипломной практики;
- б) статистические и экономические материалы по результатам работы предприятия;
- в) научные публикации отечественных и зарубежных авторов;
- г) нормативные законодательные акты Республики Беларусь.

3. Перечень подлежащих разработке вопросов или краткое содержание расчетно-пояснительной записки:

3.1. Теоретические основы разработки и использования веб-сайта и веб-приложения.

3.2. Проектирование веб-сайта и веб-приложения для организации общественного питания «Кофейня «Зерно».

3.3. Разработка веб-сайта и веб-приложения.

4. Перечень графического материала –

5. Консультанты по дипломной работе с указанием относящихся к ним разделов -

6. Примерный календарный график выполнения дипломной работы

- 1. 16 апреля 2018 г. – 30%
- 2. 3 мая 2018 г. – 60%
- 3. 18 мая 2018 г. – 90%

7. Дата выдачи задания 16 января 2018 года.

8. Срок сдачи законченной дипломной работы 14 июня 2018 года.

Руководитель _____ И.П. Стацук

Подпись студента _____

Дата «___» _____ 2018 г.

РЕФЕРАТ

Дипломная работа: 65 с., 35 рис., 8 табл., 19 источник, 3 прил.

КОФЕЙНЯ «ЗЕРНО», БИЗНЕСС-ПРОЦЕСС, АВТОМАТИЗАЦИЯ, РАЗРАБОТКА, ЭФФЕКТИВНОСТЬ

Объект исследования - организация общественного питания «Кофейня «Зерно».

Предметом исследования является разработка и влияние веб-продуктов на эффективность бизнеса.

Цель дипломной работы – разработка методических положений и рекомендаций, а также применение соответствующих инструментальных средств для разработки веб-продукта.

Методы исследования: общенаучные методы исследований: сбор фактов, методы анализа и синтеза, группировки и выборка, сравнения и другие экономико-статистические методы.

Результаты исследования и разработки: разработка веб-продукта для организации общественного питания; оценка экономической эффективности внедрения программного продукта.

Практическая значимость исследования состоит в необходимости автоматизации процессов организации и разработке программного продукта.

Автор работы подтверждает, что приведенный в ней материал правильно и объективно отражает состояние исследование процесса, а все заимствования сопровождаются ссылками на их авторов.

РЭФЕРАТ

Дыпломная работа: 65 с., 35 мал., 8 табл., 19 крыніц, 3 дадатка.

КАФЕЙНЯ «ЗЕРНЕ», БІЗНЭС-ПРАЦЭС, АУТАМАТЫЗАЦЫЯ, РАСПРАЦОЎКА, ЭФЕКТЫЎНАСЦЬ.

Аб'ект даследавання - арганізацыя грамадскага харчавання "Кафейня" Зерне".

Прадметам даследавання з'яўляецца распрацоўка і ўплыў вэб-прадуктаў на эфектыўнасць бізнесу.

Мэта дыпломнай работы - распрацоўка метадычных палажэнняў і рэкамендацый, а таксама прымяненне адпаведных інструментальных сродкаў для распрацоўкі вэб-прадукту.

Метады даследавання: агульнанавуковыя метады даследаванняў: збор фактаў, метады аналізу і сінтэзу, групоўкі і выбарка, параўнанні і іншыя эканоміка-статыстычныя метады.

Вынікі даследавання і распрацоўкі: распрацоўка вэб-прадукту для арганізацыі грамадскага харчавання; ацэнка эканамічнай эфектыўнасці ўкаранення праграмага прадукту.

Практычная значнасць даследавання складаецца ў неабходнасці аўтаматызацыі працэсаў арганізацыі і распрацоўцы праграмага прадукту.

Аўтар працы пацвярджае, што прыведзены ў ёй матэрыял правільна і аб'ектыўна адлюстроўвае стан даследавання працэсу, а ўсе запазычаныя суправаджаюцца спасылкамі на іх аўтараў.

AUBSTRACT

Thesis: 69 p, 35 figures, 8 tables, 19 sources, 3 app.

CAFE "ZERNO" BUSINESS PROCESSES, AUTOMATATION, DEVELOPMENT, EFFICIENCY.

Object of research - organization of public catering Cafe "Zerno".

Subject of research is the development and impact of web products on business performance.

The aim of the thesis - to develop methodical positions and recommendations, as well as the use of appropriate tools to optimize customer relationships.

Methods of research: general scientific methods of research: collection of facts, methods of analysis and synthesis, grouping and sample comparisons and other economic and statistical methods.

The results of research and development: development of a web-based product for catering; Evaluation of the cost-effectiveness of software implementation.

The practical significance of the research consists in the need to automate the processes of organizing and developing a software product.

The author of the work confirms that the above thesis in computational and analytical material correctly and objectively reflects the state of the process under investigation, and all borrowed from sources theoretical, methodological and methodical positions and concepts are accompanied by references to their authors.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ВЕБ-САЙТА И ВЕБ-ПРИЛОЖЕНИЯ	10
1.1.Понятие веб-сайта и веб-приложения.....	10
1.2 Принципы создания веб-сайта и веб-приложения	15
ГЛАВА 2 ПРОЕКТИРОВАНИЕ ВЕБ-САЙТА И ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПРЕДПРИЯТИЯ ОБЩЕСТВЕННОГО ПИТАНИЯ «КОФЕЙНЯ «ЗЕРНО»	23
2.1 Описание основных технико-экономических характеристик предприятия общественного питания «Кофейня «Зерно»	23
2.2 Анализ действующих инструментов для взаимодействия с клиентами	26
2.3 Выбор средств и методов разработки веб-сайта и веб-приложения	29
ГЛАВА 3 РАЗРАБОТКА ВЕБ-САЙТА И ВЕБ-ПРИЛОЖЕНИЯ	42
3.1 Разработка макета сайта и приложения.....	42
3.2 Верстка макетов веб-сайта и веб-приложения.....	53
3.3 Разработка кода веб-приложения.....	55
3.4 Расчет технико-экономической эффективности проекта по внедрению веб-сайта и веб-приложения в организации «Кофейня «Зерно».....	58
ЗАКЛЮЧЕНИЕ.....	62
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	64

ВВЕДЕНИЕ

В современном мире собственное представительство в сети Интернет является большим преимуществом для любой коммерческой и некоммерческой компании. Огромная аудитория всемирной паутины – широкие возможности для развития бизнеса, его продвижения и укрепления на рынке. Сегодня веб-сайт является мощнейшим маркетинговым инструментом, а также позволяет решать проблему автоматизации важнейших бизнес-процессов предприятия. Современные веб-технологии позволяют реализовывать взаимоотношения компании с клиентами, продвижение и развитие бизнеса, исполнять задачи маркетинга, а современные веб-приложения позволяют осуществлять взаимодействия организации с клиентами.

Основной стратегией успешного существования бизнеса является эффективное усовершенствование отношений с клиентами. Ориентация на взаимоотношения с клиентами обусловлена рядом тенденций, таких как усиление конкуренции, повышение требований клиентов к качеству продукции и оказываемых услуг, доступность информации. Знание своих клиентов и удовлетворенность их запросов позволяют организации получить новые возможности для продвижения бизнеса, повышения качества оказываемых услуг, устойчивого развития и укрепления конкурентоспособности на рынке.

В результате исследований использования веб-технологий организациями было выявлено повышение заинтересованности клиентов в оказываемых услугах, а соответственно их взаимодействия и экономических результатов деятельности. В то же время целью маркетинговых усилий различных предприятий и организаций является привлечение клиентов, реализуемое через высокозатратные маркетинговые программы, привлечение специалистов для реализации данных программ. Это требует пересмотра маркетинговой политики и усовершенствование применяемых средств реализации маркетинговой деятельности. Опыт использования веб-технологий обеспечивает повышение эффективности работы организации, уменьшение издержек и улучшение взаимоотношений с клиентами.

Цель дипломной работы состоит в разработке веб-приложения и веб-сайта, их применении для организации и определении показателей использования.

Поставленная цель обусловила решение следующих задач:

- раскрыть теоретические основы веб-сайта и веб-приложения, рассмотреть их функции и возможности;
- исследовать существующие веб-технологии на предприятии и ему аналогичных;
- проанализировать текущую деятельность организации;

- разработать требования к созданию веб-сайта и веб-приложения;
- разработать веб-сайт и веб-приложение;
- оценить экономический эффект от внедрения.

Объектом исследования является заведение общественного питания кофейня «Зерно».

Предмет исследования – разработка веб-сайта и веб-приложения, исследование влияния на эффективность бизнеса.

При написании дипломной работы использовались различные источники, такие как материалы учебных пособий, диссертации, электронные информационные ресурсы и др.

В процессе исследования различных аспектов темы дипломной работы применялись такие методы научного познания, как системность и комплексность, анализ и синтез, группировка и выборка, сравнение и др.

Практическая значимость состоит в разработке веб-сайта и веб-приложения, их применения в маркетинговой и коммуникационной деятельности предприятия, автоматизации бизнес-процессов организации, выявлении экономической эффективности внедрения. Ввиду того, что организаций общественного питания существует большое количество, результаты разработки и исследования будет возможно применить для многих организаций данной сферы деятельности.

ГЛАВА 1

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ВЕБ-САЙТА И ВЕБ-ПРИЛОЖЕНИЯ

1.1. Понятие веб-сайта и веб-приложения

Для детального понимания предмета исследования, для начала необходимо дать понятие веб-технологиям, его физической основе и используемым технологиям создания и поддержания жизнеспособности.

Поскольку основой сети Веб является Интернет, необходимо ознакомиться со структурой и протоколами данной сети.

Интернет - это самая большая в мире сеть, не имеющая единого центра управления, но работающая по единым правилам и предоставляющая своим пользователям единый набор услуг. Интернет можно рассматривать как «сеть сетей», каждая из которых управляется независимым оператором – поставщиком услуг Интернета (ISP, Internet Service Provider).

С точки зрения пользователей Интернет представляет собой набор информационных ресурсов, рассредоточенных по различным сетям, включая ISP-сети, корпоративные сети, сети и отдельные компьютеры домашних пользователей. Каждый отдельный компьютер в данной сети называется хостом (от английского термина host).

Сегодняшний Интернет обязан своему появлению объединенной сети ARPANET, которая начиналась как скромный эксперимент в новой тогда технологии коммутации пакетов. Сеть ARPANET была развернута в 1969 г. и состояла поначалу всего из четырех узлов с коммутацией пакетов, используемых для взаимодействия горстки хостов и терминалов. Первые линии связи, соединявшие узлы, работали на скорости всего 50 Кбит/с. Сеть ARPANET финансировалась управлением перспективного планирования научно-исследовательских работ ARPA (Advanced Research Projects Agency) министерства обороны США и предназначалась для изучения технологии и протоколов коммутации пакетов, которые могли бы использоваться для кооперативных распределенных вычислений.

Возможно ли централизованное управление в такой глобальной сети? Ответ на данный вопрос будет отрицательным, поскольку, во-первых, данная сеть является транснациональной и, во-вторых, в силу исторических предпосылок ее формирования.

Тем не менее, в Интернете могут проявляться опосредованные формы централизации в форме единой технической политике, согласованном наборе технических стандартов, назначении имен и адресов компьютеров и сетей,

входящих в Интернет.

То есть Интернет является децентрализованной сетью, что имеет свои достоинства и недостатки.

Достоинства:

- Легкость наращивания Интернета путем заключения соглашения между двумя ISP.

Недостатки:

- сложность модернизации технологий и услуг Интернета, поскольку требуются согласованные усилия всех поставщиков услуг;
- невысокая надежность услуг Интернета;
- ответственность за работоспособность отдельных сегментов этой сети возлагается на поставщиков услуг Интернета.

Существуют различные типы поставщиков услуг Интернета:

- просто поставщик услуг Интернета выполняет транспортную функцию для конечных пользователей – передачу их трафика в сети других поставщиков услуг Интернета;
- поставщик интернет-контента имеет собственные информационно-справочные ресурсы, предоставляя их содержание в виде веб-сайтов;
- поставщик услуг хостинга предоставляет свои помещения, каналы связи и серверы для размещения внешнего контента;
- поставщик услуг по доставке контента занимается только доставкой контента в многочисленные точки доступа с целью повышения скорости доступа пользователей к информации;
- поставщик услуг по поддержке приложений предоставляет клиентам доступ к крупным универсальным программным продуктам, например SAP R3;
- поставщик биллинговых услуг обеспечивает оплату счетов по Интернету.

Одной из важнейших особенностей сети Интернет является совместимость разнородного оборудования, являющегося частью всемирной сети, что в свою очередь требует от производителей согласования общепринятых стандартов. Открытой является система, построенная в соответствии с открытыми спецификациями.

Спецификация представляет собой формализованное описание аппаратных (программных) компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, особых характеристик. Под открытыми спецификациями понимаются опубликованные, общедоступные спецификации, соответствующие стандартам и принятые в результате достижения согласия после всестороннего обсуждения всеми заинтересованными сторонами. Использование открытых спецификаций при разработке систем позволяет третьим сторонам разрабатывать для этих систем

аппаратно-программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей.

Если две сети построены с соблюдением принципов открытости, это дает следующие преимущества:

- возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся стандарта;
- безболезненная замена отдельных компонентов сети другими, более совершенными;
- легкость сопряжения одной сети с другой.

В рамках модели OSI средства взаимодействия делятся на семь уровней: прикладной, представления, сеансовый, транспортный, сетевой, канальный и физический (Рисунок 1.1). В распоряжение программистов предоставляется прикладной программный интерфейс, позволяющий обращаться с запросами к самому верхнему уровню, а именно, - уровню приложений.

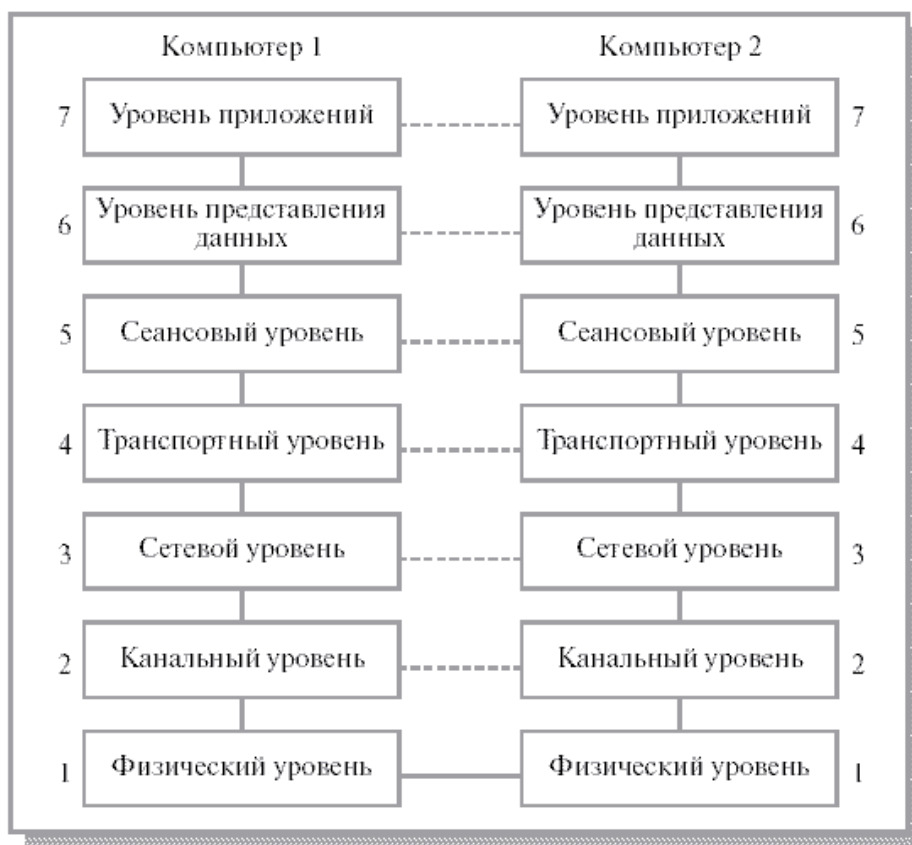


Рисунок 1.1 - Семиуровневая модель OSI

Благодаря использованию модели OSI стало возможным применение веб-приложений, которые работают независимо от используемых аппаратно-технических средств.

Сеть Интернет строилась в полном соответствии с принципами открытых систем. В разработке стандартов этой сети принимали участие тысячи специалистов-пользователей сети из вузов, научных организаций и компаний.

Разработкой стандартов для сети Веб, начиная с 1994 года, занимается Консорциум W3C (World Wide Web Consortium), основанный и до сих пор возглавляемый Тимом Бернерсом-Ли.

Консорциум W3C — организация, разрабатывающая и внедряющая технологические стандарты для Интернета и WWW. Миссия W3C формулируется следующим образом: «Полностью раскрыть потенциал Всемирной паутины путём создания протоколов и принципов, гарантирующих долгосрочное развитие Сети». Две другие важнейшие задачи Консорциума — обеспечить полную «интернационализацию Сети» и сделать ее доступной для людей с ограниченными возможностями.

W3C разрабатывает для WWW единые принципы и стандарты, называемые «Рекомендациями», которые затем внедряются разработчиками программ и оборудования. Благодаря Рекомендациям достигается совместимость между программными продуктами и оборудованием различных компаний, что делает сеть WWW более совершенной, универсальной и удобной в использовании.

Все Рекомендации W3C открыты, то есть, не защищены патентами и могут внедряться любым человеком без каких-либо финансовых отчислений Консорциуму.

Для удобства пользователей Консорциумом созданы специальные программы-валидаторы (англ. Online Validation Service), которые доступны по сети и могут за несколько секунд проверить документы на соответствие популярным Рекомендациям W3C. Консорциумом также созданы многие другие утилиты для облегчения работы веб-мастеров и программистов. Большинство утилит — это программы с открытым исходным кодом, все они бесплатные. В последнее время, повинуясь мировым тенденциям, Консорциум, в целом, гораздо больше внимания уделяет проектам с открытым исходным кодом.

Сеть WWW образуют миллионы веб-серверов, расположенных по всему миру. Веб-сервер является программой, запускаемой на подключённом к сети компьютере и передающей данные по протоколу HTTP.

