

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

Балаковский инженерно-технологический институт –

филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

ФАКУЛЬТЕТ атомной энергетики и технологий

КАФЕДРА «Информационные системы и технологии»

На правах рукописи

УДК 681.3

РЕПИН ДАНИИЛ БОРИСОВИЧ

РАЗРАБОТКА ИНТЕРНЕТ-РЕСУРСА «СРЕДСТВА ИНДИВИДУАЛЬНОЙ ЗАЩИТЫ»

Выпускная квалификационная работа бакалавра

Направление подготовки 09.03.02

(код, наименование)

«Информационные системы и технологии»

Выпускная квалификационная
работа защищена

« » 2021 г.

Оценка

Секретарь ГЭК /Н.М. Виштак/

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Балаковский инженерно-технологический институт –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

Факультет АЭТ Кафедра Информационные системы
и технологии

Направление подготовки 09.03.02
«Информационные системы и
технологии» Группа ИФСТ-41

УТВЕРЖДАЮ

Зав. кафедрой ИСТ

(подпись)

(фамилия, имя, отчество)

« » 2021г.

ЗАДАНИЕ НА БАКАЛАВРСКУЮ РАБОТУ

(выпускную квалификационную работу)

1. Фамилия, имя, отчество студента Репин Даниил Борисович

2. Тема работы Разработка интернет-ресурса «Средства индивидуальной
защиты»

3. Срок сдачи студентом готовой работы 14.06.2021 г.

4. Место выполнения работы БИТИ НИЯУ МИФИ

5. Руководитель работы (проекта) д.п.н., к.т.н., декан ФПКПП Виштак
Ольга Васильевна

(фамилия, имя, отчество, должность, место работы)

6. Консультант работы Михеев И.В., старший преподаватель кафедры
«Информационные системы и технологии»

(фамилия, имя, отчество, должность, место работы)

7. Цель работы: Разработка интернет-ресурса «Средства индивидуальной
защиты»

8. Задание:

а) литература и обзор работ, связанных с работой а

1. Никсон Р., Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. — СПб.: Питер - 2016. - 496 с.

2. Зандстра М., PHP. Объекты, шаблоны и методики программирования. — М.: «Вильямс», 2016. - 560 с.

3. Дронов В., PHP, MySQL. Разработка интерактивных Web-сайтов. БХВ — Петербург - М., 2016. - 480 с.

б) расчетно-конструкторская, теоретическая, технологическая часть ____

1. Анализ задач, функций и требований к интернет-ресурсу

2. Обзор существующих методов и программных разработок в создании интернет-ресурса

3. Разработка технического задания к интернет-ресурсу

в) экспериментальная часть

1. Разработка объектно-ориентированной модели интернет-ресурса

2. Проектирование навигационной структуры интернет-ресурса

3. Обзор и выбор инструментальных средств для реализации интернет-ресурса

4. Описание интерфейса интернет-ресурса

5. Тестирование интернет-ресурса

9. Отчетный материал работы (проекта):

а) пояснительная записка;

б) графический материал (с указанием обязательных чертежей).

10. Консультанты по работе (с указанием относящихся к ним разделов работы (проекта))

Раздел	Консультант	Подпись, дата	
		Задание выдал	Задание принял
Проектирование информационных систем	Михеев И.В	26.04.2021г.	14.06.2021г.

Календарный план работы над выпускной квалификационной работой

(составляется руководителем с участием студента в течение первой недели с начала преддипломной практики)

№ п\п	Наименование этапов работы	Сроки выполнения этапов	Степень готовности ВКР в % к объему работы	Отметка о выполнении
1	Введение Анализ требований для разработки Интернет-ресурса «Средства индивидуальной защиты» Анализ задач, функций и требований к Интернет-ресурсу. Обзор существующих методов и программных разработок в создании Интернет-ресурса. Техническое задание на разработку Интернет-ресурса.	26.04.21г.- 10.05.21г.		
2	Проектирование программного Интернет-ресурса «Средства индивидуальной защиты». Проектирование Интернет-ресурса с использованием объектного моделирования. Проектирование навигационной структуры Интернет-ресурса. Обзор и выбор инструментальных средств для реализации Интернет-ресурса.	11.05.21г.- 23.05.21г.		
3	Реализация Интернет-ресурса «Средства индивидуальной защиты» Описание интерфейса Интернет-ресурса Описание программных модулей Интернет-ресурса. Тестирование Интернет-ресурса. Заключение.	24.05.21г.- 13.06.21г.		
4	Оформление презентации. Сдача ВКР.	14.06.21г.		

Дата выдачи задания 26.04.2020

Руководитель выпускной
квалификационной
работы

(подпись)

(фамилия имя, отчество)

Задание принял к исполнению _____
(подпись)

(фамилия, имя, отчество)

«26» апреля 2021 г.

СОДЕРЖАНИЕ

Введение	8
1 Анализ требований для разработки интернет-ресурса «средства индивидуальной защиты»	7
1.1 Анализ задач, функций и требований к интернет-ресурсам	7
1.2 Обзор существующих методов и программных разработок в создании интернет-ресурса	10
1.3 Техническое задание на разработку Интернет-ресурса	15
2 Проектирование программного интернет-ресурса «средства индивидуальной защиты»	20
2.1 Проектирование Интернет-ресурса с использованием объектного моделирования	20
2.2 Проектирование навигационной структуры Интернет-ресурса	28
2.3 Обзор и выбор инструментальных средств для реализации Интернет-ресурса	31
3 Реализация Интернет-ресурса «средства индивидуальной защиты»	38
3.1 Описание интерфейса Интернет-ресурса	38
3.2 Разработка программных модулей Интернет-ресурса	44
3.3 Тестирование Интернет-ресурса	46
Заключение	51
Список использованных источников	52

					ИФСТ.466452.011 ПЗ			
Изм.	Лист	№ документа	Подпись	Дата				
Разраб.	Репин				Разработка Интернет-ресурса «Средства индивидуальной защиты» Пояснительная записка	Лит.	Лист	Листов
Рук. ВКР	Виштак						5	57
Консульт	Михеев					БИТИ НИЯУ МИФИ ИФСТ – 41		
Н.контр.	Виштак							
Утв.	Очкур							

ВВЕДЕНИЕ

Информационные технологии развиваются очень быстро и стремительно. На сегодняшний день существует множество сайтов с самым различным предназначением. Абсолютно каждой компании или организации необходим свой сайт со своим функционалом под определенные потребности. Поэтому умение создания таких ресурсов сейчас очень востребовано.

Целью выпускной квалификационной работы является разработка интернет-ресурса «средства индивидуальной защиты», который позволит повысить эффективность работы компании путем автоматизации взаимодействия с клиентом и автоматизации продаж.

В рамках выпускной квалификационной работы необходимо выполнить следующие задачи:

- провести анализ функций и требований к интернет-ресурсу;
- провести обзор существующих методов создания интернет-ресурсов;
- спроектировать навигационную структуру интернет-ресурса;
- произвести обзор и выбор инструментальных средств;
- разработать интернет-ресурс;
- описать и протестировать интерфейс.

Теоретическая часть выпускной квалификационной работы заключается во всестороннем анализе концептуальных основ и средств разработки информационных ресурсов.

Практическая часть заключается в разработке интернет-ресурса «средства индивидуальной защиты».

.

1 АНАЛИЗ ТРЕБОВАНИЙ ДЛЯ РАЗРАБОТКИ ИНТЕРНЕТ-РЕСУРСА «СРЕДСТВА ИНДИВИДУАЛЬНОЙ ЗАЩИТЫ»

1.1 Анализ задач, функций и требований к интернет-ресурсам

Сегодня существует огромное множество различных сайтов. Все эти сайты различаются по функционалу, задачам, видам и дизайну.

Выделяют следующие виды сайтов:

— Статические. Такие сайты собраны из неизменяемых страниц, то есть сайт имеет только один внешний вид и одно и то же наполнение для каждого пользователя. При запросе данного сайта сервер выдает готовый HTML-документ, который представлен в изначально созданном виде. Однако, присутствие в коде CSS и JavaScript дает возможность быстрой загрузки. Зачастую такие сайты обуславливаются малым количеством страниц и их наполнением [28]. Структура работы статического сайта изображена на рисунке 1.1.