Для идентификации ресурсов (зачастую файлов или их частей) в WWW используются идентификаторы ресурсов URI (Uniform Resource Identifier). Для определения местонахождения ресурсов в этой сети используются локаторы ресурсов URL (Uniform Resource Locator). Такие URL-локаторы представляют собой комбинацию URI и системы DNS.

Доменное имя (или IP-адрес) входит в состав URL для обозначения компьютера (его сетевого интерфейса), на котором работает программа веб-сервер.

На клиентском компьютере для просмотра информации, полученной от веб-сервера, применяется специальная программа — веб-браузер. Основная

функция веб-браузера - отображение гипертекстовых страниц (веб-страниц). Для создания гипертекстовых страниц в WWW изначально использовался язык HTML.

Веб-сайт – совокупность логически связанных между собой веб-страниц. Веб-страница – это документ или информационный ресурс Всемирной паутины. Типичная веб-страница представляет из себя текстовый файл, который может содержать в себе ссылки на файлы иных форматов, таких как текст, графические изображения, видео, прикладные программы, базы данных, веб-службы, а также гиперссылки для перехода на другие веб-страницы или доступа к ссылочным файлам. Современные браузеры позволяют просматривать содержание некоторых ссылочных файлов непосредственно на веб-странице, но и также в отрыве от нее.

Веб-сайт и веб-страница могут быть статическими и динамическими.

Статический веб-сайт – набор статических веб-страниц, обычно представляющие собой HTML документ, содержащий текст, изображения и мультимедиа содержимое. Изменения на статическом сайте вносятся в исходный код документов, к которым требуется доступ. Пользователи сайта не могут взаимодействовать с ним. Такие сайты носят информационный характер. Для создания данных сайтов не используются языки веб-программирования.

Динамический веб-сайт состоит из динамических страниц – шаблонов, контента, скриптов, в большинстве случаев в виде отдельных файлов. Страница сайта, показываемая в итоге браузером, формируется динамически на стороне сервера по запросу из шаблона страницы и отдельно хранящихся файлов. Как правило, для отображения любого количества однотипных страниц используется одна страница-шаблон, в которую по запросу от пользователя подгружается содержимое.

Генерация контента динамического сайта обычно происходит следующим образом:

- генерация содержимого на стороне сервера;
- отправка сгенерированного содержимого пользователю;
- генерация одержимого на стороне пользователя.

Чаще всего в современном мире встречается комбинированная генерация, при которой часть контента генерируется на стороне сервера, а остальная часть на стороне пользователя.

Веб-приложение – клиент-серверное приложение, в котором клиент взаимодействует с сервером с помощью браузера, а в качестве сервера выступает веб-сервер. Логика веб-приложения состоит в распределении функций между сервером и клиентом, где большая часть информации храниться на сервере, а передача происходит с использованием сети.

Существенное преимущество построения веб-приложений заключается в

выполнении функций независимо от используемых устройств, технических средств и операционных систем пользователя. Благодаря этому нет необходимости разрабатывать различные программы для каждой операционной системы, а достаточно развернуть ее на выбранной платформе.

Одним из типов программ, выполняемых на стороне клиента, являются сценарии, например JavaScript. Исходный текст сценария представляет собой часть веб-страницы и передается вместе с ней клиенту. Браузер обрабатывает HTML-документ, находит сценарий и выполняет его.

К программам, передающимся с серверов на сторону клиента, предъявляется требование не использовать ресурсы компьютера, на котором они выполняются. Данное требование обосновано тем, что данные программы выполняются без участия пользователя и не должны нанести вред ресурсам компьютера.

Программы, выполняющиеся на сервере, не передаются клиенту. Такие программы начинают свою работу при получении запроса со стороны клиента, обрабатываются на серверной части и генерируют веб-страницу, передающуюся веб-серверу для отправки клиенту.

Одним из видов веб-приложений является браузерное приложение. Это приложение выполняется внутри окна браузера на веб-странице. Эти приложения имеют ограниченные права, которые способствуют безопасному запуску программ на стороне клиентов без потенциально-опасного кода. Эти приложения не требуют инсталляции на компьютер, а запускаются и помещаются в кэш браузера.

1.2 Принципы создания веб-сайта и веб-приложения

Создание современного сайта разделено на этапы:

- разработка дизайна будущего сайта;
- верстка страниц сайта;
- наполнение содержимым;
- разработка сценариев и кода.

Разработка дизайна разделяется на две взаимосвязанные стадии разработки. Это UX и UI дизайн.

UX дизайн – это User Experience. Разработка данного дизайна позволяет определить, как человек будет взаимодействовать с будущим продуктом. Принятие решений разработчиком на этом этапе позволяет удовлетворить потребности пользователя, определяет, какие функции будет возможно пользователю осуществить, а также какие шаги он должен предпринять для их

осуществления.

UI дизайн – User Interface. Это то, как выглядит интерфейс, и то, какие характеристики приобретает. Дизайнеры на этом этапе определяют внешний вид изделия, удобство работы пользователя, отображение с разных устройств и т.д.

Использование в современном мире разнообразных устройств, способных получать доступ к сети интернет, отображать веб-сайты и использовать веб-приложения, определило необходимость в разработке адаптивного дизайна, так как различные устройства имеют разный размер экрана, разные органы управления и взаимодействия.

Разработчики дизайна сталкиваются с проблемой корректного отображения веб-страницы на разных устройствах, так как на экране компьютера возможно отобразить куда больше информации, чем на экране смартфона. Это влечет за собой необходимость изменять дизайн сайта, убирать или заменять некоторые его элементы, в свою очередь позволяя полноценно функционировать сайту, полноценно отображать информацию и позволять получать всю необходимую информацию пользователю.

Все этапы разработки дизайна должны быть согласованы с возможностями воплощения в реальность задумок дизайнера, обеспечение корректной и быстрой работы сайта.

Для разработки дизайна существует множество программных продуктов, позволяющих проводить все этапы и определять функциональность результата.

Среди этих программ широкое распространение получили продукты компании Adobe Systems. В их число входят:

- Adobe Illustrator;
- Adobe InDesign;
- Adobe Photoshop;
- Adobe Experience Design.

Это кроссплатформенные программы, широко используемые разработчиками по всему миру, обладающими огромным набором функций для создания любых видов дизайна.

Следующим этапом разработки является верстка страниц веб-сайта.

Стандартизированным языком разметки документов во Всемирной паутине для верстки сайта является HTML (HyperText Markup Language). Язык HTML был разработан в 1986 – 1991 годах Тимом Бернерсом-Ли, являющимся создателем Всемирной паутины и действующим председателем Консорциума Великой паутины W3C. HTML создавался как язык для обмена технической и научной документации, с помощью которого можно было создать простой и красиво оформленный документ. Помимо упрощения структуры документа, HTML поддерживал гипертекст – система из текстовых страниц, имеющих перекрестные ссылки.

Изначально он создавался как средство структурирования и форматирования документов без их привязки к воспроизводящим средствам. Основной целью была возможность отображения текстовых документов без стилистических и структурных изменений на различном оборудовании, таком как цветной экран компьютера, монохромные экраны, экран мобильного устройства и даже устройства и программы голосового воспроизведения текста. Но современное применение HTML отделилось от изначальной задачи. Современные тенденции определили развитие языка HTML в осуществление графического и мультимедийного оформления.

Текстовые документы, содержащие разметку на языке HTML обрабатываются специально созданными приложениями, которые представляют документ пользователю в его отформатированном виде. Такие приложения, называемые «браузерами» предоставляют пользователю удобный интерфейс для работы с веб-страницами, их просмотре и при необходимости отправки данных серверу. На сегодняшний день наиболее популярными браузерами являются Google Chrome, Safari, Mozilla.

Язык HTML имел множество версий, разрабатываемых и утверждаемых с ноября 1995 года. На данный момент рекомендована к применению версия HTML 5.1, принятая к разработке 17 декабря 2012 года и принятая с 1 ноября 2016 года.

Одним из основных требований в данной спецификации является использование специального языка описания внешнего вида документа, написанного с помощью языка разметки. Таким языком является CSS – Cascading Style Sheets.

Целью создания данного языка было использование его веб-разработчиками для оформления элементов, для представления их на веб-странице отдельно от документа, описывающего логическую структуру данной страницы. С помощью этого языка возможно оформить цветовое представление документа, используемые шрифты текста, настроить изображения и положение элементов на странице, отображение их на экране различных устройств. Кроме того, CSS имеет возможность представить один и тот же документ в различных стилях и методах вывода для разных устройств.

К CSS с 1996 года консорциумом W3C стали издаваться рекомендации. На данном этапе существует 3 уровня рекомендаций, а также еще один находится в разработке.

Наиболее полно CSS поддерживается наиболее популярными веб-браузерами, работающими на движках Gecko, WebKit и Presto.

Реализованный макет документа является базой для внесения в него содержимого. Современный подход к верстке веб-страниц предполагает создание макета таким образом, где каждому элементу и содержимому

определено его место, закреплено средствами верстки. Это позволяет при загрузке страницы сразу загружать ее макет, содержимое которой может быть загружено с задержкой. Данное свойство позволяет не изменяться странице при подгрузке дополнительного содержимого и не смещать другие элементы.

На следующем этапе создания динамических веб-страниц и веб-приложений разрабатываются коды и скрипты. Они определяют, как будет взаимодействовать страница веб-сайта на производимые пользователем действия, какова будет реакция на определенные события, взаимодействие клиентской стороны с веб-сервером.

Некоторые свойства веб-страницы реагировать на действия пользователей возможно осуществить с помощью языков HTML и CSS. Однако современные рекомендации создания веб-страниц предполагают использование отдельных программных документов.

Каждый элемент в гипертекстовом документе представляет собой объект с наличием у него различных свойств. Для работы с этими объектами, а также изменения их свойств используются специальные объектно-ориентированные языки программирования.

Языки веб-программирования делятся на клиентские языки и серверные.

Программы, написанные на клиентских языках, обрабатываются на стороне пользователя и, как правило, их выполняет браузер. Это вызывает основную проблему использования клиентских языков – такие программы зависят от браузера пользователя. Если пользователь запретил запуск клиентских программ, то они не будут исполняться без его разрешения. С другой стороны, использование клиентских программ снижает нагрузку на сервер, так как выполняются без обращения к нему и в большинстве случаев не требуют перезагрузки страницы.

При использовании программ, написанных на серверных языках программирования, все запросы от клиента обрабатываются сначала на сервере, проводятся необходимые действия со страницей, а затем отправляется на сторону клиента уже в готовом виде.

Основным и широко используемым клиентским языком программирования является язык JavaScript.

Необходимостью создания языка JavaScript была разработка браузера компанией Netscape, который должен был поддерживать язык программирования, встраиваемый в HTML-код страницы. По инициативе компании была проведена стандартизация языка ассоциацией ECMA в связи с заявлением множества компаний о намерении использования языка JavaScript в своих будущих продуктах.

JavaScript является объектно-ориентированным языком

программирования, но используемое в языке прототипирование отличает работу с объектами в отличии от традиционных класс-ориентированных языках. Кроме того, JavaScript имеет ряд функций, присущих функциональным языкам, что придает языку дополнительную гибкость.

Кроме использования JavaScript как клиентского языка веб-приложений, он нашел применение в AJAX – популярном подходе к построению интерактивных пользовательских интерфейсов веб-приложений, который заключается в асинхронном способе обмена данными браузера и сервера. Также широко используется в модели Comet – модель работы веб-приложения, использующее постоянное HTTP-соединение для отправления веб-сервером данных браузеру без дополнительных запросов от браузера.

Также JavaScript может использоваться при написании серверных приложений, что позволяет обрабатывать программы на стороне сервера, используется для написания мобильных приложений для операционной системы Palm WebOS.

При использовании языка программирования JavaScript программисту необходимо достичь высокий уровень абстракции и допустимого уровня кросс-браузерности при разработке веб-приложений. Для этого используются многочисленные библиотеки JavaScript. Они представляют собой набор многократно используемых объектов и функций.

Среди существующих библиотек можно отметить AngularJS, jQuery, Prototypeи другие.

На сегодняшний день поддержку JavaScript обеспечивают современные версии всех часто используемых браузеров.

Еще одним языком программирования, используемым при разработке веб-приложений, является язык PHP – Hypertext Processor.

PHP один из самых популярных языков веб-программирования, в частности серверной части. Популярность обуславливается наличием большого набора функций при разработке веб-приложений. Основные из них:

- взаимодействие с большим количеством различных систем управления базами данных;
- обработка файлов, загружаемых на сервер;
- работа с HTTP-авторизацией;
- работа с локальными и удаленными файлами.

Однако использование данной структуры при создании веб-сайта и веб-приложения требует определенных знаний в верстке, гипертекстовой разметке и знаний различных языков программирования. Это повлекло разработку специальных компьютерных программ и информационных систем, используемых одновременно для создания, редактирования и управление

содержимым. Такие системы получили название «Система управления содержимым» или CMS.

Основными функциями CMS являются:

- предоставление инструмента создания и организации работы над содержимым;
- управление содержимым: хранение, контроль версий, определение режима доступа, управление потоками документов;
- публикация содержимого;
- представление контента в виде удобном для навигации и поиска.

Система управления содержимым веб-страниц один из видов CMS систем. Она часто используется для создания личных страниц, блогов, интернет-магазинов и нацелены на пользователей, которые не знакомы с веб-разработкой.

Наиболее популярной системой управления содержимым веб-страниц является WordPress – свободное программное обеспечение для создания сайтов, блогов и веб-приложений.

WordPress обладает многими возможностями в создании веб-сайта. Среди них:

- настраиваемый дизайн;
- адаптивность страницы;
- высокая производительность;
- управление сайтом;
- высокая безопасность;
- гибкое управление медиафайлами.

Также существует огромное количество облачных платформ для создания собственных сайтов. Используемые редакторы работают и создают сайт на платформе HTML5 и используют язык стилевой разметки CSS3. В них включены множества наборов шаблонов сайтов.

Также, как и при использовании CMS систем, данные платформы направлены на помощь в создании сайтам людям, не обладающими навыками верстки и написания программного кода для веб-сайта.

Проведя сравнительный анализ платформ, выделились 5 наиболее известных: Wix, Jimbo, A5, UMI, SetUp. Сравнительные характеристики платформ представлены в таблице 1.1.

Таблица 1.1 – сравнительные характеристики облачных платформ

	Wix	Jimbo	A5	UMI	SetUp
Язык интерфейса	Русский и 10 других	Русский и другие	Русский	Русский	Русский
Техподдержка	Форма обратной связи	Форма обратной связи	Онлайн-чат	Форма обратной связи	Онлайн-чат
Бесплатная пробная версия	14 дней	30 дней	14 дней	45 дней	14 дней
Готовые шаблоны	Более 300	Более 500	Более 300	550	Более 700
Разработка дизайна	HTML5/CS S3	HTML5/CS S3	HTML5/CS S3	CSS3	HTML5/CS S3
Доступ к коду сайта	Да	Да	Нет	Нет	Нет
Размещение фотогалереи	Да	Да	Да	Да	Да
Организация почты	Да	Да	Да	Нет	Нет
Создание блога, ленты новостей	Да	Да	Да	Да	Да
Интеграция с системами веб-аналитики, виджетами	Google Analitics	Google Analitics	Google Analitics	Google Analitics	Google Analitics
Проверка отображения на устройствах	Да	Да	Нет	Да	Да

Каждый из выбранных способов создания сайта, будь то верстка на чистом HTML и CSS и написание кода с использованием знаний языков программирования, или же использование специальных программ, фреймворков и облачных платформ, решает проблему создания сайта по-своему. Для каждой задачи требуется определить, какими средствами возможно выполнить ее в кратчайшие сроки с достижением наилучшего результата и задействования наименьших затрат. Это является начальным этапом любой разработки, а именно

проектирование.

ГЛАВА 2

ПРОЕКТИРОВАНИЕ ВЕБ-САЙТА И ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПРЕДПРИЯТИЯ ОБЩЕСТВЕННОГО ПИТАНИЯ «КОФЕЙНЯ «ЗЕРНО»

2.1 Описание основных технико-экономических характеристик предприятия общественного питания «Кофейня «Зерно»

«Кофейня «Зерно» является комплексом, состоящим из двух заведений общественного питания, зарегистрированных на две разные организации: ЧТУП «БелАгроПарк» и ООО «Мастер Бариста».

«Зерно» является представителем предприятия общественного питания, которое оказывает услуги общественного питания посредством производства кулинарной продукции, ее реализации и организации питания различных групп населения.

«Зерно», зарегистрированное на ЧТУП «БелАгроПарк», было открыто в 2012 году в городе Минске по адресу: проспект Независимости, дом 46. Реализует кондитерскую продукцию собственного производства, алкогольную и безалкогольную продукцию, занимается реализацией товаров кофейной сферы.

«Зерно», зарегистрированное на ООО «Мастер Бариста» как детское кафе, было открыто в 2014 году в городе Минске по адресу: улица Козлова, дом 6, помещение 47. Реализует кондитерскую продукцию, безалкогольные напитки, занимается реализацией товаров кофейной сферы.

Сразу после открытия кофейня закрепились на рынке оказания услуг и стала развивать новые направления деятельности.

На данном этапе развития организации идет открытие подразделения по производству и реализации свежееобжаренного зернового кофе.

Основная задача предприятия: оказание услуг в сфере общественного питания.

Основной целью коммерческой деятельности предприятия является получение прибыли.

Основными видами услуг, оказываемыми кофейней «Зерно», являются:

- производство кондитерской продукции;
- производство свежееобжаренного зернового кофе;
- производство безалкогольных напитков, кофе, чая и напитков на основе алкоголя;
- реализация собственной продукции на территории предприятия общественного питания;
- организация потребления продуктов питания;
- организация отдыха, развлечений и досуга потребителей;

- предоставление продукции и услуг высокого качества;
- предоставление дополнительных сопутствующих услуг;
- обучение по направлениям бармен-официант, бариста.

Все услуги, оказываемые предприятием, и деятельность соответствуют требованиям пожарной безопасности, санитарных норм и придерживаются требованиям ГОСТ 30389 от 2013 года.

Стратегия развития предприятия:

- совершенствование оказания услуг;
- обеспечение потребности населения в производимой продукции и оказываемых услугах;
- повышение качества производства;
- выход на новые рынки сбыта продукции;
- расширение деятельности предприятия.