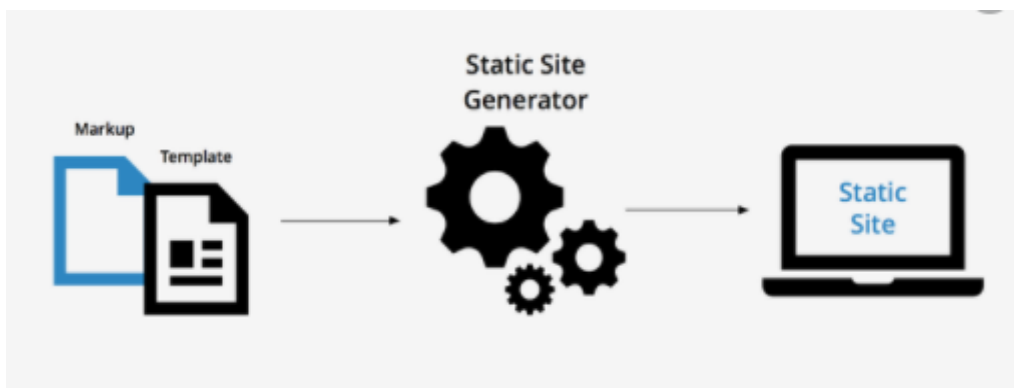


Рисунок 1.1 - Структура работы статического сайта

— Динамические. Сайты этой категории имеют изменяемые страницы, которые ориентируются под каждого пользователя. Эти страницы собираются по новой при каждом запросе. Исходный код генерируется в момент обработки HTML файла интерпретатором Java, PHP или Ruby. После выполнения всех нужных действий интерпретатор отдает сгенерированный код веб-серверу, где уже, в свою очередь, сервер отправляет код документа пользователю [12]. Структура работы динамического сайта представлена на рисунке 1.2.

Изм.	Лист	№ документа	Подпись	Дата

ИФСТ. 466452.011 ПЗ

Лист

7

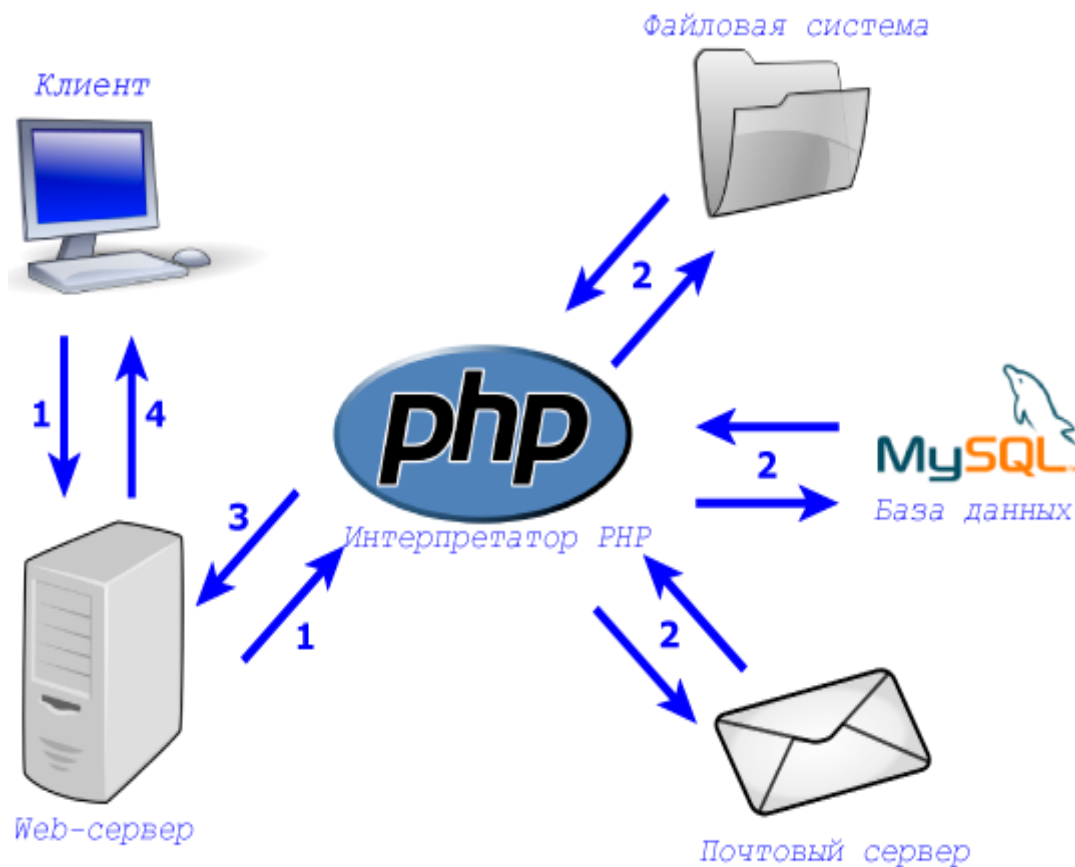


Рисунок 1.2 - Структура работы динамического сайта

Система интернет-магазина ориентирована непосредственно на покупателя, администратора сайта и владельца. Такая система, в первую очередь, должна обеспечивать удобный и интуитивный функционал пользователям.

В функциональные возможности интернет-ресурса входит:

- наличие регистрационной формы;
- наличие поискового конфигуратора;
- размещение контактной информации;
- возможность комментирования и оценки товаров;
- строка поиска с возможностью выбора атрибутов;
- наличие корзины для выбранных пользователем товаров.

Здесь представлен минимально необходимый набор функций сайта, который, по желанию, может быть расширен на неограниченное количество пунктов.

Администраторы сайта имеют отдельный набор возможностей для администрирования и модерации сайта:

- Удаление и редактирование комментариев пользователей;
 - Возможность добавления, редактирования и удаления товаров;
- Это необходимый набор функций администратора, который также может быть расширен.

В функциональные возможности пользователей входит:

- регистрация учетной записи в системе;
- просмотр ленты товаров;
- возможность внесения товара в корзину;
- поиск товара с возможностью указания атрибутов;
- удаление товаров из корзины;
- возможность поставить оценку товару;
- оформление заказа.

Это минимальный набор характеристик сайта. Естественно, что этот набор может быть в дальнейшем расширен.

Интернет магазин регулируется следующей документацией:

1. Гражданский кодекс РФ: нормы, регулирующие обязательственное и авторское право.
2. Федеральный закон «О защите прав потребителей» от 07.02.1992 № 2300-1.
3. Федеральный закон о защите прав персональных данных № 152-ФЗ от 27.07.2006. Закон направлен на предотвращение противоправного и несанкционированного раскрытия и использования персональных данных.
4. ФЗ «О защите детей от информации, причиняющей вред их здоровью и развитию» от 29.12.2010 № 436-ФЗ.
5. Письмо Роспотребнадзора «О контроле за соблюдением Правил продажи товаров дистанционным способом» № 0100/10281-07-32 от 12.10.2007.
6. Письмо Роспотребнадзора «О пресечении правонарушений при дистанционном способе продажи товара» № 0100/2569-05-32 от 08.04.2005.
7. Письмо Роспотребнадзора «О пресечении правонарушений при дистанционном способе продажи товара» № 0100/2569-05-32 от 08.04.2005.

					ИФСТ. 466452.011 ПЗ	Лист 9
Изм.	Лист	№ документа	Подпись	Дата		

8. Правила продажи товаров дистанционным способом, утвержденные Постановлением Правительства РФ 27.09.2007 № 612.

9. ГОСТ Р 57489-2017. «Руководство по добросовестной практике продажи товаров дистанционным способом с использованием сети Интернет». Утвержден и введен в действие 7 июня 2017 года. Этот ГОСТ регулирует взаимоотношения между интернет магазинами и розничными покупателями. Настоящий стандарт распространяется на деятельность лиц, которые занимаются предпринимательской деятельностью в сфере дистанционных продаж товаров покупателям.

10. ГОСТ 34.320-96. «Концепции и терминология для концептуальной схемы и информационной базы». Утвержден и введен в действие 1 июля 2001 года. Настоящий стандарт определяет основные понятия информационных баз, а также манипулирования информацией и реализации информационного процесса.

11. ГОСТ 34.321-96. «Информационные технологии. Система стандартов по базам данных». Утвержден и введен в действие 1 июля 2001 года. Настоящий стандарт определяет модель управления данными. Эталонная модель определяет общую терминологию и понятия, имеющие отношение к области информационных систем.

12. ГОСТ 34.601-90. «Автоматизированные системы. Стадии создания». Утвержден и введен в действие 1 января 1992 года. Настоящий стандарт распространяется на автоматизированные системы, использующиеся в проектировании, управлении и другое [2].

На основании проведенного анализа выбираем динамический тип сайта.

1.2 Обзор существующих методов и программных разработок в создании интернет-ресурса

На данный момент используются следующие способы и средства создания сайтов:

— разработка на CMS;

					ИФСТ. 466452.011 ПЗ	Лист 10
Изм.	Лист	№ документа	Подпись	Дата		

- разработка на конструкторе сайтов;
- самостоятельная разработка [3].

Проанализируем достоинства и недостатки этих способов, чтобы выбрать способ для разработки интернет-ресурса.

CMS является комплексом программных инструментов для управления веб-контентом. То есть это базовый каркас из инструментов и надстроек, позволяющий организовать веб-сайт или веб-приложение, а также регулировать и поддерживать его работу [18].

Достоинствами этого способа являются:

Бесплатный доступ. В основном CMS бесплатны и в сети существует множество готовых шаблонов сайтов под них. Необходимо лишь взять любой понравившийся шаблон и поменять дизайн по своему усмотрению.

Дружелюбность интерфейса. Все CMS имеют панель управления с относительно дружелюбным интерфейсом, с помощью которого управлять сайтом не составит труда. Любая CMS может позволить создать даже очень сложные решения, такие как интернет-магазины или большие корпоративные сайты.

Множество готовых решений. В сети есть большое количество плагинов, модулей и дополнений для реализации различных задач [20].