Для эффективного управления предприятием оно должно иметь соответствующую целям и задачам деятельности структуру. Основой для формирования отдельных административных функций является организационная структура. Структура выявляет и закрепляет взаимоотношения между отделами и сотрудниками организации, распределяет обязанности по качественному управлению.

Организационная структура обеих кофеен, зарегистрированных на разные предприятия, схожа в связи с аналогичным видом деятельности и осуществлением работы одними людьми, работающими на два заведения. Таким образом возможно представить структуру двух заведений как единую.

Организационную структуру схематично можно представить в виде «дерева», поскольку существует иерархия во взаимоотношениях сотрудников и отделов. Управление предприятием строится по линейной функции взаимоотношений и делегирования полномочий, так как управление исходит от начальника к непосредственным подчиненным и так далее.

Структура управления предприятием приведена на рисунке 2.1



Рисунок 2.1 - Организационная структура предприятия

Прежде чем начать описывать функции отдельных звеньев, следует обратить внимание, что предприятие является объектом малого бизнеса, в котором работает небольшое число сотрудников, поэтому сложно отметить грань между исполняемыми обязанностями каждого элемента организационной структуры. На практике часто происходит разделение обязанностей между сотрудниками организации. Традиционно принято выделять три уровня управления: высший, средний и оперативный. Помимо управляющих отделов в организации существует штат рядовых сотрудников, выполняющих должностные обязанности и занимающимися рутинными делами, то есть занятые непосредственно в производстве продукции и оказании основных услуг. Сведения о персонале предприятия представлены в таблице 2.1.

Таблица 2.1 – Сведения о персонале предприятия

Наименование структурных подразделений и должностей	Количество штатных единиц
Учредитель	1
Управляющий директор	1
Бухгалтер	1
Руководящий состав	1
Специалисты	1
Рабочие	20

Продолжение таблицы 2.1

Итого	25
-------	----

На предприятии часть функций оперативного управления выполняют непосредственные работники, а именно составляют расписание работы, определяют ассортимент предлагаемой продукции, занимаются организацией рабочего процесса. Контролем над уровнем качества работ, количеством отработанного времени, объемом выполненных работ, составлении отчетов занимается руководящий состав организации.

Высший уровень управления, на котором располагается учредитель организации, исполняет стратегическую и тактическую политику, определяет стратегическое планирование и цели организации.

Сотрудники кофейни «Зерно» осуществляют свою деятельность согласно должностным инструкциям, разработанным и утвержденным на основании трудового договора и в соответствии с положениями Трудового кодекса Республики Беларусь и других нормативных актов, регулирующих охрану труда и трудовые правоотношения на территории Республики Беларусь.

Результат хозяйственной деятельности предприятия общественного питания характеризуется объемом его производственной и торговой деятельности.

Товарооборот в кофейне включает в себя две основные части:

- реализация продукции собственного производства;
- реализация покупных товаров.

Товарооборот по продукции собственного производства рассчитывается исходя из определения продажной стоимости продукции на основе данных о затратах на сырье, арендную плату, заработную плату работников и используемые ресурсы.

2.2 Анализ действующих инструментов для взаимодействия с клиентами

На основе проведенного анализа используемых инструментов взаимодействия с клиентами было выявлено активное использование информационных технологий в ходе работы.

Были выявлены следующие инструменты:

- Активное продвижение продукции и услуг, оказываемых предприятием, в социальных сетях. Реализованы личные страницы, активно наполняемые контентом, ведущие активное взаимодействие с клиентами. Их представление можно увидеть на рисунках 2.2 и 2.3.

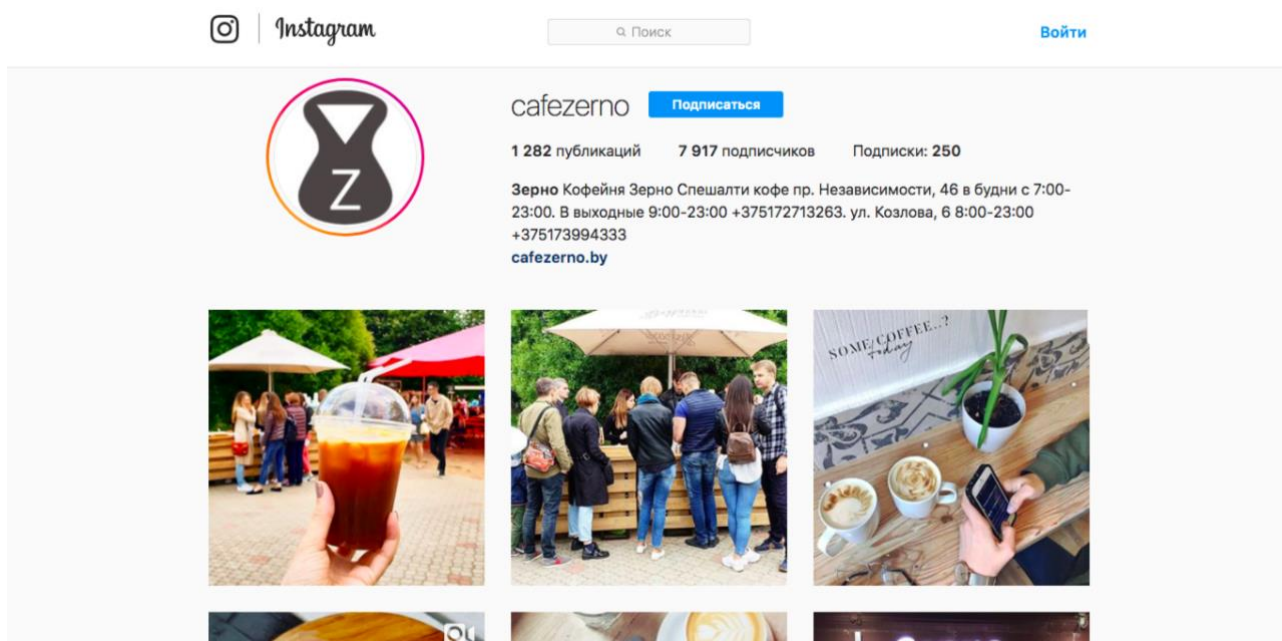


Рисунок 2.2 – Страница в социальных сетях

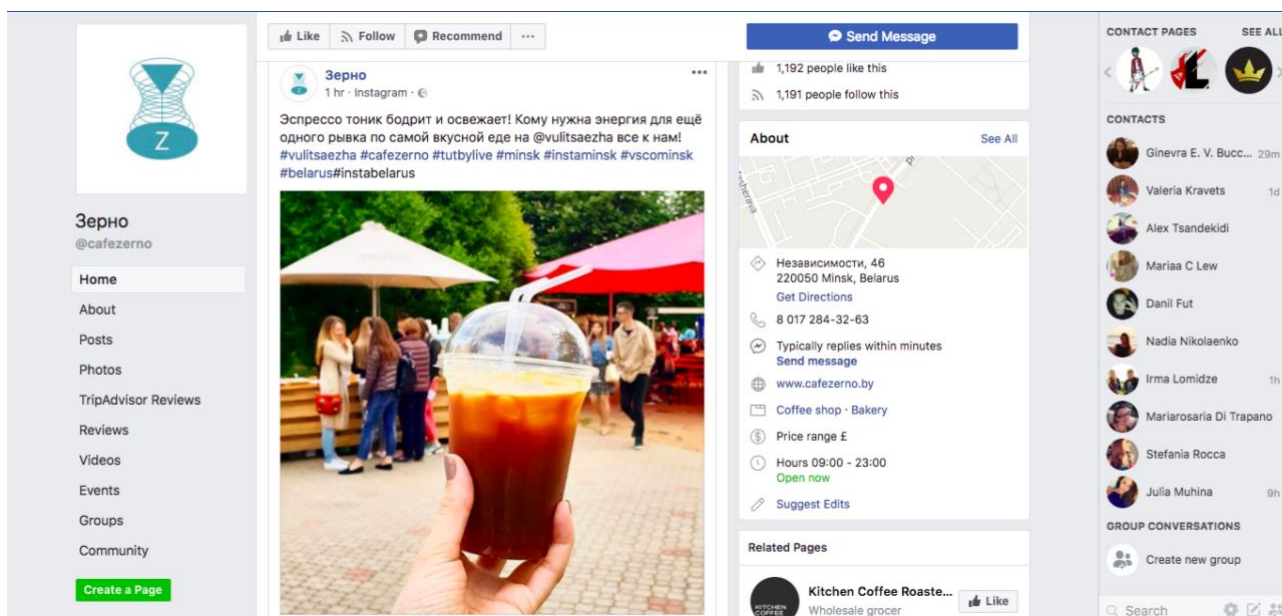


Рисунок 2.3 – Страница в социальных сетях

– Продвижение с использованием сторонних интернет ресурсов.

К таким ресурсам можно отнести порталы, на которых собраны различные заведения, интернет поисковики, новостные блоги, сообщества и сайты. Также отметка на популярных онлайн-картах входит в их число.

Использование данных ресурсов помогает получить обратный отклик от клиентов, собрать отзывы. Они позволяют определить рейтинг заведения. Примеры представлены на рисунках 2.4 и 2.5.

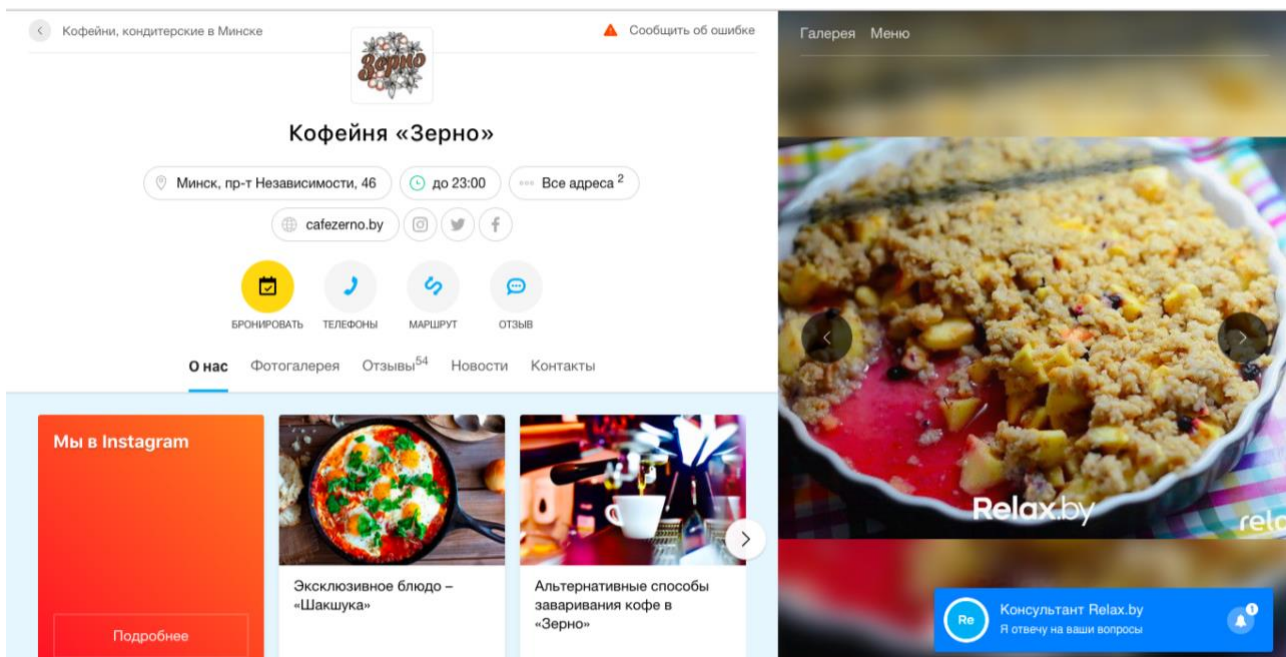


Рисунок 2.4 – Страница на популярном интернет-портале.

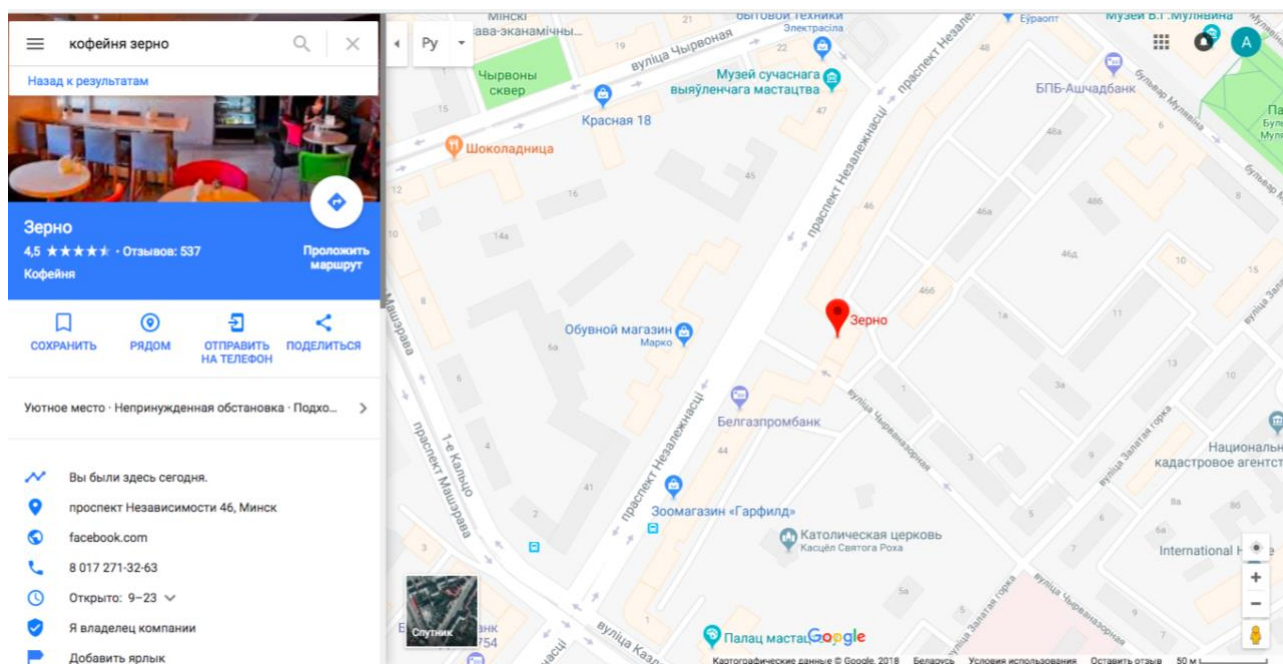


Рисунок 2.5 – Отметка на карте с информацией из поисковика

– Сайт организации.

Организация имеет свой сайт, размещенный по адресу <http://cafezerno.by>. Данный сайт был написан с помощью CMS WordPress. Однако не соответствует требованиям заведения и не отвечает поставленным задачам маркетинговой стратегии организации. Главная страница сайта представлена на рисунке 2.6.

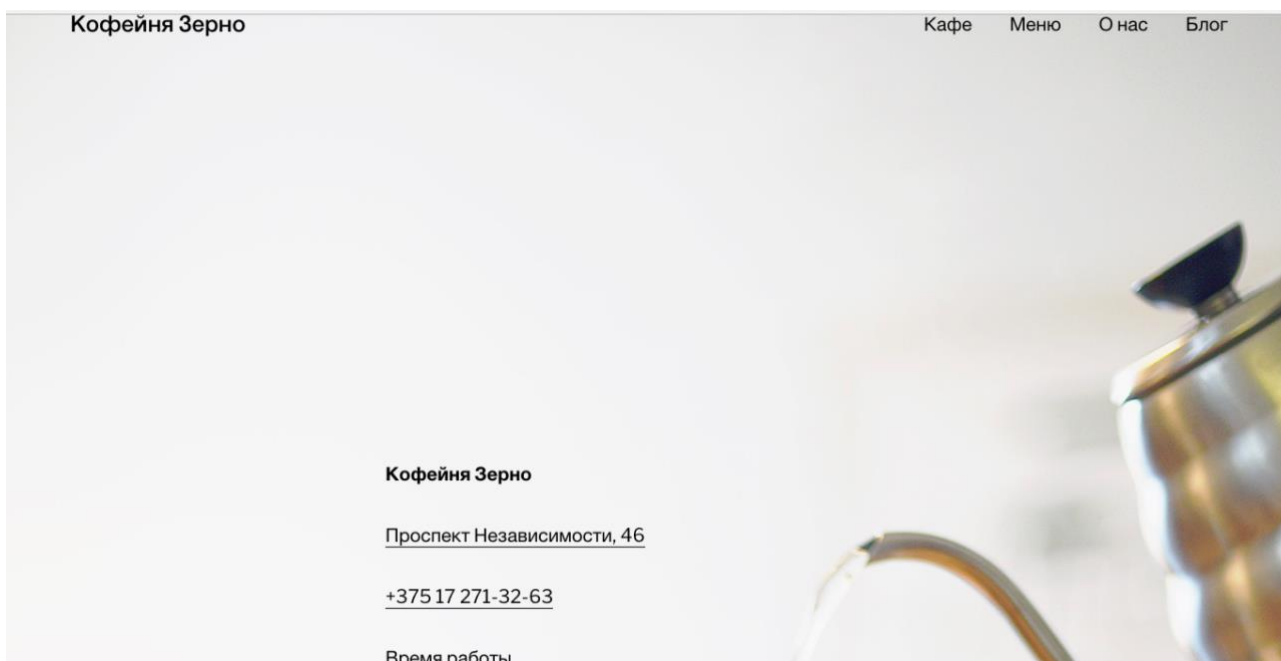


Рисунок 2.6 – Главная страница сайта

На сайте не хватает информации о заведении, оказываемых им услугах и продукции. Отсутствует интерактивная составляющая.

В ходе исследования было выявлено отсутствие полноценного сайта. Также были выявлены процессы, осуществляемые без использования информационных технологий, которые возможно было бы реализовать с применением веб-разработки. Из этого следует вывод, что необходимо реализовать разработку личного сайта организации с внедрением разработанного приложения для реализации некоторых бизнес-процессов.

2.3 Выбор средств и методов разработки веб-сайта и веб-приложения

Проектирование веб-сайта и веб-приложения стоит начать с определения первоначальных целей и области решаемых задач. На этом этапе также стоит определить заинтересованные лица, пользователей и потребителей.