Но этот способ имеет недостатки:

Уязвимость. Сайты, которые сделаны на конструкторах порой лучше защищены от взлома, чем те, которые находятся под управлением WordPress и других CMS.

Требования к знаниям. При разработке сайта для добавления нового функционала или создания отдельного нового своего проекта пользователю потребуются базовые знания по верстке и программированию.

Трудности с переносом. Популярные CMS в настоящее время имеют автоматизированные инструменты установки практически на любой хостинг, но если вам нужно переместить сайт или управлять его положением, у вас могут

возникнуть трудности, так как вам придется выполнить всю процедуру установки заново.

Стоимость дополнительного контента. Дополнительные модули и расширения сайта для CMS стоят денег, причем в данном случае уже больше, чем для конструкторов. Практически любой модуль, относящийся к постоянно используемым, требует один из двух вариантов оплаты:

- разовая оплата большой суммы в начале использования;
- оформление платной ежемесячной подписки.

Большой сайт обусловлен высокими затратами. Разработка сложного крупного проекта на CMS обойдется не дешевле, а в некоторых случаях и дороже, чем разработка на чистых языках или с использованием фреймворков [30].

Разработка в конструкторе является программным решением, чаще всего онлайн, которое позволяет строить сайт на модульной основе, которая обуславливается созданием структуры с помощью готовых элементов сайта, которые предоставляет конструктор. Такой подход позволяет создать сайт без каких-либо знаний в области веб-разработки.

Достоинствами этого способа являются:

Низкая цена. Почти все конструкторы изначально бесплатны, а стоимость плагинов очень низкая.

Простота использования. Для того чтобы создать свой собственный сайт, достаточно уметь пользоваться компьютерным браузером и иметь чувство вкуса.

Все процедуры выполняются конструктором. Большие программные скрипты, плагины, хостинг и другие скрытые процессы реализуются через понятную панель управления с помощью нескольких нажатий на кнопки [16].

Но этот способ имеет недостатки:

Скрытые издержки. Первичная дешевизна часто скрывает дополнительные, порой значительные, расходы: хостинг, домен второго уровня, электронная почта с доменным именем и др.

Домены. При использовании бесплатных аккаунтов в конструкторах, клиент получает размещение своего сайта только на домене не ниже третьего уровня. Такие домены не вызывают большого доверия у интернет-сообщества, а домены второго уровня стоят дороже, чем если их приобретать у регистраторов доменов напрямую.

Тяжеловесность сайта. Сайт, сделанный на конструкторе всегда будет загружаться дольше аналогичного сайта, сделанного на CMS или разработанного самостоятельно. Объясняется это тем, что конструктор содержит в себе огромное количество программного кода, который не относится к Вашему сайту, но необходим для построения его итогового внешнего вида;

Нагрузка сайта. Сайт, который сделан на конструкторе всегда будет загружаться дольше, чем сайты, сделанные на CMS или собранные самостоятельно. В структуре собранного сайта на конструкторе содержится много лишнего программного кода, который изначально прилагается к устанавливаемым плагинам.

Отсутствие SEO. SEO является комплексом мероприятий по работе с кодом, индексации в поисковых системах, построению структуры сайта. Все перечисленное требует доступ к программному коду и разметке, что невозможно в конструкторе [19].

Самостоятельная разработка является свободным, но и самым трудоемким процессом. Требуется серьезных знаний не только самих языков программирования, но и понимания архитектуры и бизнес-процессов. Заказчик получает персонализированный уникальный продукт, в дальнейшем решающий его задачи и цели, а также избавляет его от траты времени на ненужные процессы. Самостоятельная разработка позволяет создавать проекты любой сложности и в соответствии с любыми пожеланиями клиента [7].

Достоинствами этого способа являются:

Свобода выбора. Позволяет заказать все необходимое для эффективного решения поставленных бизнес-задач. Весь функционал в итоге будет написан именно под конкретные нужды.

Собственный дизайн. Только чистый код позволит вам создать продукт, который будет соответствовать вашему индивидуальному видению [17].

Но этот способ имеет недостатки:

Наличие специальных знаний. Такая разработка требует наличия не только базовых знаний, но и специальных, без которых не обойтись при использовании языков программирования, построения архитектуры веб-приложений, алгоритмов и структур данных, бизнес-процессов и другое.

Временные затраты. Самописные сайты требуют большого количества времени при написании. Разработчику необходимо внимательно подходить к проектированию сайта. В тех проектах, где время - это главный ресурс, играющий большую роль, не имеет смысла обращаться к самописным сайтам [13].

Таблица 1 - Итоговая таблица анализа

Критерии	Разработка на CMS	Разработка на конструктор сайтов	Самостоятельная разработка
Значительные временные затраты	-	-	+
Тяжеловесность сайта	+	+	-
Свобода выбора	-	-	+
Собственный дизайн	-	-	+
Уязвимость	+	+	-
Наличие специальных знаний	+	-	+
Трудности с переносом	+	-	+
Бесплатный доступ	+	-	+

На основании проведенного анализа основных критериев предъявляемых способов создания сайтов (таблица 1) делаем вывод, что предпочтения следует отдать самописному способу создания сайтов, так как он представляет свободу выбора, возможность собственного дизайна, а также выполняется полностью самим программистом, что позволяет сделать этот способ бесплатным.

1.3 Техническое задание на разработку Интернет-ресурса

1. Общие сведения

1.1 Наименование сайта

Rocket shoes

1.2 Назначение и область применения

Сайт предназначен для обычных пользователей ПК.

Целью сайта является получение прибыли, а задачей - продажа товаров клиенту.

2 Требования к сайту

2.1 Требования к функциональным характеристикам

Сайт должен обеспечивать возможность выполнения перечисленных ниже функций:

- просмотр и покупка товаров;
- занесение товаров в корзину;
- Наличие поискового configurатора;
- Размещение контактной информации;
- написание отзывов и проставление оценок.

2.2 Требования к надежности

2.2.1 Требования к обеспечению надежного функционирования сайта

Сайт должен предусматривать базовую защиту от основных видов атак: межсайтового скриптинга (XSS), SQL-инъекций, CSRF-уязвимостей.

Рекомендуется хранить одну копию программного обеспечения на внешнем носителе. Также рекомендуется регулярно копировать файл базы данных на внешний носитель, если такую услугу не предоставляет хостинговая компания.

2.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии

					ИФСТ. 466452.011 ПЗ	Лист 15
Изм.	Лист	№ документа	Подпись	Дата		

соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

2.2.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой.

Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

3 Условия эксплуатации

3.1 Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

3.2 Требования к квалификации и численности персонала

Количество персонала минимально - Администраторы, которые должны иметь высшее профильное образование, а также конечный пользователь программы - оператор.

3.3 Требования к составу и параметрам технических средств

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), выполняющий роль сервера, включающий в себя:

- процессор Pentium-2.0Hz, не менее;
- оперативную память объемом, 256 Мбайт, не менее;
- свободного пространства на жестком диске, 1 Гигабайта, не менее;
- Операционная система Windows XP не ниже SP3.

3.4 Требования к информационной и программной совместимости

					ИФСТ. 466452.011 ПЗ	Лист 16
Изм.	Лист	№ документа	Подпись	Дата		

3.4.1 Требования к информационным структурам и методам решения

База данных работает под управлением Microsoft SQL Server, посредством многопоточного доступа к базе данных.

3.4.2 Требования к исходным кодам и языкам программирования

Дополнительные требования не предъявляются.

3.4.3 Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 2000 Server или Windows 2003 и Microsoft SQL Server 2000.

3.4.4 Требования к защите информации и программ

Проектируемый интернет-ресурс и его отдельные компоненты должны соответствовать требованиям федерального законодательства, нормативно-правовых актов регуляторов в области защиты информации и персональных данных.

3.5 Специальные требования

Специальные требования к данной программе не предъявляются

4 Требования к программной документации

4.1 Предварительный состав программной документации

Состав программной документации должен включать в себя:

- техническое задание;
- программу и методики испытаний;
- руководство оператора.

5 Технико-экономические показатели

5.1 Экономические преимущества разработки

Ориентировочная экономическая эффективность не рассчитывается. Аналогия не проводится ввиду уникальности предъявляемых требований к разработке.

6 Стадии и этапы разработки

6.1 Стадии разработки

Разработка должна быть проведена в три стадии:

					ИФСТ. 466452.011 ПЗ	Лист 17
Изм.	Лист	№ документа	Подпись	Дата		

- разработка технического задания;
- рабочее проектирование;
- внедрение.

6.2 Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

- разработка сайта;
- разработка документации;
- испытания.

6.3 Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к сайту;
- определение стадий, этапов и сроков разработки сайта и документации на неё;
- согласование и утверждение технического задания.

На этапе испытаний сайта должны быть выполнены перечисленные ниже виды работ:

- разработка, согласование и утверждение методики испытаний;
- проведение приемо-сдаточных испытаний;
- корректировка сайта и документации по результатам испытаний.

На этапе подготовки и передачи приложения должна быть выполнена работа по подготовке и передаче сайта и программной документации в эксплуатацию на объектах Заказчика.

7 Порядок контроля и приемки

7.1 Виды испытаний

					ИФСТ. 466452.011 ПЗ	Лист 18
Изм.	Лист	№ документа	Подпись	Дата		

Приемо-сдаточные испытания должны проводиться на объекте Заказчика в оговоренные сроки.

Приемо-сдаточные испытания сайта должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком Программы и методик испытаний.

Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний

7.2 Общие требования к приемке работы

На основании Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывает Акт приемки-сдачи сайта в эксплуатацию.

В ходе проведения анализа задач, функций и требований к интернет-ресурсу, обзора существующих методов и программных разработок в создании интернет-ресурса было разработано техническое задание, регламентирующее проектные решения и документацию по системе и ее частям.

					ИФСТ. 466452.011 ПЗ	Лист
						19
Изм.	Лист	№ документа	Подпись	Дата		

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ИНТЕРНЕТ-РЕСУРСА «СРЕДСТВА ИНДИВИДУАЛЬНОЙ ЗАЩИТЫ»

2.1 Проектирование Интернет-ресурса с использованием объектного моделирования

2.1.1 Диаграмма вариантов использования

Диаграмма вариантов использования является диаграммой, которая позволяет описать систему на концептуальном уровне.

Такая диаграмма состоит из актеров, вариантов использования и отношений между ними. Вариант использования - это спецификация функций, которая актеру предоставляет система. Актер - это любой объект, который взаимодействует с системой извне [26].

Выделим актеров, которые будут пользоваться информационной системой и опишем их полномочия, **таблица 2.**

Таблица 2 – Описание действующих лиц

№ п/п	Имя	Описание
1	Администратор сайта	Лицо, отвечающее за контроль товаров, а также контролирующее работу сайта.
2	Пользователь	Клиент, который находится в системе за тем, чтобы произвести заказ понравившегося товара.

Выделим множество вариантов использования системы. **Они представлены в таблице 3.**

Таблица 3 – Описание вариантов использования

№ п/п	Имя	Описание
1	Аутентификация	Вход в систему для распределения возможностей пользователей.
2	Контроль аккаунтов	Контроль и удаление пользователей администратором
3	Добавление в корзину	Выбор конкретного товара из списка товаров
4	Заказ товара	Ввод данных для оформления заказа

Продолжение таблицы 3

5	Просмотр товара	Возможность просмотра всего списка товаров
6	Удаление товара	Удаление конкретного товара из списка товаров администратором
7	Сортировка товара	Определение категории товара и внесение товара в подходящие списки администратором
8	Добавление товара	Добавление поступающей продукции в базу данных фирмы.
9	Отслеживание статуса	Вывод формы с данными о местоположении товара

На рисунке 2.1 представлена диаграмма вариантов использования системы.

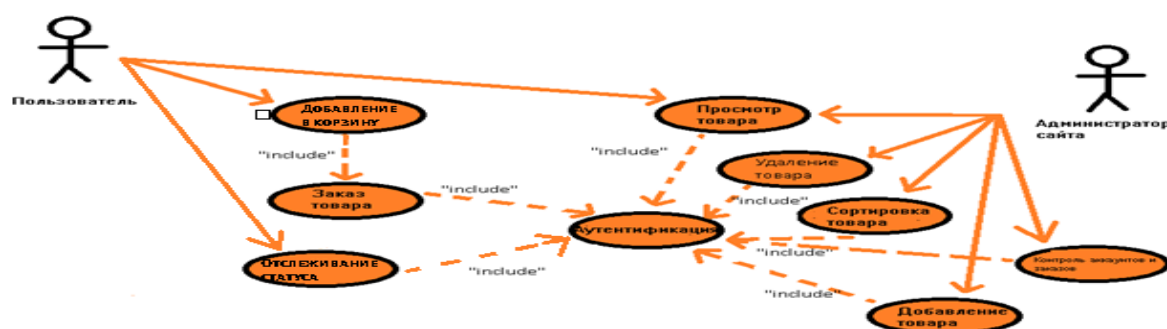


Рисунок 2.1 – Диаграмма вариантов использования

2.1.2 Диаграмма последовательности

Диаграмма последовательности дает возможность моделирования взаимодействия пользователя и системы в целом.

Такая диаграмма изображена на рисунке 2.2.

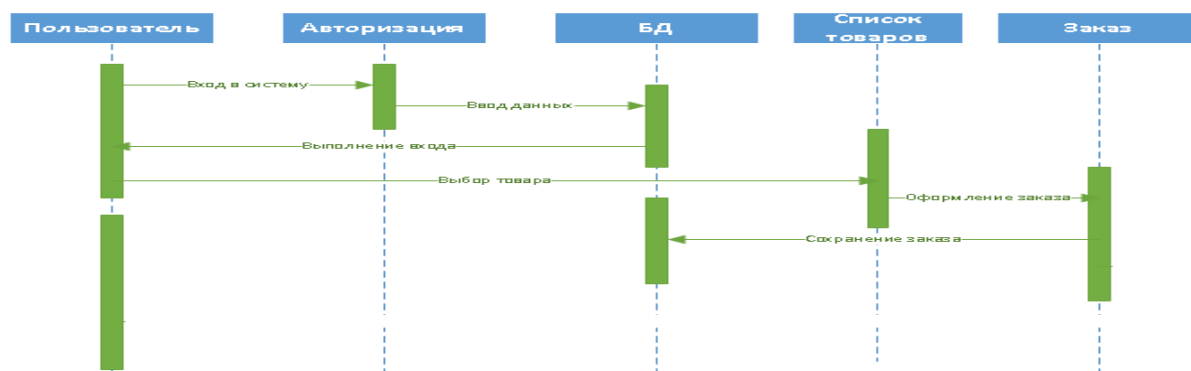


Рисунок 2.2 - Диаграмма последовательности

Данная диаграмма содержит следующие элементы:

— Пользователь;

- Авторизация;
- База данных;
- Список товаров;
- Заказ.

На этой диаграмме продемонстрировано как пользователь работает с сайтом. Сначала он проходит авторизацию, где данные проверяются в базе данных, далее система подтверждает правильность введенных данных, после чего дает доступ к выбору товара. Когда пользователь выбрал товар из списка товаров, происходит оформление заказа, далее его сохранение в нашей базе данных.

2.1.3 Диаграмма классов

Диаграмма классов отображает типы данных и отношения между ними. Также на диаграмме присутствуют классы. Класс - это план объекта, содержащий атрибуты, операции, отношения и семантику. Обозначается он как прямоугольник [21].

Классы сущностей, отвечающие за хранение данных и объектов, представлены в таблице 4.

Таблица 4 – Описание классов

№	Имя класса	Краткое описание
1	Выбор заказа	Информация о товаре
2	Детали заказа	Информация о деталях заказа
3	Заказ	Информация о заказе
4	Мониторинг заказа	Информация об отправленном заказе
5	Администратор	Информация об Администраторе

В данной диаграмме имеется пять классов, каждый из которых имеют свою определенную информацию.

Каждый класс имеет определенные атрибуты. Атрибуты, описывающие классы, представлены в таблице 5.

Таблица 5 – Описание атрибутов классов

Имя класса	Заказ		
Атрибут	Видимость	Тип атрибута	Множественность
ID заказа	#	int	1...* к Мониторинг заказов
ID заказчика	#	int	1 к Администратор
Дата заказа	#	date	
Дата доставки	#	date	
Имя класса	Выбор заказа		
Атрибут	Видимость	Тип атрибута	Множественность
ID товара	Public	int	0...1 к Детали заказа
Имя класса	Детали заказа		
Атрибут	Видимость	Тип атрибута	Множественность
	Public	string	0...1 к Администратор 0...1 к Выбор заказа
Имя класса	Администратор		
Атрибут	Видимость	Тип атрибута	Множественность
ID администратора	-	int	0...* к Заказ
Имя		varchar	0...1 к Детали заказа
Фамилия	-	varchar	
Отчество	-	date	
Дата рождения	-		
Имя класса	Мониторинг заказа		
Атрибут	Видимость	Тип атрибута	Множественность
ID заказа	#	int	1 к Заказ
Описание заказа	#	string	
Местоположение заказа	#	string	
Пункт прибытия заказа	#	string	

Область видимости делится на три вида:

- «#». Этот символ обозначает атрибут с защищенным типом видимости.
- «-». Этот символ обозначает атрибут с закрытой областью видимости.
- «+». Этот символ обозначает атрибут с общедоступной областью видимости.

Таким образом, получаем статическую диаграмму, описывающую структуру системы, демонстрирующую ее классы, атрибуты и зависимости между классами. На рисунке 2.3 изображена диаграмма классов.