Веб-сайт должен отвечать требованиям по полноценному информированию клиентов, как существующих, так и потенциальных. Необходимо, чтобы он обладал всей информацией об организации, предоставляемых ей услугах, виде деятельности.

Веб-приложение, размещенное на сайте организации, решает задачи по автоматизации бизнес-процесса заведения, отвечает целям минимизации затрат и времени на выполнение процесса, увеличения качества выполняемых работ и повышения уровня качественного оказания услуг клиентам.

Список заинтересованных лиц:

- клиенты организации. Разработанный ресурс позволит им в полной мере получать актуальную информацию о состоянии заведения, оказываемых

им услугах, оценивать качество предоставляемых услуг, выполнять действия по взаимодействию с организацией.

- работники организации. Станет возможным снижение возлагаемых задач на работников, упрощение их рутинных процессов.
- маркетолог. Веб-сайт является наилучшим инструментом современного продвижения бизнеса, увеличения возможностей по информированию и взаимодействию с клиентами, как существующими, так и потенциальными.
- руководство. Со их стороны выгода от использования данных технологий отразится на росте выручки, минимизации затрат ресурсных и трудовых. Также повышение имиджа заведения является одной из приоритетных целей для руководства.

На этапе проектирования определяется список задач, поставленных перед будущим ресурсом. Этот список задач определяется заказчиком проекта, выраженном в техническом задании. Разработчику остается реализовать готовый проект и, возможно, внести некоторые изменения, связанные с навыками, возможностями разработки и опыта.

Был предъявлен следующий список требований к сайту и веб-приложению:

- Дизайн. Дизайн должен быть сдержанным, не отвлекать от контента, располагаемого на странице. Обязаны содержаться логотипы заведения. Дизайн выполняется в цветовой гамме, присущей заведению. Дизайн же веб-приложения должен быть максимально функциональным, расположение элементов интуитивно понятным для пользователя.
- Функционал. Основными функциями сайта является донесение информации до потребителя, так как в основном сайт должен носить информационный характер. Должна отсутствовать сложность навигации, интуитивно понятное расположение функциональных элементов, ясный порядок выполняемых действий.
- Контент. Контент должен в полной мере отражать сущность заведения, сферу его деятельности, оказываемые им услуги, возможности по взаимодействию с клиентами. Информация, представленная на сайте, должна с лучшей стороны представлять организацию, ее работников и качество оказываемых услуг.

Отдельно были выдвинуты требования к разработке веб-приложения для организации.

Веб-приложение должно реализовывать функцию бронирования клиентом заведения места на определенную дату. Веб-приложение должно располагаться на сайте организации. Его функциями будет являться отправка в организацию формы с данными клиента, его желаниями по поводу осуществления

бронирования, в которую включается дата и время брони. Это приложение составляет список для каждого отдельного клиента о его осуществленных бронях.

Выбор средств разработки сводится к разделению процесса на основополагающие этапы.

Первым этапом является разработка функционального и пользовательского дизайна интерфейса.

Для разработки различного вида дизайнов существует огромное количество графических редакторов с необходимым функционалом. С помощью них возможно составить функциональный макет с обозначением основных элементов, проверкой удобства использования разработанного интерфейса веб-сайта и веб-приложения.

Данный дизайн носит название UX/UI дизайн.

В таблице 2.2 приведен сравнительный анализ существующих популярных средств разработки функционального и пользовательского дизайнов.

Таблица 2.2 – Сравнительный анализ графических редакторов

	Sketch	Adobe Photoshop	Adobe Illustrator	Adobe InDesign
Операционная система	MacOS	MacOS, Windows	MacOS, Windows	MacOS, Windows
Стоимость	200 BYN	20-100 BYN в месяц	20-100 BYN в месяц	20-100 BYN в месяц
Форматы изображения	Векторный	Растровый, векторный	Векторный	Растровый

Для разработки функционального макета сайта необходим векторный редактор, так как макет представляет из себя представление будущих функциональных блоков сайта и веб-приложения. Данные блоки представляют из себя графические фигуры. Исполнение данных блоков в векторном формате позволит в процессе работы перемещать, масштабировать их изображения без потери качества отображения элементов на экране

Разработка пользовательского дизайна требует работы с готовыми изображениями, фотографиями. Эти элементы будут размещены непосредственно на самом сайте, будучи переведенными в растровый формат. Поэтому для разработки пользовательского дизайна потребуется графический редактор, способный работать с растровыми изображениями и фотографиями.

В идеальном случае разработчику удобнее использовать универсальный графический редактор, обладающий функционалом как растрового, так и векторного редакторов. Это позволит более эффективно вести процесс разработки функционального и пользовательского интерфейса, так как на

практике этим занимается один человек.

Из сравнительной таблицы можно сделать вывод об удобстве использования графического редактора Adobe Photoshop.

Adobe Photoshop является многофункциональным графическим редактором, разработанный и распространяемый компанией Adobe Systems. Основой его работы являются растровые изображения – изображения, представляющие из себя набор пикселей определенного цвета, однако в своем функционале он имеет некоторые инструменты, способный создавать векторные изображения – способ представления, который основывается на математическом представлении элементарных геометрических объектов, таких как точки, линии, круги и окружности, многоугольники. Данный продукт является лидером рынка в области коммерческих средств редактирования растровых изображений. Он поддерживает большинство цветовых моделей изображений: RGB, LAB, CMYK, черно-белые, градации серого, индексированные и мультисканальные. Способен создавать необходимые в графическом дизайне сайта изображения с прозрачным фоном. Также при создании макета возможно распределять элементы по слоям, для последующего использования элементов макета при разработке программного продукта.

Следующим шагом является верстка макета.

Верстка представляет из себя создание HTML-кода, размещающего элементы веб-страницы в окне браузера согласно разработанному макету, таким образом, чтобы элементы дизайна выглядели аналогично макету. Верстка веб-страниц отличается от полиграфической тем, что необходимо учитывать разные возможные отображения при использовании разных браузеров и разницу в размерах рабочего пространства устройств, на котором будет отображаться будущий сайт. Существует огромное количество применяемых подходов к верстке, имеющие свои преимущества и недостатки. На каждом этапе разработчик сам выбирает используемый им подход, сталкивается с решением задач. Также стоит учитывать поддержание будущего макета адаптивным к изменениям в спецификациях языка верстки, изменений в версиях веб-браузеров и актуальности используемых библиотек.

Как правило верстальщик получает готовый дизайн-макет будущей страницы. Далее идет процесс анализа полученного макета. Анализируется макет путем разбиения его на отдельные элементы и так называемые «этажи». «Этажи» представляют из себя горизонтально выделенные элементы страницы, занимающие всю ее ширину и, как правило, логически и функционально определенные на будущей странице. Далее эти элементы делятся на блоки, из которых они состоят.

Далее происходит этап рекурсивного сбора этих элементов из макета на будущий проект. Каждому элементу задается область, в которой он должен

располагаться, границы допустимого смещения.

Затем идет проверка на кросс-платформенность – способность программного продукта работать на более чем одной платформе, не теряя своих функциональных характеристик и отображения на устройстве пользователя. В общем случае данная проверка должна ответить на следующие вопросы:

- одинаково ли отображается страница на разных браузерах, операционных системах и устройствах;
- происходит ли смещение за пределы границ допустимого при изменениях настройки браузера (размер стандартного шрифта браузера, величина окна браузера);
- произойдет ли смещение блоков при отключении показа изображений в браузере, либо при ошибке загрузки этих изображений;
- как окажет влияние разрешение монитора на отображение содержимого.

Критические исправления необходимо внести в документ, после чего весь процесс проверки повторяется до полного удовлетворения требований кросс-платформенности будущего продукта.

Различаются разные подходы к верстке будущей веб-страницы. Данные подходы различаются по своей сущности, но не являются закреплёнными в правилах. Каждый верстальщик выбирает для себя подход, опираясь лишь на свой опыт. Некоторые варианты могут быть рекомендованы к использованию.

На данный момент существует несколько основных подходов к верстке:

- фреймы;
- таблицы;
- слои;
- блоки.

Фреймы являются отдельными и законченными HTML-документами, которые могут быть совместно отображены в окне браузера. Фреймы разбивают веб-страницу на набор отдельных, которые могут иметь свой адрес. Для каждого фрейма определен свой собственный размер. А переход по ссылкам в одном фрейме перезагрузит только его содержимое, но не помешает видеть другие страницы в окне браузера. Фреймы часто использовались для навигации по веб-сайту. При этом навигационная страница располагалась в одном окне, а текст подгружался в другое окно. Однако в современном стандарте HTML5 фреймы больше не поддерживаются.

При табличном подходе верстка веб-страницы похожа на составление обычной текстовой таблицы. Окно веб-сайта делиться на строки, которые в свою очередь разделяются на столбцы. Их возможно, как и в обычной таблице, объединять, разделять, задавать им размер и определять расположение объектов в них. Ячейки таблиц имеют особые настройки выравнивания, которые невозможно повторить, используя другие подходы, поэтому иногда

использование такого подхода является незаменимым.

Недостатками табличного подхода является тот факт, что страница грузиться из HTML-файла построчно, а это приводит к тому, что ни один элемент таблицы не отобразится, пока загрузка не дойдет до последнего элемента таблицы, обозначающего ее завершение. Таким образом при медленном интернет-соединении или обрыве в сети, невозможность загрузить один из элементов таблицы, будь то изображение большого размера или ссылку, приведет к невозможности загрузки всей страницы. Также при написании табличных страниц недостатком является избыток кода, усложняющего его понимание.

Верстка слоями была разработана компанией Netscape и использовалась в браузере, произведенной их компанией. Слои было возможно скрывать, накладывать друг на друга, устанавливать положение относительно окна браузера. В слой можно было включать содержимое блока из файла. Однако этот подход не был включен в спецификацию HTML, поэтому содержимое могло отображаться только при использовании браузера компании Netscape.

Блочная верстка является по праву самой широко-используемой. Верстка происходит с помощью специального тега `<div>` стили которого описываются в таблице CSS. С помощью блочной верстки реализуют концепцию семантической верстки.

Семантическая верстка заключается в использовании HTML тегов в соответствии с их предназначением. Каждый тег получил отдельное название, описывающее его предназначение. Данные блоки не дают определенных преимуществ в верстке, так как они просто являются аналогами одного блочного элемента и обладают набором одинаковых функций. Однако с помощью их использования код становится более понятным при его анализе, они помогают логически разбить страницу на элементы, исходя из их функционального назначения. Оформление элементов происходит с использованием каскадных таблиц стилей.

С появлением семантической верстки, разработчикам представляется возможным разбивать логически страницу на шапку сайта, блок навигации, основное содержимое сайта, боковые панели, подвал страницы. Это упрощает разработку и понимание кода, делает его более дружелюбным.

По принципам использования средств разметки HTML различают логическую и физическую разметки. При использовании физической разметки обычно ориентируются на внешний вид, а используя логическую разметку отмечают логическое предназначение выделенного элемента, преимуществом которого является независимость верстки от используемого типа устройств и дизайна веб-страницы.

Макеты верстки сайтов можно разделить на 5 типов:

- фиксированные (статические);
- резиновые (имеющие процентную величину);
- эластичные;
- адаптивные;
- комбинированные.

Верстальщик при разработке использует следующее необходимое ПО:

- текстовый редактор или редактор HTML для написания и редактирования кода или интегрированную среду разработки, включающую в себя текстовый редактор, компилятор, средства автоматизации сборки и отладчик;
- графическая программа для нарезки макета на элементы.

В данном случае целесообразно использовать графическую программу, аналогичную используемой графическим дизайнером макетов, обладающую набором схожих функций и поддерживающей одинаковые форматы изображения.

Исходя из этой особенности стоит использовать при подготовке макета к верстке графический редактор Adobe Photoshop, описанный выше.

Выбор же текстового редактора или интегрированной среды разработки представлен довольно большим разнообразием альтернативных вариантов.

Использование обычного текстового редактора подходит для написания несложных HTML кодов. Их доступно огромное количество для разных платформ. Данные редакторы занимают малое место на рабочем компьютере, мало загружают его процессор и довольно просты в использовании. К таким текстовым редакторам можно отнести «Блокнот», «Notepad++» имеющий дополнительный функционал. Также подробнее опишем наиболее популярные среди верстальщиков редакторы.

Sublime Text – самый обычный текстовый редактор. Он имеет большое количество тем оформления и ресурсов, существует настройка рабочей области, навигация по файлам, используемым для работы, миникарта для навигации по открытому файлу. Также редактор имеет большое количество расширений, которые позволяют писать сокращенный код, автоматически преобразуемый в полноценный, создавать и редактировать файлы прямо из текстового редактора, контролировать ошибки в синтаксисе и ориентироваться в парности написанных тегов, различать элементы кода по цвету.

Adobe Brackets – текстовый редактор популярной компании. В него также включены различные темы оформления и возможность разбиения экрана на несколько рабочих областей. Имеет полезное расширение, позволяющее видеть изменение в браузере без сохранения проекта и частой перезагрузки браузера. Также имеет полезное расширение, позволяющее изменять таблицу стилей через страницу редактирования разметки.

Atom – текстовый редактор, написанный другой известной командой - GitHub. Данный редактор имеет те же функции, что и предыдущие: выбор тем оформления и расширения, браузер файловой структуры, миникарту, поиск по файлу.

Отличия заключаются в структуре. Редактор написан на веб-языках HTML, CSS и JavaScript, в связи с чем имеет гибкую и расширяемую структуру. Его исходный код открыт, что позволяет разработчикам самим создавать модули для работы в этом приложении.

Также одной из важных особенностей данного редактора является возможность его работы с Git. Это технология, позволяющая получать доступ и работать с файлами, загруженными на созданном вами удаленном репозитории. С его помощью можно загрузить файлы на свой компьютер, внести изменения в них, прокомментировать, отправить обратно исправленный файл. Также репозиторий имеет историю всех шагов по работе и изменениям файлов, что позволяет отменять все внесенные изменения, откатываться к любой версии файла и следить за всеми происходящими изменениями в проекте.

Visual Studio Code – кроссплатформенный редактор, который имеет базовые возможности интегрированной среды разработки. Таким образом он уже выходит за пределы функционирования обычного текстового редактора и предоставляет большинство функций ИСР. Он имеет в своей структуре компилятор, Git, консоль и дебаггер. Редактор осуществляет поддержку более 30 языков, что позволяет легче писать код на выбранном языке. Это позволяет писать сокращенный код, преобразуемый в полноценный, на выбранном вами языке, получать подсказки при вводе, дает описание используемым элементам, отображает разные элементы в разной цветовой палитре, для лучшей навигации по коду. Требуется минимального свободного места на диске, в отличие от полноценной среды разработки. Также имеет встроенный отладчик для Node.js. Обладает возможностями работы с Git прямо из редактора.

Исходя из необходимых функций будущего проекта, объема работ для верстки макета был выбран редактор кода Visual Studio Code.

Одновременно с этапом верстки макета наступает необходимость в визуальном оформлении веб-страницы. Для этого используется каскадная таблица стилей CSS.

Задача разработчика на этом этапе применить стили к элементам HTML-кода так, чтобы макет соответствовал дизайну, разработанному на первом этапе. Этот этап взаимосвязан с версткой макета, так как в современных рекомендациях все стили указываются с помощью CSS. Верстальщик связывает текстовый файл HTML с таблицей стилей.

CSS файл также является текстовым файлом, созданным с помощью тех же текстовых редакторов.

В файле каскадной таблицы стилей указываются ссылки на элементы HTML файла, к которым присваиваются определенные характеристики. Все элементы в HTML документе строятся на основании иерархии, то есть одни элементы помещаются внутрь друг друга, другие являются соседними. Друг для друга они могут быть родительскими, дочерними, предками, потомками, сестринскими. В связи с этим различается обращение к данным элементам из таблиц стилей.

С помощью CSS осуществляется адаптивная верстка веб-страницы.

Также, CSS присущи свойства, помогающие в реализации кросс-браузерности будущего продукта. Так как для каждого браузера существуют свои стандартные настройки отображения содержимого, позиционирования веб-страницы в окне браузера и применения различных элементов, таких как шрифт текста, с помощью таблицы стилей возможно изменение стандартных настроек браузера как для отдельных элементов страницы, так и для всего документа в целом. Одной из широко применяемых практик среди верстальщиков является нормализация или сброс стандартных стилей. Это приводит к абсолютно одинаковым настройкам разных браузеров, а позиционирование элементов на странице происходит исключительно заданными стилями разработчика.

Исходя из факта, что HTML и CSS являются взаимосвязанными файлами, целесообразно будет использование единого текстового редактора, в котором данные файлы добавлены в единую рабочую область. Выбор редактора остается неизменным, так как он обладает всеми необходимыми функциями и требованиями к разработке. Этим редактором является Visual Studio Code.

Так, с помощью верстки, стало возможным получить статическую версию сайта.

В современном мире для успешной работы сайта и соответствия задачам и целям разработки сайта для организации становится недостаточным ограничиться лишь статической версией сайта. Динамические сайты позволяют пользователю взаимодействовать с ним, что привлекает их внимание к организации, дает возможности для общения, получения информации в более интересной форме.

Динамический сайт – сайт, состоящий из динамических страниц – шаблонов, контента, скриптов и прочего.

Для генерации содержимого на стороне клиента разработчики применяют язык программирования JavaScript, с помощью которого возможно реализовать как малозначительные операции с содержимым, так и полноценные приложения и последовательности.

Изначально JavaScript был создан для того, чтобы сделать страницы сайта «живыми», то есть позволить им реагировать на совершаемые действия пользователем сайта. Однако сейчас на данном языке возможно написание

программ не только для браузера.

Программы на JavaScript называются скриптами и подключаются напрямую к странице, написанной на HTML. JavaScript использует элементы страницы как объекты и позволяет совершать с ними разнообразные действия.

Существует два подхода к использованию кода: Компиляция и интерпретация.

При компиляции код, написанный на любом языке, загружается в другую программу, называемую компилятором, где переводится в машинный код, который затем распространяется и запускается. Исходный код программы остается у разработчика.

При интерпретации исходный код, получаемый программой интерпретатором, распознается как есть и выполняется. При этом распространяется сам исходный код. Этот подход применяется для языка JavaScript. А интерпретатором выступает браузер пользователя.

Современные интерпретаторы преобразуют JavaScript в машинный код или близко к нему, оптимизируют, а затем выполняют. Поэтому программы, написанные на этом языке, выполняются достаточно быстро.

Использование языка JavaScript имеет ряд преимуществ и недостатков.

Среди преимуществ можно выделить безопасность этого языка, так как он был создан для работы в браузерах, где не требуется доступ к операционной системе, памяти и процессору. Также этот язык поддерживается всеми браузерами и включен в них по умолчанию. Так что все программы, написанные на этом языке, запустятся и выполнятся в окне браузера пользователя без дополнительных требований, установок и ограничений. Язык является довольно легким в использовании – простые вещи делаются просто. Также полностью интегрирован с CSS и HTML, что дает полную свободу в работе с элементами страницы.

Язык очень быстро развивается, набирает все большую популярность. Это приводит к появлению всесторонних помощников, готовых разработанных решений, специальных библиотек языка и фреймворков.

Библиотека – структурированный набор полезного функционала. Библиотеки содержат множество функций для работы с различными элементами: строками, датами, событиями, анимацией, запросами и другим. Каждая функция возвращает программе полученное значение, которое впоследствии используется в зависимости от логики разработчика. Это позволяет разработчику не создавать функции, которые часто используются при программировании, а взять уже готовую.

Фреймворк – каркас приложения. Он определяет логику написания программы, в которую разработчик добавляет свои элементы, не нарушая ту самую логику.

Одной из самых популярных библиотек по праву является jQuery. Данная библиотека содержит функциональность, полезную для максимально-широкого круга задач. Однако, разработчики библиотеки не стремились к созданию максимально возможного числа функций, так как это привело бы к увеличению размера дистрибутива и огромному количеству кода, который был бы не востребован при использовании только некоторых возможностей программистом. Для этого была реализована структура универсального ядра библиотеки, к которой возможно подключить плагины с необходимым набором полезных функций. Это позволило собрать для ресурса именно ту необходимую функциональность, востребованную для реализации заданных целей.

Так как JavaScript также является текстовым документом, для него будет целесообразно использовать тот же текстовый редактор Visual Studio Code.

Также необходимым инструментом разработчика при реализации приложения является отладчик. Для реализации отладки все современные браузеры содержат «инструмент разработчика». С ними намного проще и быстрее происходит реализация исправления ошибок.

На текущий момент самый многофункциональный инструментом разработчика создан в браузере Google Chrome. Он изображен на рисунке 2.7.

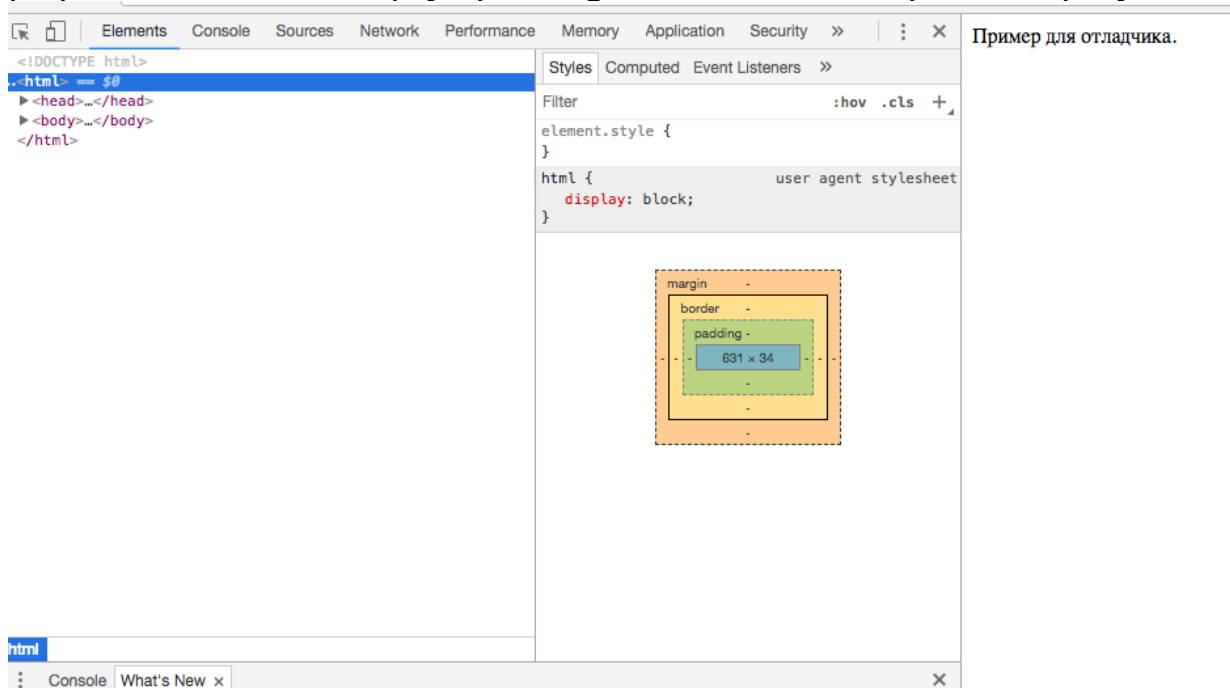


Рисунок 2.7 – Окно инструмента разработчика Google Chrome

Рабочая область отладчика делиться на 3 зоны: зона исходных файлов, зона текста, зона информации и контроля.

На вкладке верхнего меню расположены переходы к различным окнам отладчика. Самыми часто используемыми при разработке являются окно Elements, Console и Sources.

На вкладке Elements (Рисунок 2.7) отображается код HTML каркаса.

Наводя на каждый элемент можно посмотреть его расположение на странице, применяемые к нему стили CSS, заданные размеры отображаются в виде наглядного блока, разделенного на основные зоны элемента. Инструмент разработчика позволяет вносить изменения на страницу, добавлять или отменять стилевые особенности элемента для того, чтобы определить, как изменится результирующая страница при данных изменениях.

На вкладке Console, изображенной на рисунке 2.8, возможно запускать команды JavaScript, выводить отладочную информацию при выполнении скриптов. Очень полезный инструмент разработчика, в котором можно посмотреть реакцию на определенные функции, проводить работу с состоянием браузера и его элементов, просматривать сообщения об ошибке.

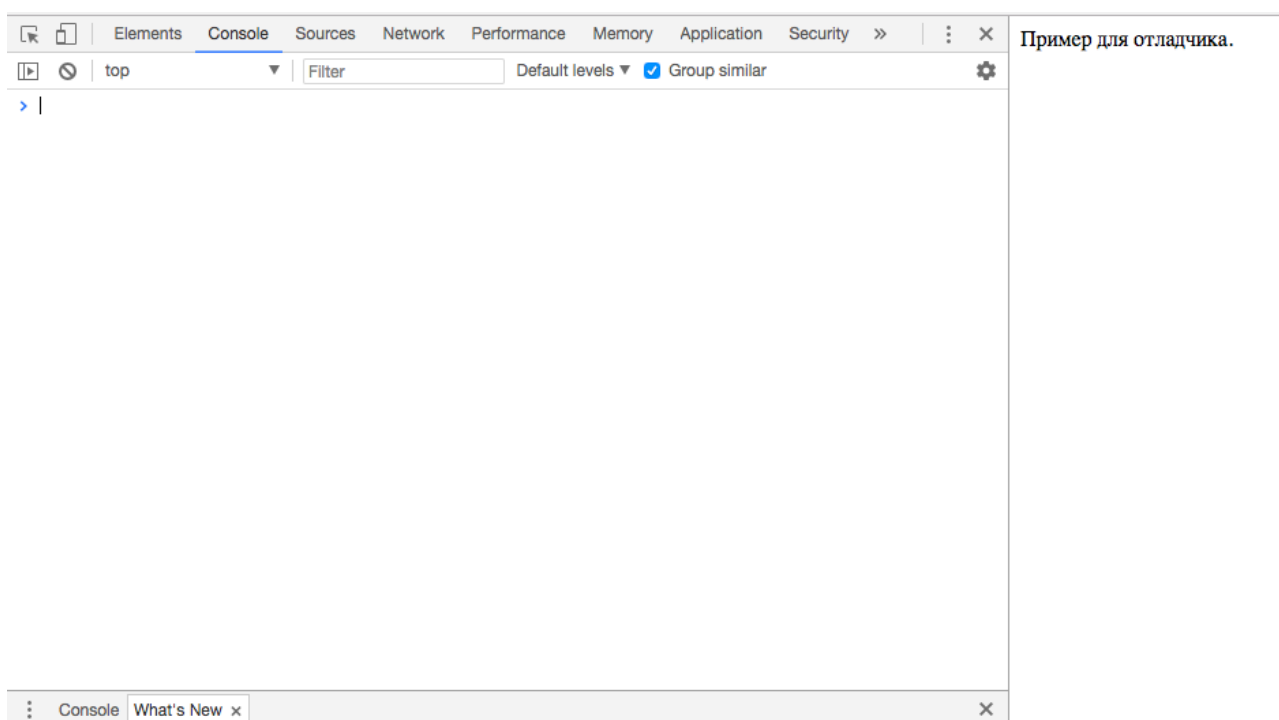
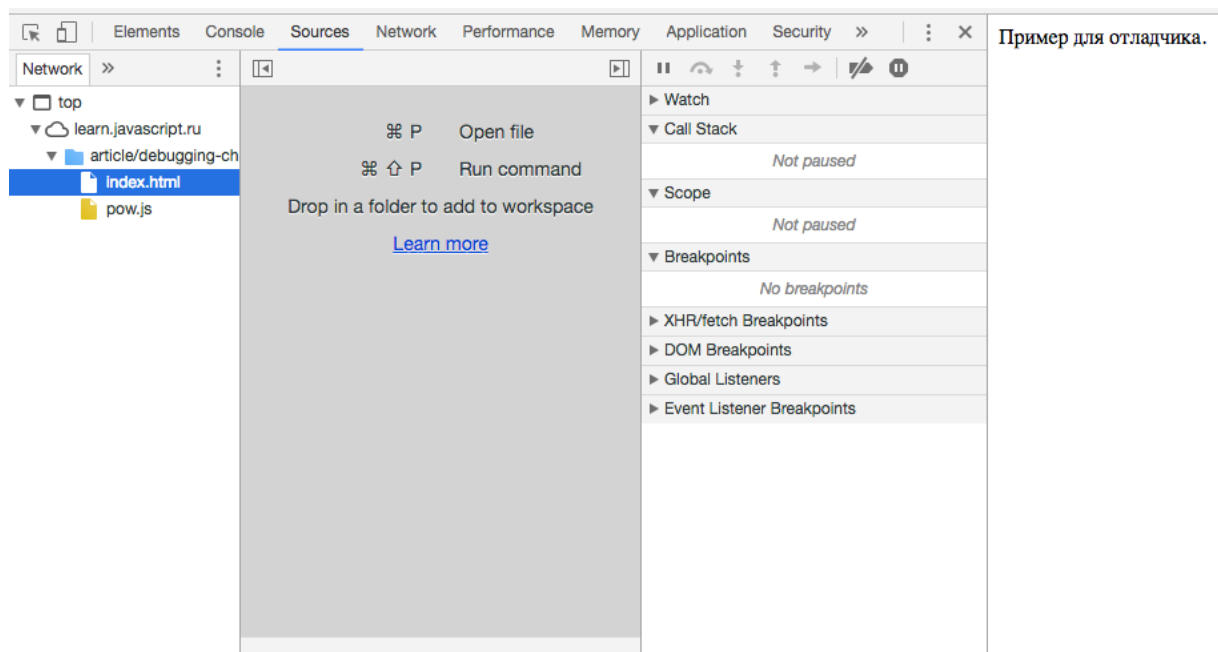


Рисунок 2.8 – Панель Console инструмента разработчика

Вкладка Sources, изображенная на рисунке 2.9, располагает в себе отображение всех файлов, подключенных к HTML документу. В основном используется для отладки работы скриптов.

Открыв файл со скриптами, возможно отслеживать последовательность их отработки, результаты каждого шага, текущее значение переменных. Выбирая так называемые точки останова, возможно приостановить в выбранном вами месте действие скрипта, посмотреть, как меняется страница с каждым шагом отработки последующей команды или же наблюдать состояние страницы в определенный момент.



Пример для отладчика.

Рисунок 2.9 – Вкладка Sources инструмента разработчика

ГЛАВА 3

РАЗРАБОТКА ВЕБ-САЙТА И ВЕБ-ПРИЛОЖЕНИЯ

3.1 Разработка макета сайта и приложения

Разработка макета делится на два этапа. Первым этапом является разработка функционального макета. Он будет определять расположение функциональных блоков: навигация, текстовые поля, размещение логотипов, наличие форм ввода, графические и анимационные блоки.

Разработка будет производиться с использованием программного продукта Adobe Photoshop с использованием инструментов векторной графики, которые позволяют масштабировать изображение, перемещать и производить другие манипуляции без потери качества исходного изображения.

Процесс создания макета протекает по принципу определения функциональных блоков от большего к меньшему. Сначала будет разбиваться вся страница на основные блоки. Таких блоков в современном подходе к структурированию сайта выделяется три: шапка, основная часть и подвал.

Шапка страницы отображается в самом верху страницы. Она практически не содержит информационной нагрузки.

Далее следует главная часть страницы, располагающая в себе всю необходимую информацию и функциональные блоки.

В самом низу располагается подвал. Это дополнительный блок, в котором может быть расположена некоторая незначительная информация, как номера телефонов или адреса, которая обычно дублирует какой-то раздел в кратком содержании.

Данный шаблон представлен на рисунке 3.1. Этот шаблон будет использоваться при создании всех страниц сайта. Элементы шапка и подвал будут оставаться неизменными по своему размеру, а главная часть остается динамической и зависит от размера контента, который будет варьироваться исходя из назначения страницы.

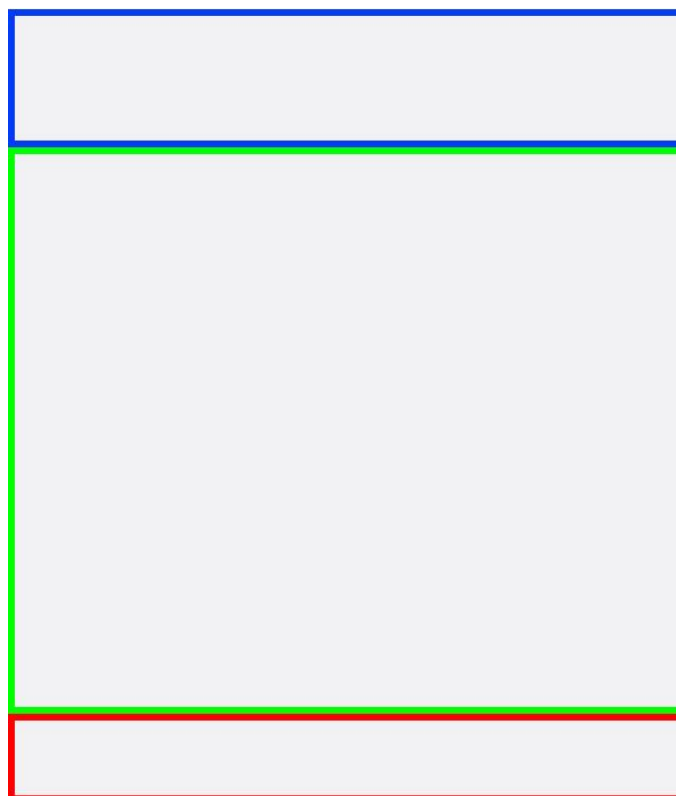


Рисунок 3.1 – Макет основных блоков страницы.

На следующем этапе отдельно составляются макеты каждого из элементов основного.

Шапка сайта будет наполнена логотипами организации, блоком навигации, блоком ссылок в социальные сети и сторонние ресурсы организации. Макет шапки представлен на рисунке 3.2.

На макете размечены элементы:

- 1 и 2 элемента - это область логотипов организации;
- 3 элемент – блок ссылок на сторонние ресурсы;
- 4 элемент – блок навигации по сайту.

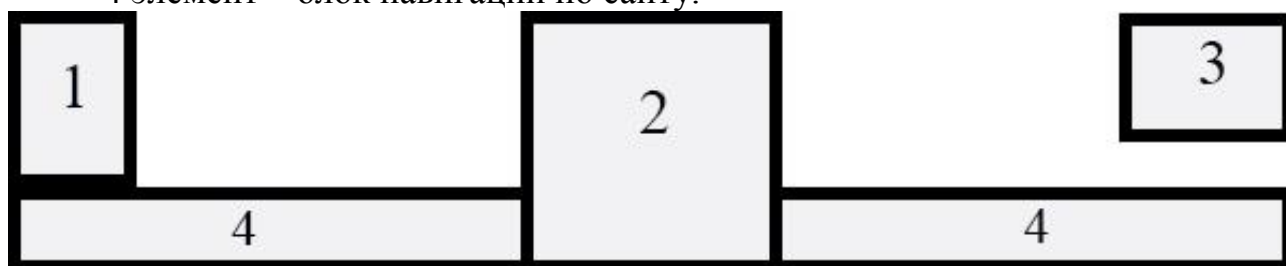


Рисунок 3.2 – Макет шапки сайта

На основе полученного макета будет строиться шапка сайта и создаваться ее пользовательский дизайн.

Подвал сайта располагает в себе логотип организации и краткую информацию с контактами. Так как во владениях организации находятся две кофейни, информацию с контактами целесообразно разделить на два блока, отражающей информацию о каждой кофейне в отдельности.

Реализация макета подвала сайта представлена на рисунке 3.3.

Элементы, представленные на макете:

- 1 – область логотипа организации;
- 2 и 3 – блоки контактной информации.



Рисунок 3.3 – Макет подвала сайта

Главный блок каждой страницы несет в себе различную информацию и имеет различные элементы. Поэтому макет главного блока сайта создается отдельно для каждой страницы данного сайта.

Согласно разработанному проекту сайта, будет создано 5 различных страниц:

- главная страница сайта. Располагает в себе виджеты с изображениями, разделяющими две кофейни, название данной кофейни и кнопки перехода к более подробной информации о выбранной кофейне.
- страница бронирования. Содержит в себе приложение, осуществляющее бронирование клиентами.
- страница контактной информации. Содержит в себе полный список контактов и адресов, кнопку перехода к просмотру на карте по каждой из кофеен.
- две страницы подробной информации о каждой из кофеен. Содержат описание каждой кофейни, зону фотогалереи, реализованную в виде слайдера, блок с интерактивной картой, в которой маркером отмечено расположение заведения.