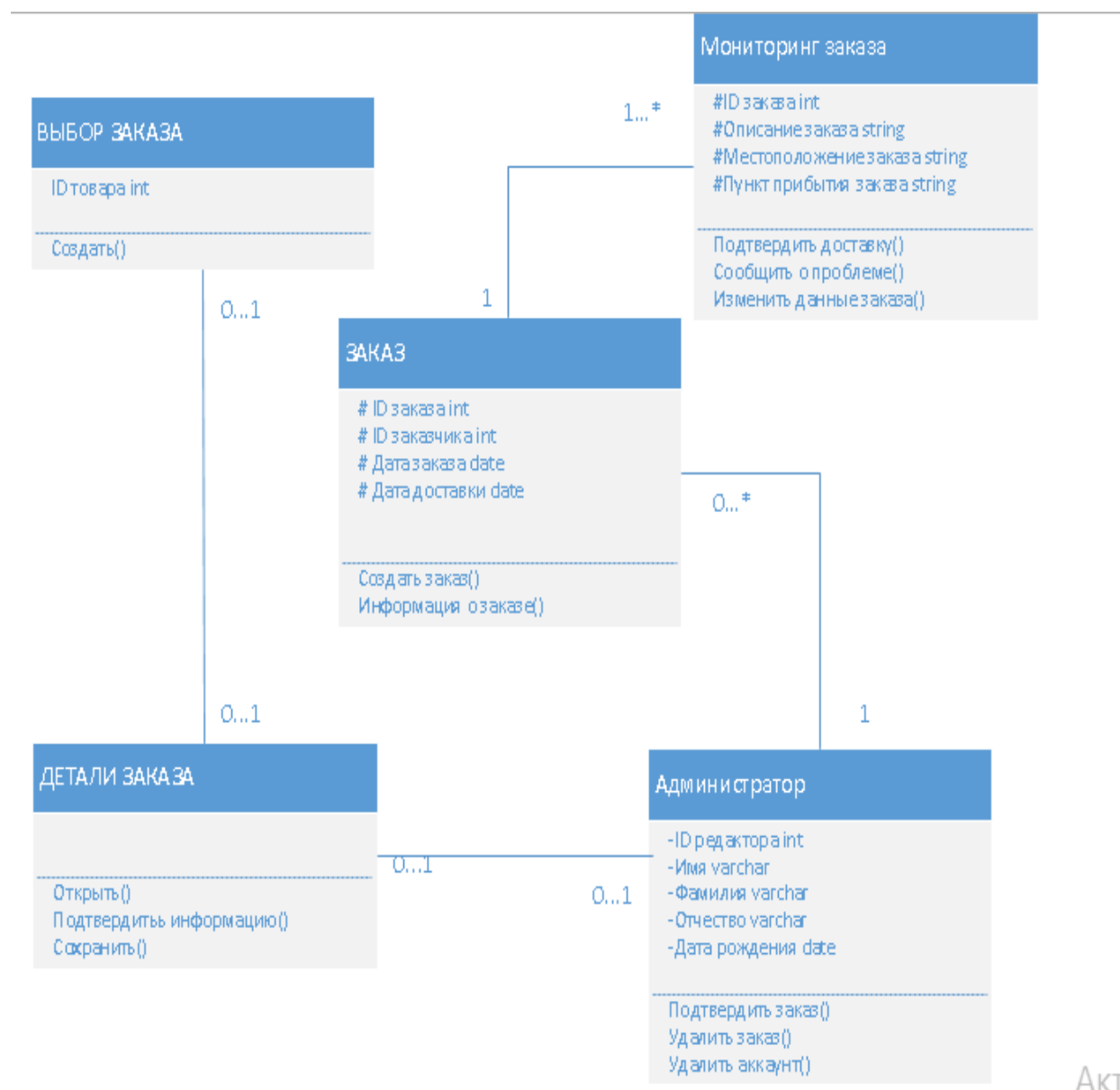


Рисунок 2.3 – Диаграмма классов

2.1.4 Диаграмма состояний

Диаграмма состояний - средство описания поведения систем. Такие диаграммы определяют все возможные состояния объекта и сам процесс смены состояний этого объекта.

В UML есть пять типов данных, с которыми можно связать состояние:

- входное действие;
- выходное действие;
- деятельность;
- событие;
- история состояния.

Состоянием, в этом языке, является период жизни объекта, пока он удовлетворяет условию. На схеме изображается состояние как прямоугольник с закругленными углами.

Для указания действий используются метки «вход/» и «выход/». Все переходы между состояниями должны быть четко определены, на диаграмме не должно быть произвольных переходов [22].

Начальное состояние означает запуск системы. Далее происходит ожидание входа в систему. После того как вход завершен, начинается ввод данных для авторизации. Если проверка данных не выявила ошибок, то пользователь приступает к выбору товара, если данные ошибочны, то выполняется переход на состояние ввода данных. После того, как покупатель подтвердил выбранный товар, то товар отправляется в корзину, где пользователю предлагается выбрать еще товары, либо начинать оформление заказа. На вход в этом состоянии поступает загрузка формы оформления заказа, а на выход поступают введенные данные. Следующим состоянием является проверка введенных данных, если данные ошибочны, то система вернет на состояние оформления заказа. Если же все данные верны, то пользователь выбирает продолжать ли ему работу с системой или же нет. Если клиент хочет продолжить, то система переведет его на состояние выбора товара, если же

пользователь хочет закончить, то начинается завершение работы. Конечное состояние - это выход из системы.

На рисунке 2.4 изображена диаграмма состояний системы при работе пользователя с системой.

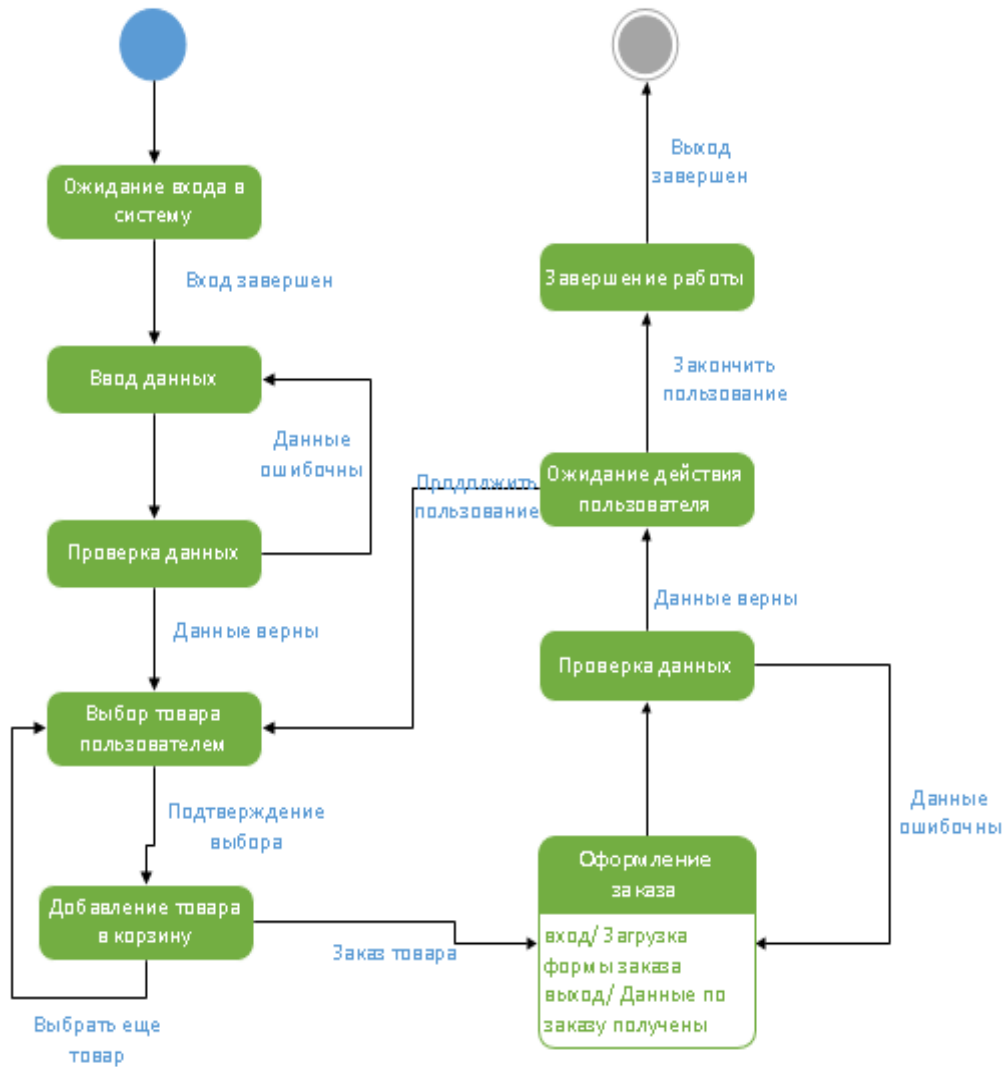


Рисунок 2.4 – Диаграмма состояний

Таким образом, была построена диаграмма состояний.

2.1.5 Диаграмма компонентов

Диаграмма компонентов обуславливается согласованным переходом от логического к физическому представлению системы в виде компонентов.

Основные цели при разработке диаграмм компонентов:

- спецификация исполнимого варианта системы;
- спецификация всей структуры системы.

Изм.	Лист	№ документа	Подпись	Дата

В основные элементы таких диаграмм входят:

- компоненты;
- интерфейсы;
- зависимости между ними;
- классы, которые входят в компоненты [23].