Макет главной страницы изображен на рисунке 3.4.

На макете расположены следующие элементы:

- 1 и 2 – элементы изображения, отражающего кофейню.
- 3 и 4 – название кофейни
- 5 и 6 – кнопка перехода к странице, описывающей каждую отдельную кофейню.

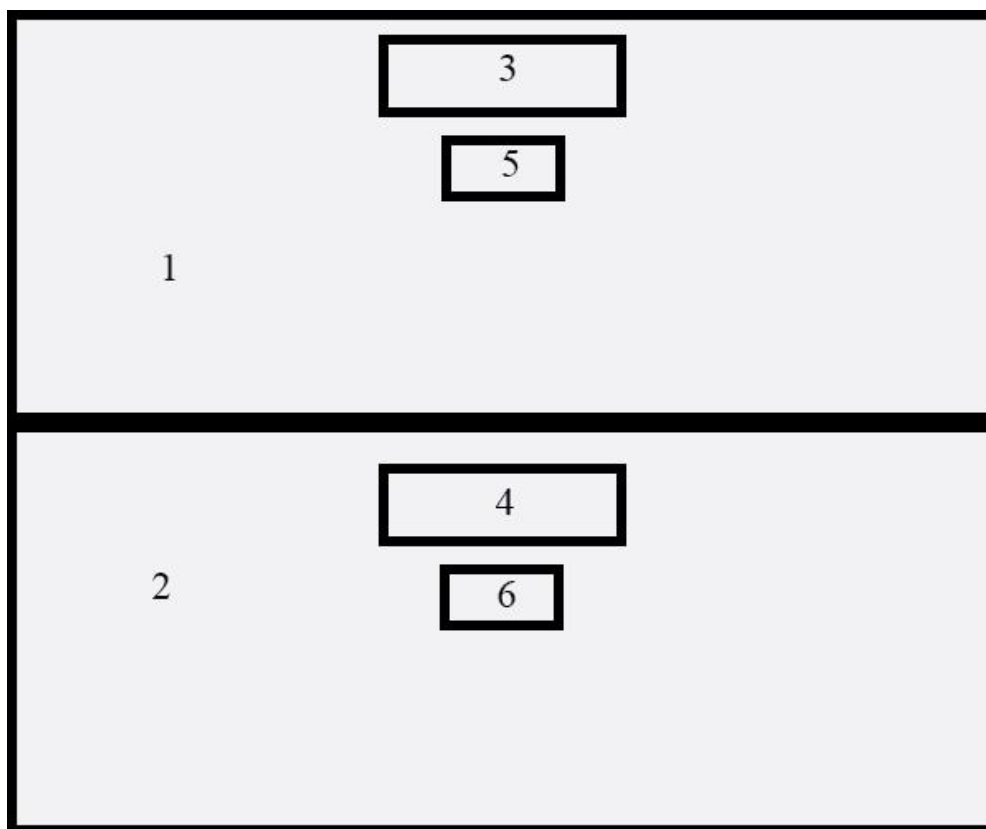


Рисунок 3.4 – Макет главной страницы сайта

Макет страницы бронирования изображен на рисунке 3.5.

Он содержит в себе поле самого приложения (1) и поле, в которое будет динамически добавляться результат выполнения действий пользователя(2).

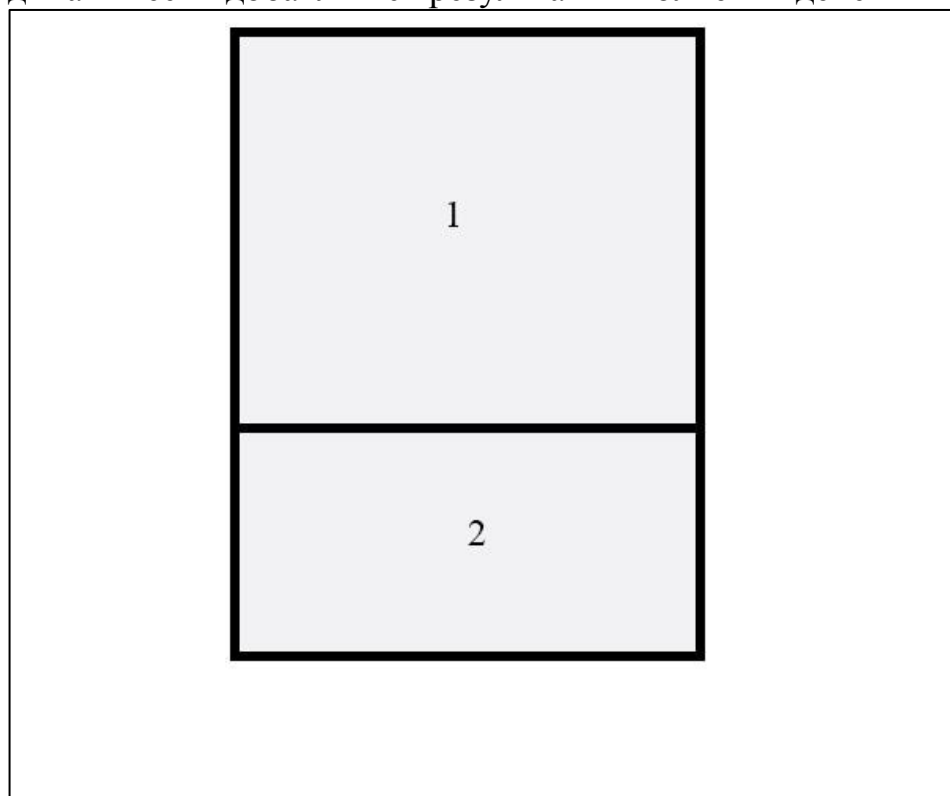


Рисунок 3.5 – Макет страницы бронирования

Макет страницы контактной информации изображен на рисунке 3.6.

Содержит в себе следующие блоки:

- 1 и 2 – область, отделяющая контактную информацию по каждой из кофеен;
- 3 и 4 – область, в которую будет помещен текст с контактной информацией;
- 5 и 6 – кнопка для перехода к интерактивной карте.

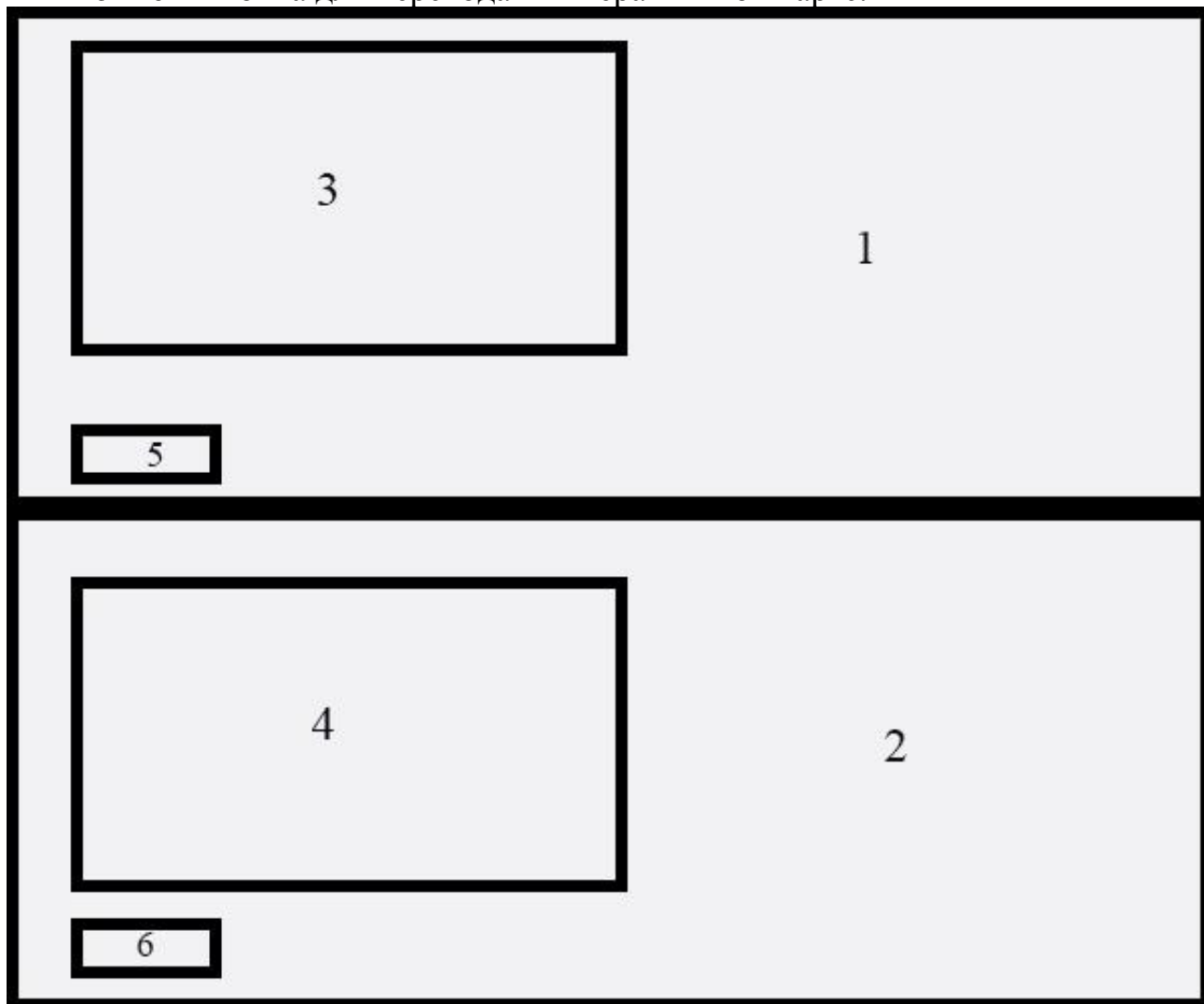


Рисунок 3.6 – Макет страницы контактной информации

Структура макетов двух страниц с подробной информацией о каждой из кофеен по определению строиться, с использованием схожего содержимого и функциональных элементов. Следовательно, целесообразно будет разработать один макет, для двух страниц.

Макет этих страниц представлен на рисунке 3.7.

На макете изображены:

- 1 – блок с текстовой информацией;
- 2 – блок с графической галереей, выполненной в виде слайдера;
- 3 – блок интерактивной карты с меткой расположения.

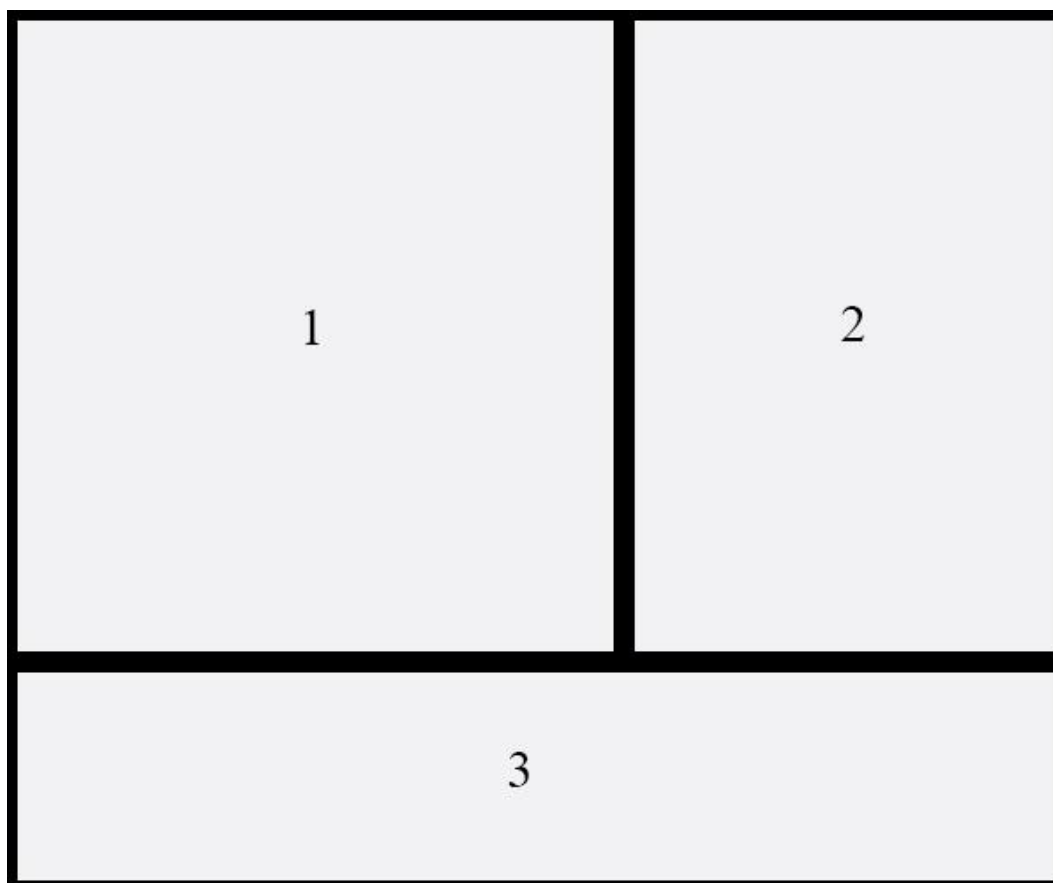


Рисунок 3.7 – Макет страниц с информацией о кофейнях

Далее необходимо отдельно разработать функциональный макет приложения. Он также будет располагать в себе функциональные блоки, реализующие заданные параметры.

В приложении, осуществляющее бронирование, необходимы следующие функциональные блоки:

- поле формы заполнения информации о брони;
- блок функциональных клавиш;
- поле вывода информации.

Поле формы заполнения информации о брони будет разделено на подблоки, назначение которых определяется требованиям к информации, вносимой клиентом данного приложения, для осуществления корректного выполнения функций приложения. В их числе:

- поле выбора адреса заведения;
- поле информации о клиенте;
- поле информации о брони.

Макет приложения показан на рисунке 3.8.

В состав макета входят следующие блоки:

- 1 – поле формы заполнения информации о брони;
- 2 – блок функциональных клавиш;
- 3 – поле вывода информации;

- 4 – поле выбора адреса заведения;
- 5 – поле информации о клиенте;
- 6 – поле информации о брони.

Рисунок 3.8 – Макет приложения бронирования

На этом этап разработки функционального дизайна считается завершенным. За ним следует этап разработки пользовательского дизайна.

На этапе разработки пользовательского дизайна разрабатывается макеты страниц сайта и приложения в том виде, в котором их уже будет видеть непосредственно пользователь. Разработчик определяет:

- дизайн элементов, помещаемых в функциональные блоки;
- размер этих элементов, а соответственно и размер блоков;
- используемые шрифты;
- используемую цветовую гамму;
- изображения, помещаемые на страницу.

Результатом разработки пользовательского дизайна будет являться изображение веб-сайта со всеми элементами – то, каким он будет отправлен верстальщику.

Дизайнер совместно с заказчиком утверждает используемые изображения и шрифты, определяет цветовую гамму сайта.

В качестве основного шрифта было решено использовать «PT Sans».

Цветовая гамма выполнена в бежевых тонах, отражающая цвет кофейных бумажных фильтров и цвет светло-обжаренного кофе.

В результате разработки пользовательского дизайна потребуется создать 5 макетов страниц сайта. Количество макетов соответствует количеству страниц сайта. И отдельно макет приложения, который будет являться частью макета страницы брони.

Макет главной страницы показан на рисунке 3.9

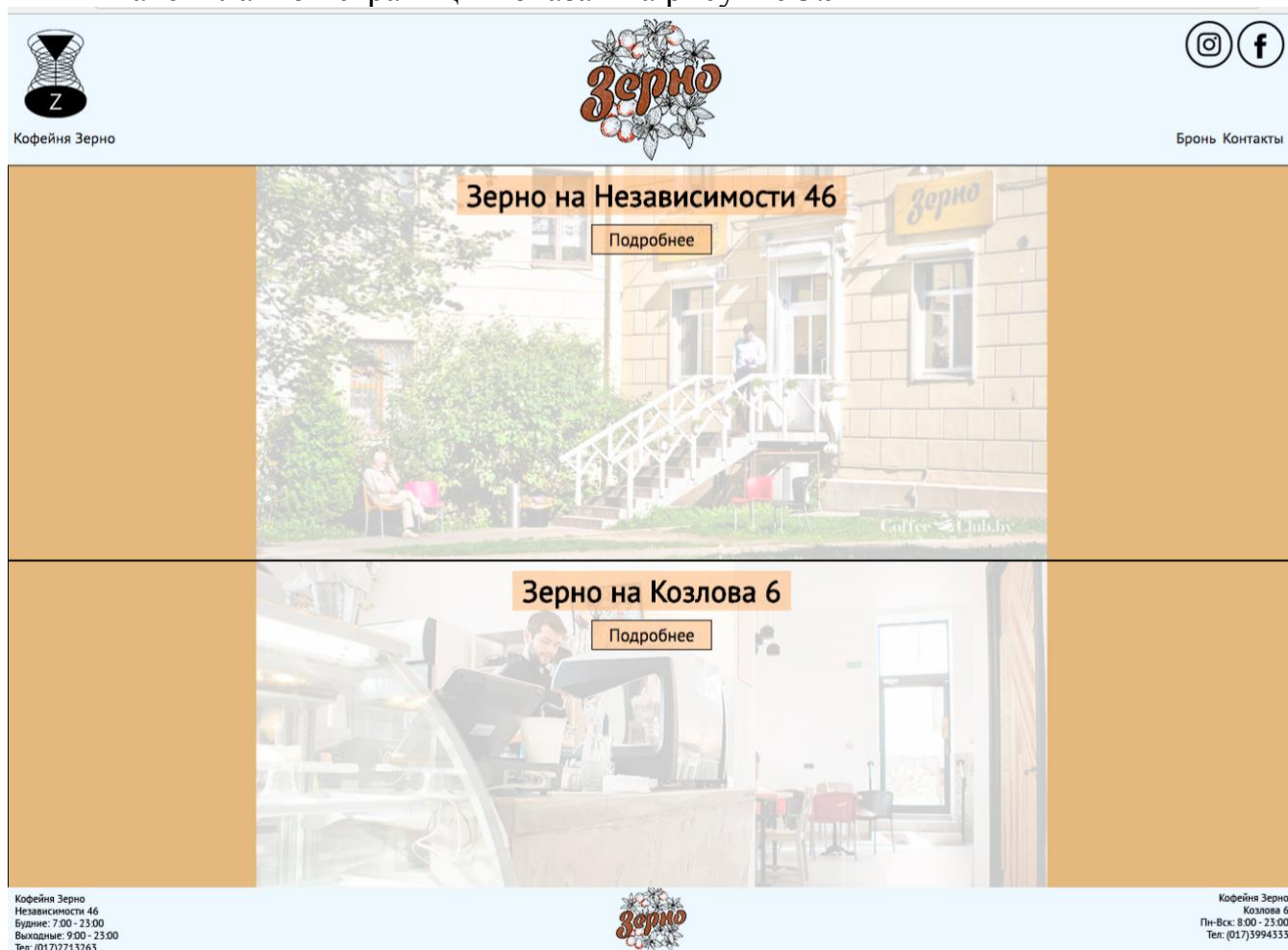


Рисунок 3.9 – Макет главной страницы

Макет веб-приложения представлен на рисунке 3.10.