Интерфейс доступен только после аутентификации. Он связывает пользователя с модулями. При этом информация из модулей сохраняется в той же базе.

На рисунке 2.5 изображена диаграмма компонентов моделируемой системы.

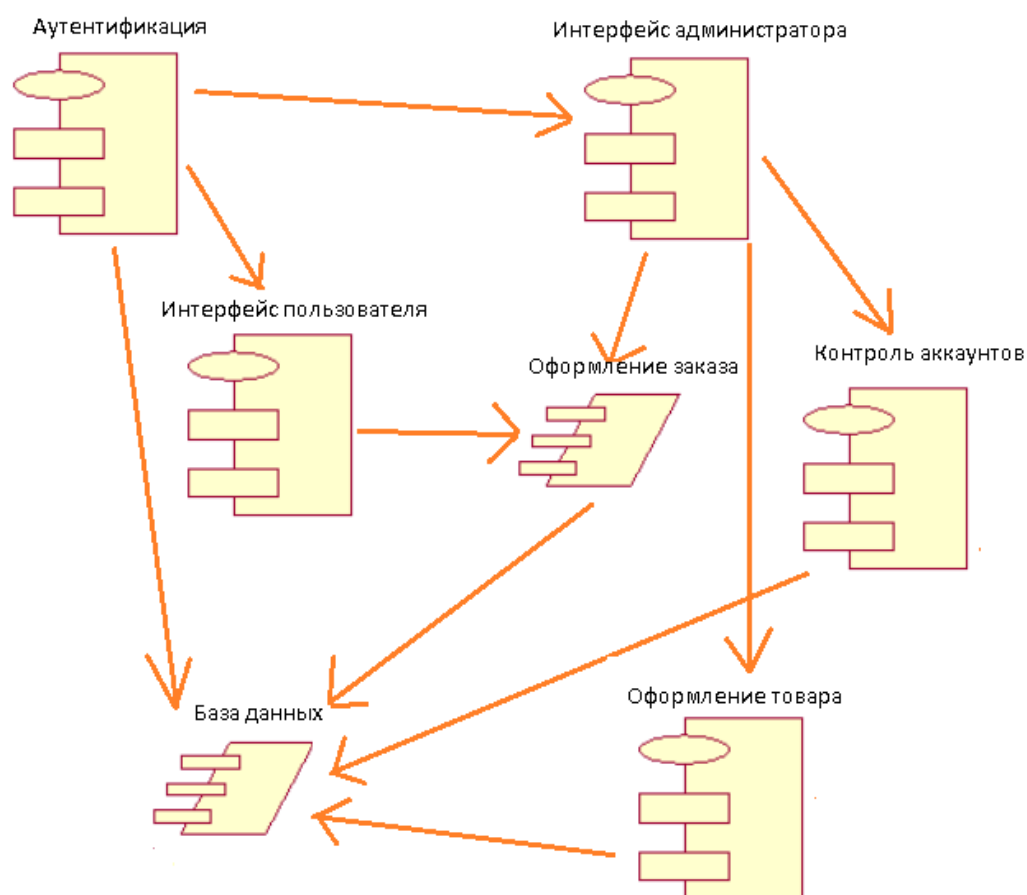


Рисунок 2.5 – Диаграмма компонентов

Компонент является физической частью системы. Они должны обладать согласованным набором интерфейсов.

Интерфейс является набором операций, которые компонент может предоставить другому компоненту для выполнения отдельных функций.

2.1.6 Диаграмма развертывания

Такие диаграммы позволяют графически представить ИТ-системы, а также грамотно распределить все компоненты системы по узлам сети, ведь это влияет на производительность.

Вторым пунктом пользы таких диаграмм является возможность решения множества задач, которые связаны с безопасностью системы.

Обычно требуется создать одну диаграмму развертывания для системы в целом [24].

На рисунке 2.6 представлена диаграмма развертывания.



Рисунок 2.6 – Диаграмма развертывания

Эта диаграмма содержит топологию системы и распределение компонентов всей системы по узлам, а точнее их графическое представление.

Так же на диаграмме присутствуют соединения - маршруты транслирования информации между узлами.

Разработка диаграммы развертывания является последним этапом спецификации модели.

2.2 Проектирование навигационной структуры Интернет-ресурса

Структура web-сайта — это перечень разделов, страниц, подразделов и других функциональных блоков, которые могут использоваться в качестве отдельных элементов, а также их взаимосвязей.

Структурой сайта является правильное построение разделов и страниц. Она показывает логическую связку страниц веб-ресурса.

Различают пять основных вида структур:

— Иерархическая структура, заключающаяся в переходах с главной страницы на разделы второго уровня, а те, в свою очередь, имеют ссылки на разделы третьего уровня. Такая структура представлена на рисунке 2.7.

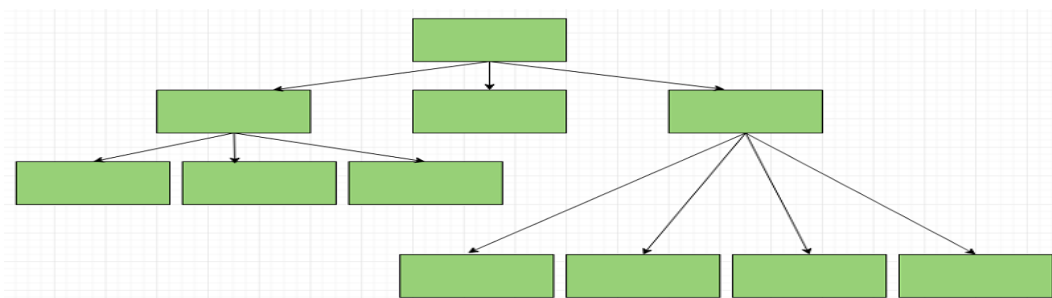


Рисунок 2.7 - Иерархическая структура веб-ресурса

— Решетчатая. Одна страница связана с двумя и более страницами одновременно. Эта структура представлена на рисунке 2.8.

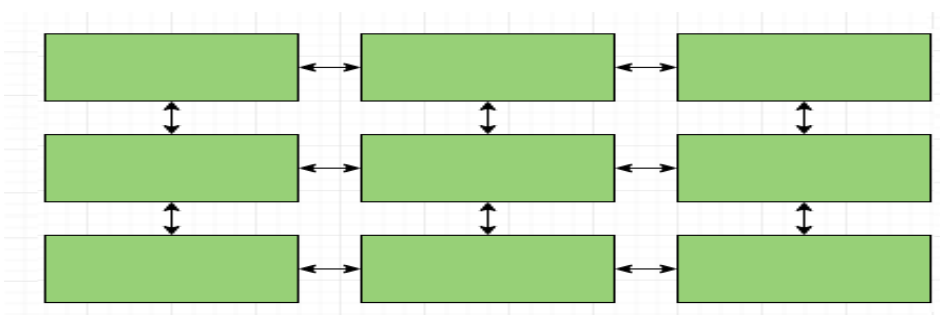


Рисунок 2.8 - Решетчатая структура веб-ресурса

— Линейная. Такая структура является самой простой. Каждая страница имеет переход на следующую и предыдущую. Линейная структура изображена на рисунке 2.9.

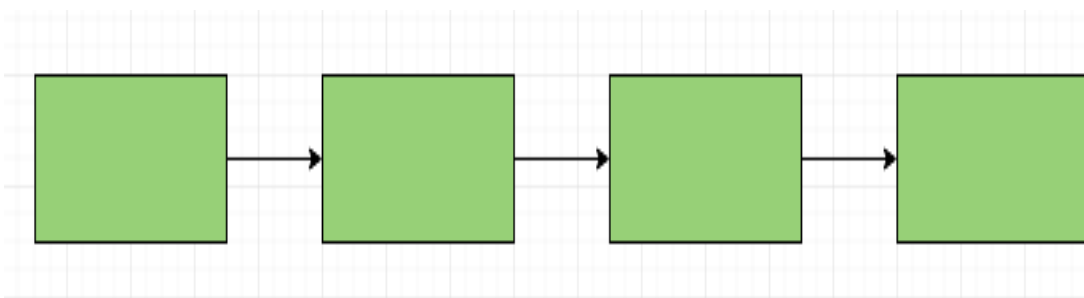


Рисунок 2.9 – Линейная структура веб-сервиса

— Паутинообразная. Она имеет полную связь между страницами. Клиент может перейти с любой страницы на любую другую. Такая структура представлена на рисунке 2.10.

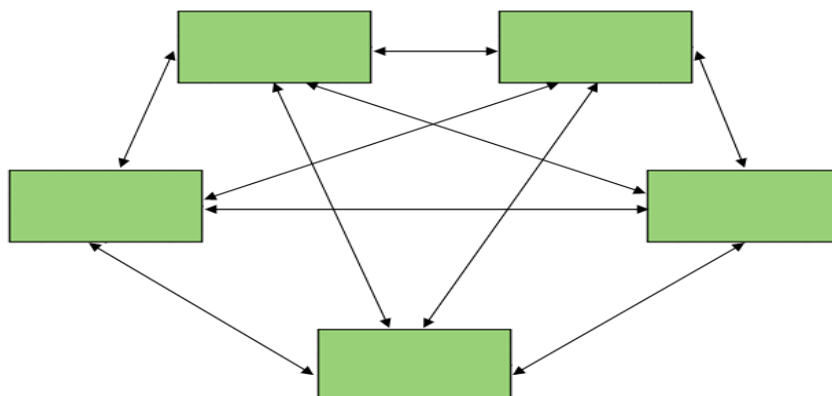


Рисунок 2.10 - Паутинообразная структура веб-сервиса

— Гибридная - является смесью всех выше перечисленных структур.

Для проектирования структуры разрабатываемого интернет-ресурса была выбрана паутинообразная модель. Так как она позволит пользователю быстро переходить с каждой страницы на любую другую, минуя какие-либо промежуточные страницы.

Структура навигации разрабатываемого сайта представлена на рисунке 2.11.



Рисунок 2.11 - Структура навигации сайта

При открытии сайта через браузер, пользователь попадает на главную страницу «Каталог», который содержит в себе списки товаров разных категорий с указанием соответствующих названий и цен.

Страница «Авторизация/Регистрация» содержит два текстовых поля, в которые вводится логин и пароль пользователя. После чего введенные данные сравниваются с уже имеющимися в базе данных данными и система выдает либо доступ, либо отказ.

Страница «Корзина товаров» будет содержать в себе все выбранные пользователем товары, с целью дальнейшего приобретения. Сама корзина используется для того, чтобы запомнить конкретный товар и не искать его по новой. Также на этой странице присутствует возможность удаления товара, если пользователю перехотелось держать данный товар в корзине. Помимо этого, отображается полная стоимость всех внесенных товаров.

«Личный кабинет» это страница, содержащая основную информацию клиента и историю покупок. Также на этой странице присутствует возможность выхода из учетной записи. Для администраторов данного сайта в личном кабинете появляется дополнительное меню, в котором имеются возможности добавления, удаления и редактирования товаров и комментариев.

2.3 Обзор и выбор инструментальных средств для реализации Интернет-ресурса

Для создания интернет-магазина могут быть задействованы такие языки программирования, как Python, PHP, JavaScript, C++ и другие.

Выбор языков и технологий программирования является необходимой частью для разработки интернет-ресурса. Основными критериями являются скорость написания и простота языка.

На рисунке 2.12 представлено сравнение востребованности самых популярных языков программирования.

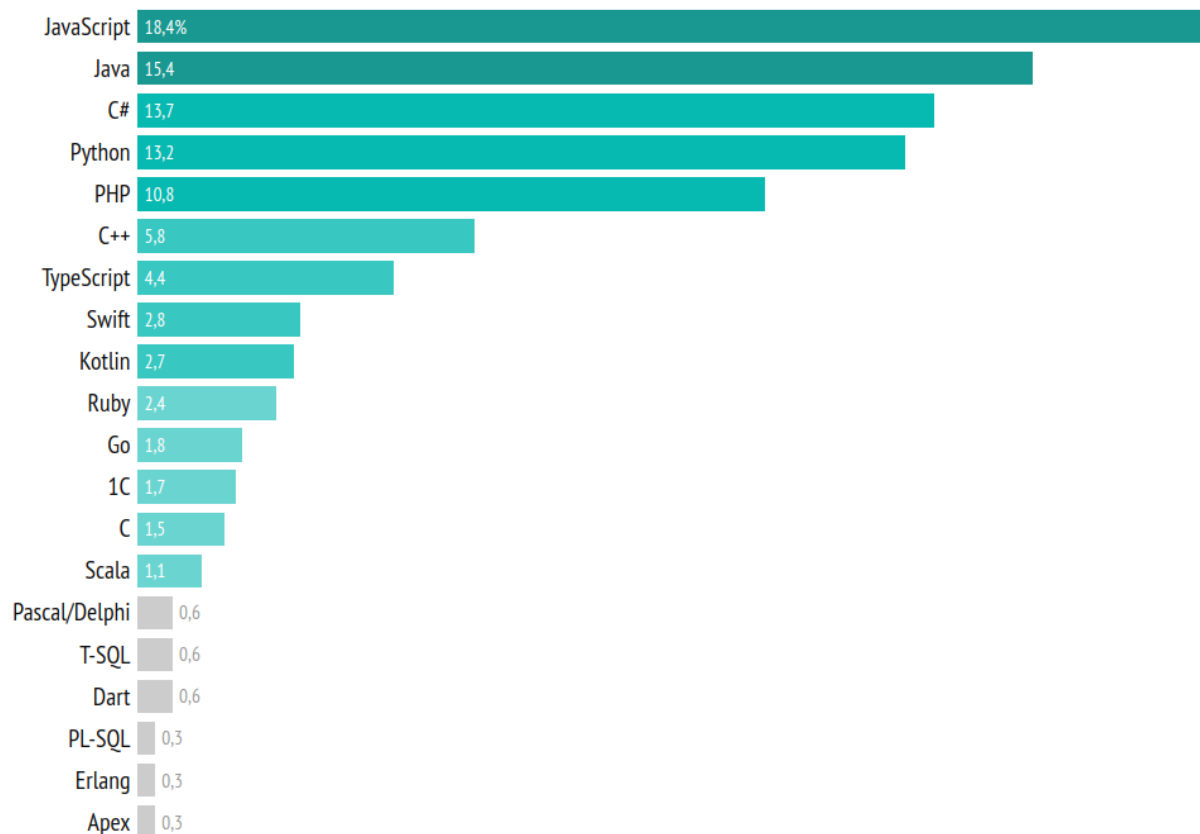


Рисунок 2.12 - Рейтинг востребованности языков программирования

Самыми часто используемыми языками программирования, с помощью которых создают сайты, являются языки: Python, Java, JavaScript, PHP, C#.

Необходимо рассмотреть все достоинства и недостатки каждого из языков, чтобы выбрать более подходящий ресурса язык программирования для создаваемого интернет-ресурса.

Чистый язык. На таком языке создаются большие проекты с повышенными требованиями, где бюджет не является главным аспектом. Большой проект обуславливается большим стеком технологий, который в нем используется. Зачастую, в порталах используются сразу несколько языков программирования.

Язык PHP, зачастую, применяется в основном для простых и средних проектов. В его арсенал входят огромное количество библиотек и расширений языка.

Это язык программирования, который используется для динамической генерации HTML-страниц на стороне WEB-сервера.

PHP является языком программирования для разработки веб-приложений. Вследствие этого, он содержит в себе все функции, которые обязательны при

разработке на веб-сервере. Но в отличие от его конкурентов этот язык лишен избыточности [6].

Команды этого языка программирования включаются в обычные HTML-страницы при помощи специальных тегов. Такие тэги направляют РНР-машину на выполнение необходимых действий на сервере. Как правило, существует возможность произвольного чередования языков HTML и РНР на одной страничке.

Python является современным языком программирования, использующийся для средних и больших проектов, разработка на котором проходит довольно быстро.

Это изначально объектно-ориентированный, интерпретируемый язык программирования. Такой язык содержит небольшое количество ключевых слов, но при этом очень гибок и прост. Это язык имеет встроенные высокоуровневые структуры данных, что делает его уровень более высоким, в сравнении с Pascal, C и C++ [14].

Достоинством этого языка является интерпретатор Python, который реализован на большинстве операционных систем и платформ. Типы данного языка на разных машинах не занимают разное количество памяти, что не является препятствием при написании переносимой программы.

Также немаловажной чертой является расширяемость языка. Она позволяет совершенствовать язык всем желающим.

Из недостатков можно выделить невысокую скорость выполнения программ. Это обуславливается ее интерпретируемостью.

Для разработки интернет-ресурса, также использовался язык разметки гипертекста. Для формирования контента, сначала нужно собрать каркас из языка HTML. Он не является языком программирования, поэтому не может создавать динамические функции. Вместо этого он позволяет систематизировать и форматировать документы.

Ряд элементов, из которых состоит HTML, позволяет добавлять и оборачивать различные части контента, а также задает контенту способ отображения или действия при определенных манипуляциях [10].

Java представляет собой объектно-ориентированный язык общего назначения, независимый от платформы как на двоичном уровне, так и на уровне исходного кода.

Используется для создания приложений в достаточно разных областях:

- веб-сервисы;
- разработка игр;
- мобильное ПО;
- программы для корпораций и другое [3];

Простой синтаксис вместе с удобной в работе средой разработки позволяет языку Java высвободить мощь объектно-ориентированного проектирования приложений. Поэтому у широкого круга программистов есть возможность быстро создавать новые программы и апплеты [1].

Java обеспечивает богатый набор классов объектов, что позволяет ясно абстрагировать многие системные функции, которые используются при работе с сетью, окнами, а также для ввода-вывода. Классы Java предоставляют возможность создания абстракций для широкого спектра системных интерфейсов, которые независимы от применяемой платформы [10].

JavaScript является интерпретируемым языком программирования высокого уровня. В арсенале такого языка имеются анонимные функции, синтаксис фигурных скобок, наличие объектов как списков, динамическая типизация, прототипное программирование и наличие функций как объектов первого класса [5].