Макет веб-приложения для бронирования. Форма имеет белый фон с тонкой серой обводкой. Вверху — выпадающий список с текстом 'Выберите кофейню' и стрелкой вверх-вниз. Ниже — две текстовые поля: 'Укажите имя' и 'Укажите телефон'. Затем — поле для даты с текстом 'Введите дату mm/dd/yyyy' и иконкой календаря, и поле для времени с текстом '-- : --'. В самом низу — широкая зеленая кнопка с белым текстом 'Отправить'.

Рисунок 3.10 – Макет веб-приложения

Макет веб-приложения с динамической областью результата выполнения программы представлен на рисунке 3.11.

The mockup shows a web form with the following elements:

- A dropdown menu at the top with the selected option "Независимости 46".
- Two input fields: one containing "Алексей" and another containing "296718117".
- Two more input fields: one containing "05/26/2018" with a calendar icon, and another containing "14 : 00".
- A large green button labeled "Отправить" (Send).
- A result box below the button displaying: "05/26/2018 14:00", "Алексей 296718117", and "Независимости 46".

Рисунок 3.11 – Макет веб-приложения

Затем разработаем дизайн макета страницы брони, на котором располагается данное веб-приложение. Оно показано на рисунке 3.12.

The mockup shows a full page layout for coffee booking. The header is light blue and contains:

- Logo "Кофейня Зерно" on the left.
- Decorative "Зерно" logo in the center.
- Social media icons (Instagram and Facebook) on the right, with the text "Бронь Контакты" below them.

The main content area has a tan background and features the same booking form as in Figure 3.11, centered on the page.

The footer is light blue and contains contact information for two locations:

- Left side: "Кофейня Зерно", "Независимости 46", "Будние: 7:00 - 23:00", "Выходные: 9:00 - 23:00", "Тел: (017)2713263".
- Right side: "Кофейня Зерно", "Козлова 6", "Пн-Вск: 8:00 - 23:00", "Тел: (017)3994333".

Рисунок 3.12 – Макет страницы бронирования

Макеты страниц с описанием о каждой кофейне отображены на рисунках 3.13 и 3.14.



Рисунок 3.13 – Макет страницы сайта

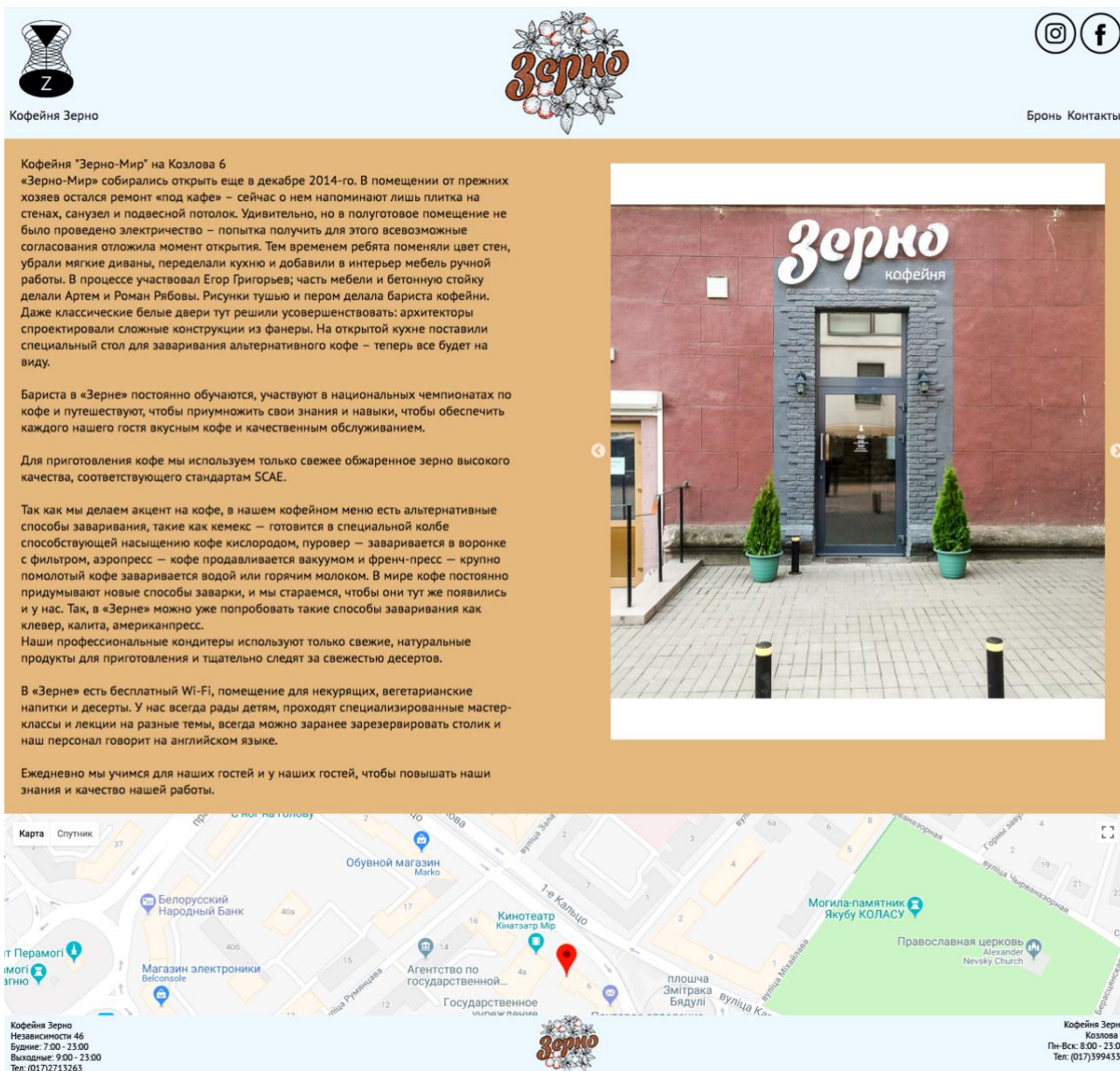


Рисунок 3.14 – Макет страницы сайта

И макет последней страницы сайта с контактной информацией изображен на рисунке 3.15.

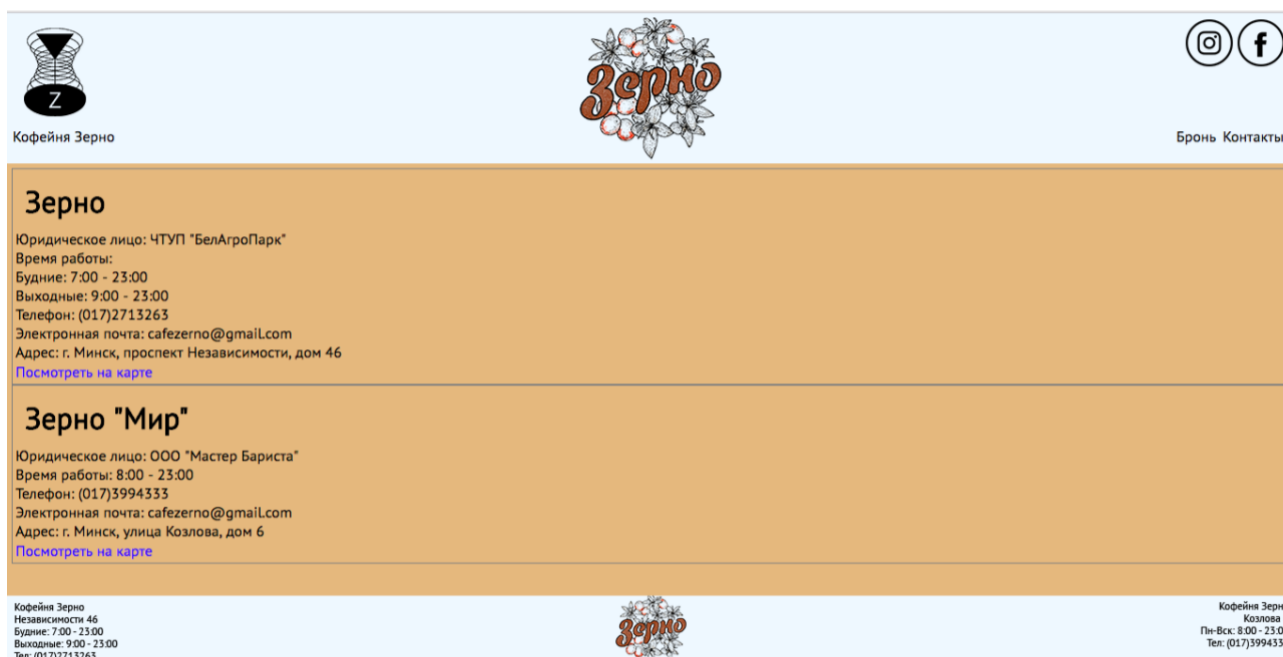


Рисунок 3.15 – Макет страницы контактов

Когда все макеты готовы, дизайнер отправляет их разработчику на стадию верстки.

Все макеты отправляются в формате, поддерживающем сохранение слоев, для упрощения процесса анализа макетов верстальщиком. Верстальщик должен разбить их на составляющие части и начать процесс построения элементов в HTML-документе.

3.2 Верстка макетов веб-сайта и веб-приложения

Верстальщик получает макеты, созданные на этапе дизайнерской разработки. Все макеты отправляются в формате, поддерживающем сохранение слоев, для упрощения процесса анализа макетов верстальщиком. Верстальщик должен разбить их на составляющие части и начать процесс построения элементов в HTML-документе.

В верстке применяются следующие программные продукты:

- текстовый редактор Visual Studio Code для написания HTML документов и составления каскадных таблиц стилей CSS;
- веб-браузер Google Chrome для отображения результата созданных документов;
- инструмент разработчика браузера Google Chrome для отладки и более быстрого внесения корректив в созданные документы.

Коды документов гипертекстовой разметки расположены в приложении А данной дипломной работы, а текст каскадных таблиц стилей CSS располагается в приложении Б.

Результаты верстки проверяются на кросс-браузерность в различных

браузерах.

Для примера на рисунках 3.16, 3.17 представлен результат отображения главной страницы в браузерах Google Chrome и Safari.

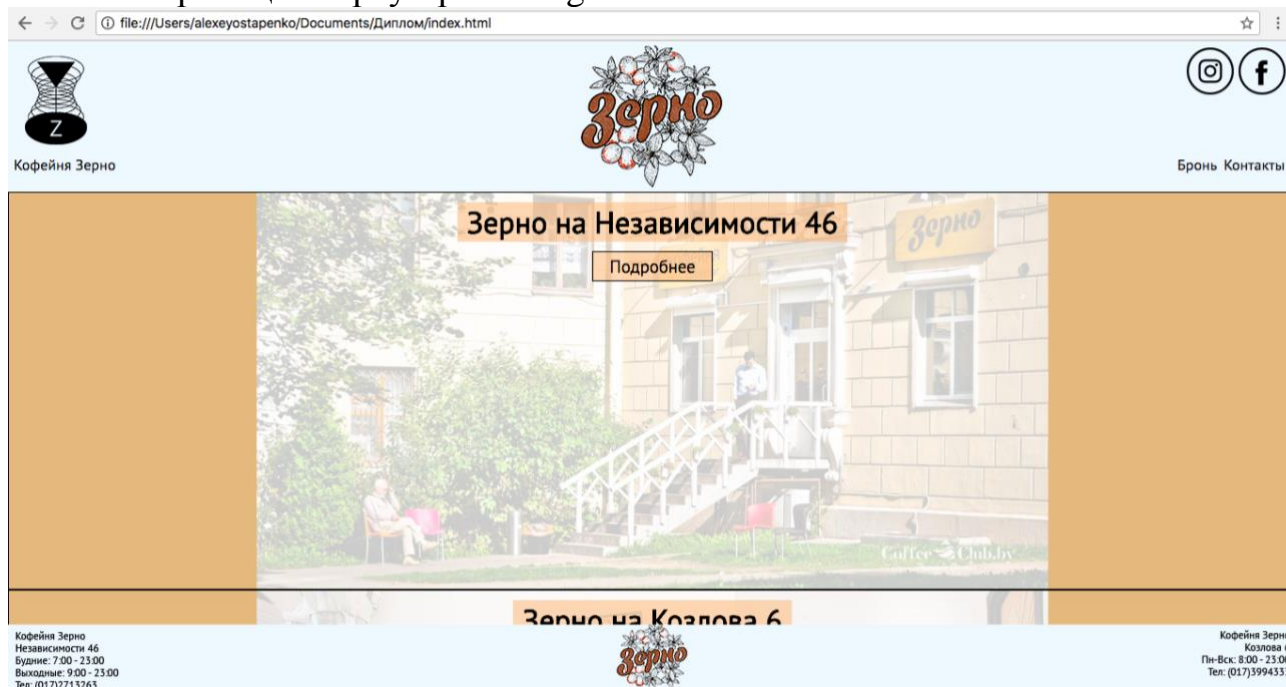


Рисунок 3.16 – Отображение главной страницы в браузере Google Chrome

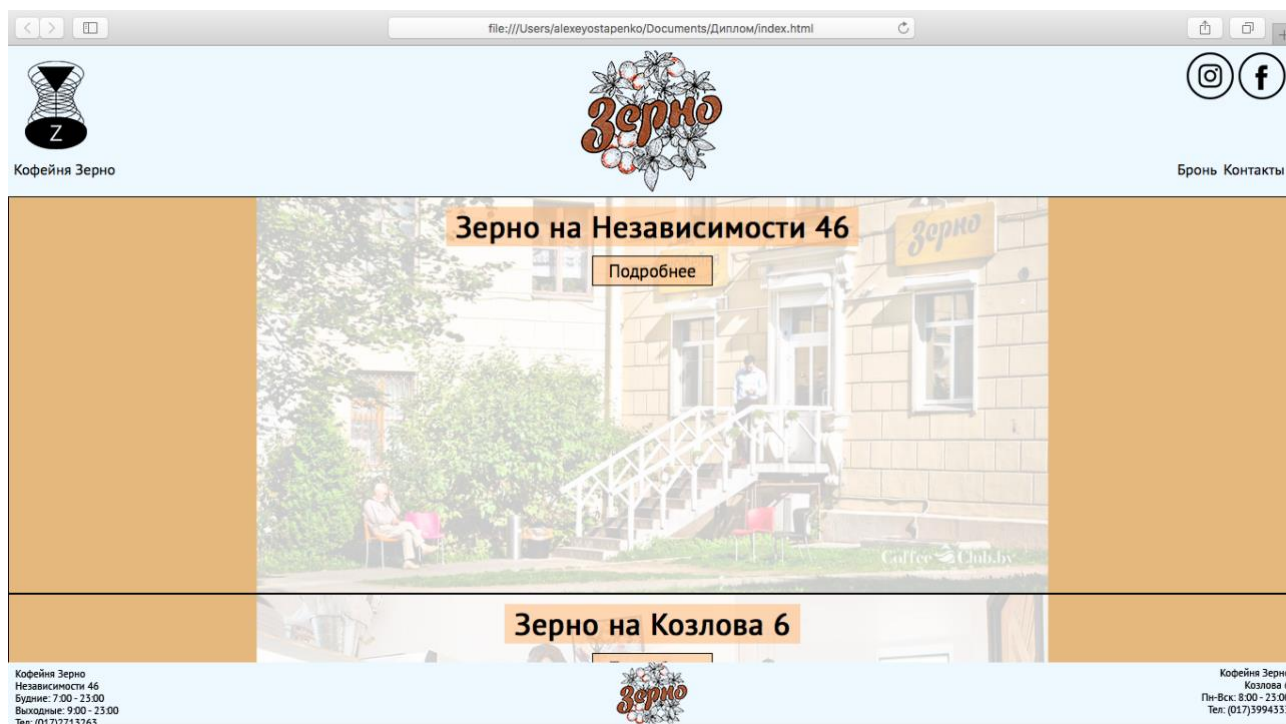


Рисунок 3.17 – Отображение главной страницы в браузере Safari

Из примера следует вывод, что страницы выглядят идентично в разных браузерах и результат их отображения свидетельствует о прохождении требований проверки верстки на кросс-браузерность.

Верстка интерфейса веб-приложения является идентичной верстке элементов любой веб-страницы. Разработчик же кода самого приложения будет

добавлять необходимые ему атрибуты, значения и свойства элементов верстки макета веб-приложения. Именно поэтому хороший разработчик JavaScript должен также знать основы верстки и составления таблиц стилей.

3.3 Разработка кода веб-приложения

На этапе разработки кода для динамических страниц сайта и веб-приложения будут использоваться следующие программные продукты:

- текстовый редактор Visual Studio Code для написания JavaScript кода и работы с документами HTML и CSS;
- веб-браузер Google Chrome для отображения результата работы скриптов и приложения;
- инструмент разработчика браузера Google Chrome для отладки и более быстрого внесения корректив в созданные документы.

Программный код страниц сайта и веб-приложения размещен в приложении В.

Дополнительно с написанием чистого кода на языке JavaScript разработчики прибегают к использованию специальных библиотек, плагинов и готовых решений. Это позволяет сократить время разработки, составлять более краткий код, но не терять читаемости и понимание кода, а иногда и улучшать эти характеристики.

Разработка кода веб-сайта и веб-приложения начинается с анализа проекта. Разработчик изучает требования, предъявленные к функционалу сайта и приложения, устанавливает необходимые инструменты, определяет этапы работы.

К динамическим страницам сайта были выдвинуты следующие требования:

- реакция на наведения мышью на навигационные элементы и ссылки. Среди них: добавление рамки к элементам навигации при наведении мышью, реакция изменения цвета кнопок при наведении мышью, изменение курсора.
- реализация галереи в виде слайдера с возможностью прокрутки стрелками, мышью или пальце на сенсорном устройстве.
- добавление интерактивной карты Google с метками на адреса кофейни.

К приложению заказчик составил следующие требования:

- проверка заполнения формы;
- цветовое выделение пустых элементов формы;
- создание выпадающего календаря в поле ввода даты;
- запись данных в LocalStorage браузера;
- загрузка сохраненных данных из LocalStorage;

- отправка данных формы на почтовый ящик организации.

В качестве базы данных приложения используется LocalStorage браузера. Это позволяет экономить серверное место, данные определенного пользователя хранятся лишь у него на устройстве, и никто не сможет получить его данные. Также сами хранимые данные не имеют большого объема и не займут много свободного места в памяти.

LocalStorage – постоянное локальное хранилище данных. Оно обеспечивает хранение пользовательских данных непосредственно у самого клиента по принципу использования схемы «ключ-значение». Если данного типа хранения недостаточно, возможно на основе создать свою базу данных, придумать свой формат файлов и применить много иных способов. Данное хранилище было введено с появлением стандарта HTML5. Данные сохраняются даже при закрытии страницы сайта, закрытии вкладки браузера или самого браузера. В отличие от куки-технологии, данные из локального хранилища не передаются веб-серверам, если сам разработчик не установит функцию их отправки.

В процессе разработки веб-сайта и веб-приложения использовались дополнительные инструменты.

При написании кода для работы галереи, реализованной в виде слайдера была использована библиотека jQuery и плагин Slick Slider.

jQuery – библиотека JavaScript, фокусирующаяся на взаимодействии кода с документами HTML. Она помогает легко получить доступ к любому элементу страницы и манипулировать ими.

Slick Slider – это плагин библиотеки jQuery, позволяющий создать адаптивные слайдеры любой сложности. Плагин имеет широкий функционал. Среди основных его функций следует выделить:

- добавление автопрокрутки с настраиваемой скоростью;
- функция перелистывания слайдера мышью или пальцем на сенсорном экране устройства;
- добавление анимации переходов;
- закливание слайдов;
- выбор типа подгрузки слайдов.

К слайдеру возможно применять различные методы, настраивать события.

Для подключения интерактивной карты использовался программный интерфейс приложения или интерфейс прикладного программирования API. Он представляет из себя набор готовых классов, процедур, функций, структур и констант, предоставляемый приложением, сервисом или операционной системой для использования во внешних программных продуктах.

В HTML5 предусмотрены дополнительные типы элементов формы, позволяющие пользователю вводить только дату, номер телефона, почтовый

адрес и прочее. Однако данные стандарты могут поддерживаться не всеми браузерами, да и часто более старые версии одного браузера могут не поддерживать функционал, поддерживаемый в более новом релизе. Поэтому есть уже разработанные решения с открытым кодом, которые возможны к использованию.

В данном проекте был использован кросс-браузерный выпадающий календарь для поля ввода формы. Вид данного календаря представлен на рисунке 3.18.

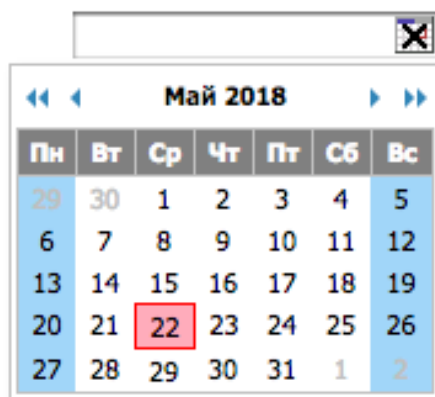


Рисунок 3.18 – Кросс-браузерный календарь

Он представляет из себя удобное изображение классического календаря, открывается по нажатию иконки в крайнем углу поля ввода. При его открытии он всегда указывает на текущую дату, имеет два языка: английский и русский. В текстовое поле дата всегда записывается в едином формате «месяц/день/год», что удобно для последующей работы со значением данного поля.

Результат работы веб-приложения по отправке брони представлен на рисунке 3.19.

При отправке файлов веб-приложение использует установленное на устройстве программное обеспечение по работе с электронной почтой.

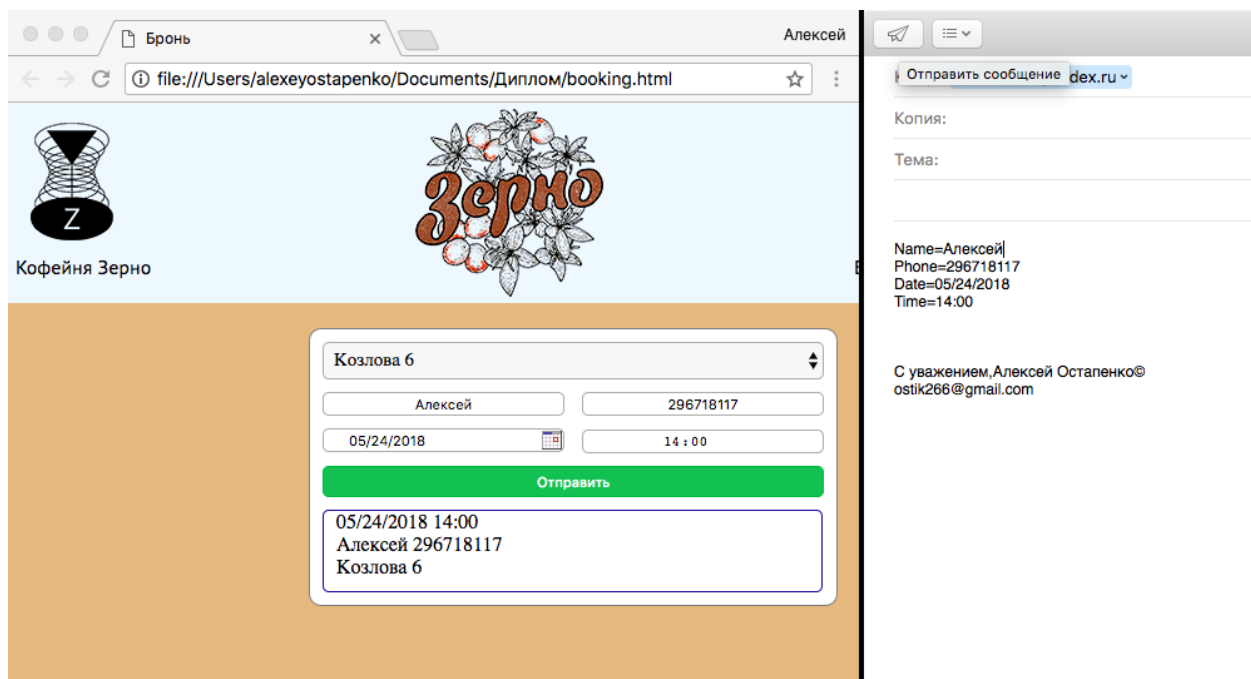


Рисунок 3.19 – Работа приложения бронирования

3.4 Расчет технико-экономической эффективности проекта по внедрению веб-сайта и веб-приложения в организации «Кофейня «Зерно»

В проекте намечены пути использования скрытых ресурсов для решений поставленных целей и задач, устранения недостатков в работе, обосновываются и принимаются решения, которые полезно применить в практике менеджмента.

Для оценки проекта рассчитываются следующие показатели:

- цена продукта (Ц);
- дополнительные доходы от реализации программного продукта (Д_{доп});
- экономическая эффективность (прибыль) (Э);
- рентабельность проекта (Р);
- срок окупаемости продукта (Т_{ок}).

$$Ц = З + П + НДС \quad (3.1),$$

где З – сумма затрат, П – прибыль, составляющая процент от себестоимости продукта, определяемый организацией, НДС – налог на добавленную стоимость.

$$Э = Д_{доп} - Ц \quad (3.2);$$

$$Р = П/Ц * 100\% \quad (3.3);$$

$$Т_{ок} = Ц/П \quad (3.4).$$

График выполнения работ представлен в таблице 3.1.

Таблица 3.1 – График выполнения работ

Наименование работ	Длительность работ	
	В днях	В часах
Разработка ТЗ	1	8
Планирование веб-продукта	3	24
Разработка	10	80
Тестирование и отладка	5	40
Оценка результатов	2	16
итого	21	168

Таким образом, на разработку программного продукта затрачено 168 человеко-часов при восьмичасовом рабочем дне.

При расчете затрат учитываются следующие виды расходов:

- материальные затраты;
- затраты на оплату труда;
- отчисления на социальные нужды;
- амортизация;
- прочие затраты.

Список материальных затрат с их стоимостью приведен в таблице 3.2

Таблица 3.2 – Перечень материальных затрат

Наименование	Единица измерения	Количество	Цена за единицу, руб	Стоимость, руб
Специальная литература	шт.	3	25	75
Программное обеспечение	шт.	1	20	20
Канцтовары	шт.	2	1	2
Бумага формата А4	упаковка	1	8	8
Итого:				105
Транспортные расходы (10% от итога)				10,5
Итого:				115,5

Расчет затрат на оплату труда (ЗОТ) производится с учетом заработной ставки почасовой специалистов (ЗС) и количества затраченных человека-часов

(ЧЧ).

$$ЗОТ = ЗС * ЧЧ \quad (3.5).$$

$$ЗОТ = 5 * 168 = 840 \text{ руб.}$$

Таким образом затраты на оплату труда составили 1680 рублей.

К отчислениям на социальные нужды (ОСН) относятся отчисления в фонд заработной платы (34%) и отчисления в пенсионный фонд (1%). Итого, отчисления на социальные нужды составляют 35% от фонда заработной платы.

$$ОСН = ФЗП * 0.35 \quad (3.6).$$

$$ОСН = 840 * 0.35 = 294 \text{ руб.}$$

Расходы на амортизацию в проекте по разработке программного продукта можно опустить в связи с небольшим сроком осуществления работ и применением лишь одной единицы оборудования – персональный компьютер.

Прочие расходы – дополнительные расходы, сопутствующие основному производству, но не связанные напрямую. Они направлены для обеспечения процесса производства и обращения.

Прочие расходы ($P_{пр}$) составляют 20% от суммы фонда заработной платы.

$$P_{пр} = ФЗП * 0,2 \quad (3.7).$$

$$P_{пр} = 840 * 0.2 = 168 \text{ руб.}$$

В таблице 3.3 представлена итоговая смета затрат

Таблица 3.3 – Итоговая смета затрат

Наименование статей расходов	Сумма затрат, руб
Материальные затраты	115.5
Затраты на оплату труда	840
Отчисления на социальные нужды	294
Амортизация	-
Прочие расходы	168
Итого:	1417.5

Цена программного продукта определяется итоговыми затратами, прибылью, которая составляет 30% от фонда заработной платы и налога на добавленную стоимость, составляющего 20 процентов от суммы затрат и прибыли от создания и реализации программного продукта.

$$Ц = 1417.5 + (0,3 * 840) + (0,2 * (1417.5 + (0,3 * 840))) = 2003,4 \text{ руб.}$$

Оценка экономической эффективности заключается в сравнении результатов работы до и после мероприятия по введению в эксплуатацию программного продукта.

Введение программного продукта в кофейне «Зерно» решает две основные цели:

- повышение выручки за счет увеличения количества посетителей в следствии повышения информированности посредством сайта;

- повышение выручки за счет увеличения количества посетителей в следствии увеличения числа броней с помощью приложения.

Показатели взяты усредненными за период в 1 месяц до введения программного продукта и 1 месяц после внедрения программного продукта. Количество посетителей до внедрения программного продукта примем за 100%.

Результаты от внедрения программного продукта сведены в таблицу 3.4.

Таблица 3.4 – Результаты внедрения программного продукта

Наименование программного продукта	Количество посетителей, %		Доходы от деятельности, руб.		Изменение, %
	До	После	До	После	
Веб-сайт	100%	107%	20000	21500	107,5%
Веб-приложение	100%	103%	20000	20900	104,5%

В сумме получился рост посетителей на 10% и рост выручки, являющейся дополнительной выручкой, составил 2400 рублей, что является повышением на 12%.

Экономическая эффективность (прибыль) от внедрения программного продукта по формуле 3.2 составила

$$\text{Э(П)} = 2400 - 2003,4 = 396,6 \text{ руб.}$$

Рентабельность проекта составила

$$P = 396,6 / 2003,4 * 100 = 19,8\%.$$

Срок окупаемости составит

$$T_{\text{ок}} = 2003,4 / 396,6 = 5 \text{ месяцев}$$

Таким образом, срок окупаемости составит 5 месяцев или 150 дней.

ЗАКЛЮЧЕНИЕ

При выполнении работы была изучена организация общественного питания «Кофейня «Зерно», в частности, направление деятельности организации, бизнес-процессы, которые планируется автоматизировать внедрением программного продукта, исследована область проектирования и разработки веб-продуктов.

Были изучены основные виды услуг, оказываемыми кофейней «Зерно», среди которых:

- производство кондитерской продукции;
- производство свежесобранного зернового кофе;
- производство безалкогольных напитков, кофе, чая и напитков на основе алкоголя;
- реализация собственной продукции на территории предприятия общественного питания;
- организация потребления продуктов питания;
- организация отдыха, развлечений и досуга потребителей;
- предоставление продукции и услуг высокого качества;
- предоставление дополнительных сопутствующих услуг;
- обучение по направлениям бармен-официант, бариста.

Также была изучена стратегия развития предприятия, которая подразумевала следующие шаги:

- совершенствование оказания услуг;
- обеспечение потребности населения в производимой продукции и оказываемых услугах;
- повышение качества производства;
- выход на новые рынки сбыта продукции;
- расширение деятельности предприятия.

Были исследованы применяемые решения в сфере информационных технологий на предприятии, среди которых:

- продвижение в социальных сетях;
- собственный веб-сайт;
- существующие интернет ресурсы сторонних организаций.

Были разработаны рекомендации для повышения эффективности работы организации в области маркетинга и использования информационных технологий, так как было выяснено, что организация не использует собственных информационных технологий в сфере маркетингового продвижения бизнеса, а имеющиеся ресурсы повышают затраты.

Таким образом был сделан вывод о внедрении веб-сайта для маркетинговой деятельности организации и веб-приложения для осуществления автоматизации бизнес-процесса бронирования посетителями мест в заведениях.

В следствии полученных результатов исследования был выполнен анализ применяемых технологий для реализации проекта по созданию программных решений, изучены альтернативы.

Также была проведена работа по проектированию программных продуктов, их разработке и оценке экономической эффективности их внедрения и использования.

Затраты на введение программного продукта составили 2003,4 рублей. С учетом рентабельности в 19,8 процентов примерный срок окупаемости продукта составит примерно 5 месяцев.

Можно сделать вывод, что внедрение является целесообразным и экономически эффективным.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лекционный материал для студентов [Электронный ресурс] -Москва, 2012. – Режим доступа: <http://studopedia.su>. –Дата доступа: 15.04.2018.
2. Сайт сообщества людей, занятых в индустрии высоких технологий, «Хабр» [Электронный ресурс] – 2006. – Режим доступа: <https://habr.com/> – Дата доступа: 13.04.2018.
3. Электронный справочник «HTMLBook» [Электронный ресурс] – 2002. – Режим доступа: <http://htmlbook.ru/> - Дата доступа: 4.05.2018.
4. Электронный справочник «LearnJavaScript» [Электронный ресурс] – 2002. – Режим доступа: <http://learnjavascript.ru/> - Дата доступа: 4.05.2018.
5. Электронный справочник «WEB спутник» [Электронный ресурс] – 2002. – Режим доступа: <http://web-sputnik.info/>- Дата доступа: 14.05.2018.
6. Дакетт Джон HTML и CSS. Разработка и дизайн веб-сайтов (+ CD-ROM); Эксмо - Москва, 2013. - 480 с.
7. Дронов Владимир HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов; БХВ-Петербург - Москва, 2011. - 416 с.
8. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Управление внедрением информационных систем. – Москва, 2008. - 240 с
9. Дронов Владимир PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов; БХВ-Петербург - М., 2010. - 448 с.
10. Ищенко В. А. 100% самоучитель. Web-дизайн. Создавай свои сайты; Технолоджи-3000, Триумф - М., 2009. - 144 с.
11. Контентная стратегия управления сайтом; Питер - М., 2013. - 224 с.
12. Мацеевский Николай Разгони свой сайт. Методы клиентской оптимизации веб-страниц; Интернет-университет информационных технологий, Бином. Лаборатория знаний - М., 2009. - 264 с.
13. Петин Виктор API Яндекс, Google и других популярных веб-сервисов. Готовые решения для вашего сайта; БХВ-Петербург - М., 2012. - 480 с.

14. Уильямс Б., Дэмстра Д., Стэрн Х. WordPress для профессионалов. Разработка и дизайн сайтов; Питер - М., 2014. - 464 с.
15. Цыганова, И.А. Совершенствование маркетинговой информации системы предприятия в условиях конкурентного рынка: авторефер. дис./ И.А. Цыганова, 2007. – 17с.
16. Ароян, Н.М. Повышение эффективности систем управления продажами на основе анализа маркетинговых технологий/ Н. М. Ароян. – Ростов-на-Дону, 2015. – 27 с.
17. Орлова, М. В. Маркетинговые технологии управления интеллектуальным сервисом/М.В. Орлова. – Москва, 2015. – 30 с.
18. Шмитт Кристофер , Блессинг Кимберли , Черни Роб , Эванс Мерил , Ловер Кевин , Траммель Марк Применение Web-стандартов CSS и Ajax для больших сайтов; Корона-Век, МК-Пресс - М., 2011. - 244 с.
19. Черкашин П., Стратегия управления взаимоотношениями с клиентами. – Бином, 2007 г. – 376 с.