Такой язык представляет собой объектно-ориентированный язык программирования, который легко интегрируется к другим языкам.

JavaScript является неотъемлемой частью веб-приложений, так как обеспечивает интерактивные веб-страницы. Основные веб-браузеры имеют специальный механизм JavaScript для его выполнения [25].

JavaScript — это язык программирования, который служит для управления сценариями просмотра веб-страниц. Главная особенность данного языка заключается в том, что при его использовании есть возможность изменять свойства среды отображения при просмотре веб-сайта. Перегрузка веб-страниц при этом происходить не будет. К примеру, с помощью этого языка можно вывести различные уведомления, заменить цвет фона веб-страницы или изменить изображение, которое загружено на веб-страницу [9].

Таблица 6 - Таблица сравнения языков

Язык программирования	Преимущества	Недостатки
Java	<ul style="list-style-type: none"> -безопасность; -кроссплатформенность; -объектно-ориентированность; -многопоточность; -стабильность; -простой синтаксис. 	<ul style="list-style-type: none"> -низкая производительность; -отсутствие нативного дизайна; -многословный и сложный код; -платное коммерческое использование; -требует большого количества памяти.
JavaScript	<ul style="list-style-type: none"> -широкая область применения; -отлично работает с другими языками; -много функциональных настроек; -бесперебойное обновление; -скорость загрузки; -удобство пользовательских интерфейсов; -простота и рациональность применения; 	<ul style="list-style-type: none"> -безопасность; -разными браузерами интерпретируется по разному; -широкое распространение;
Python	<ul style="list-style-type: none"> -синтаксис языка прост для новичка; -логичный и понятный; -кроссплатформенность; -широкое применение; -высокая востребованность; -множество качественных библиотек. 	<ul style="list-style-type: none"> -низкая скорость выполнения программ; -проблемы с отступами при копировании кода; -проблема с русскими символами; -большое потребление памяти.
C#	<ul style="list-style-type: none"> -поддержка Microsoft; -постоянно совершенствуется; -большое сообщество программистов; -высокая востребованность; 	<ul style="list-style-type: none"> -платное коммерческое использование; -легко дизассемблируется; -приоритетная ориентированность на Windows платформу.

Продолжение таблицы 6

PHP	<ul style="list-style-type: none"> -низкий порог вхождения; -кроссплатформенность; -поддержка веб-сервисов; -бесплатное распространение; -постоянное развитие; -наличие учебных материалов. 	<ul style="list-style-type: none"> -узкопрофильность; -безопасность; -противоречия в коде; -имеет слабые средства для работы с исключениями.
C++	<ul style="list-style-type: none"> -высокая производительность; -кроссплатформенность; -многопоточность; -поддержка разных парадигм программирования; -доступность литературы; -бесплатность; 	<ul style="list-style-type: none"> -низкоуровневость; -высокий порог вхождения; -запутанный и громоздкий синтаксис; -сложность нахождения ошибок; -плохая поддержка модульности.

Для создания интернет-ресурса был выбран язык программирования JavaScript. Так как этот язык обеспечивает высокую производительность и скорость работы, отлично работает с другими языками, а также скрипты поддерживаются всеми популярными браузерами и имеет удобства в пользовательских интерфейсах

Веб-сервер позволит нашему интернет-ресурсу работать на PHP. Чаще всего, провайдерами предлагается использовать сервер Apache. В таком сервере имеется поддержка PHP и MySQL.

SQLite - это популярная встраиваемая кроссплатформенная база данных, которая поддерживает полный набор команд SQL [15].

MySQL является системой управления реляционными базами данных с открытым исходным кодом. Такая система реализует модель «клиент-сервер» [27].

Таблица 7 - Таблица сравнения баз данных

Критерии	MySQL	SQLite
Безопасность	+	-
Мощность и масштабируемость	+	-
Скорость	+	-
Набор функций SQL	+	+
Проблемы при переносе	-	+

Исходя из вышеприведенного анализа баз данных, в качестве средства реализации БД была выбрана система MySQL. Такая система предоставит большие функциональные возможности, обеспечит безопасность, скорость и производительность, а также обеспечит множественный контроль доступа.

Для подключения базы данных к сайту использовался phpMyAdmin через локальный сервер LocalHost.

PhpMyAdmin является веб-приложением, которое написано на языке PHP. Оно позволяет администрировать СУБД MySQL, запускать команды SQL, а также просматривать и редактировать содержимое таблиц и баз данных.

Это основной инструмент в работе с динамическими сайтами. PhpMyAdmin имеет открытый код и работает через браузер. Интуитивный интерфейс этого веб-приложения позволяет быстро получить доступ к базе данных.

3.1 Описание интерфейса Интернет-ресурса

На данном этапе необходимо представить интерфейс создаваемого интернет-ресурса. Интерфейс должен быть грамотно продуман для того, чтобы пользователь интуитивно мог без проблем пользоваться сайтом [8].

На рисунке 3.1 изображена стартовая страничка интернет-ресурса. На этой вкладке продемонстрирован «Каталог».

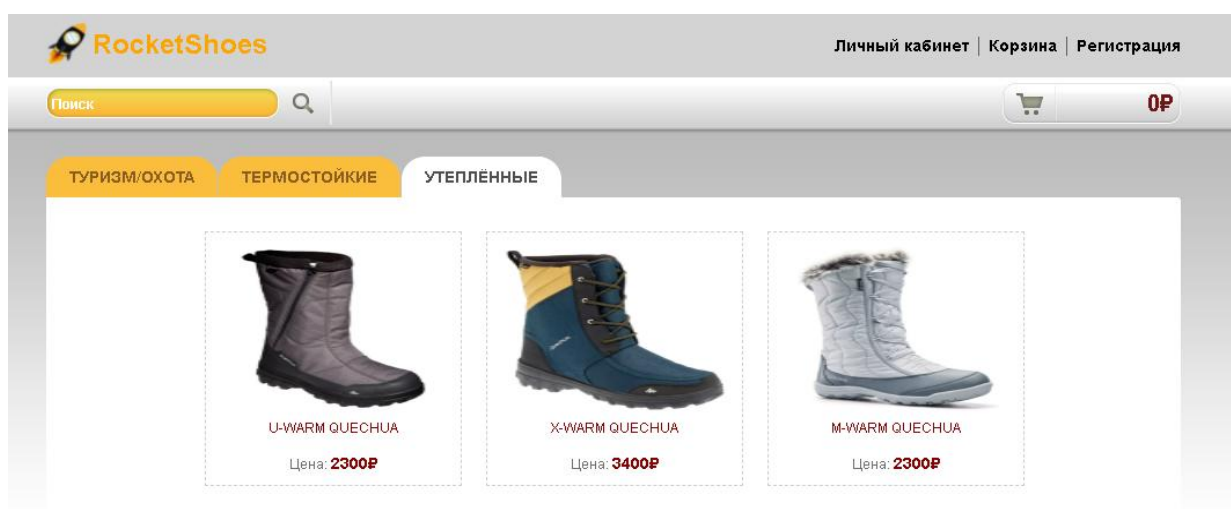


Рисунок 3.1 - Вкладка «Каталог»

На этой странице отображаются все товары, распределенные по категориям, которые есть в системе с указанием их цен и наименований.

На рисунке 3.2 изображена страница регистрации.

Регистрация в личном кабинете

Ваш логин:

Ваш пароль:

Регистрация

Уже зарегистрированы? [Войти в личный кабинет](#)

Рисунок 3.2 - Страница регистрации

Изм.	Лист	№ документа	Подпись	Дата

ИФСТ. 466452.011 ПЗ

Лист
38

На этой странице есть два текстовых поля, куда пользователь должен ввести свой логин и пароль. Если у клиента уже есть аккаунт, то под регистрацией имеется приглашение на ссылку авторизации.

Рисунок 3.3 содержит изображение страницы авторизации. На этой странице имеется два текстовых поля, которые предназначены для ввода данных.

Вход в личный кабинет

Ваш логин:

Ваш логин:

Ваш пароль:

Ваш пароль:

Войти

Вы так же можете зарегистрироваться

Рисунок 3.3 - Страница авторизации

На рисунке 3.4 изображена вкладка «Корзина». Эта страница отображает информацию о добавленных товарах, позволяет удалять товар из корзины, а также позволяет произвести оформление заказа.

Личный кабинет | Корзина | Регистрация

Поиск

156300Р

Оформить заказ

ELKLAND

Цена: 4000Р

Убрать из корзины

dsfsdf

Цена: 150000Р

Убрать из корзины

M-WARM QUECHUA

Цена: 2300Р

Убрать из корзины

Рисунок 3.4 - Вкладка «Корзина»

Далее, на рисунке 3.5 представлена карточка товара.

ELKLAND



Изготовлены из водостойкого полиэстера и микрофибры высокой плотности. Боковая поддержка и фиксация стопы, что является важным аспектом в условиях неровных поверхностей. Внутренняя часть ботинка выполнена из синтетических нитей – Mesh, а также мембранной ткани, которые позволяют ногам оставаться сухими даже в очень влажную погоду, а также быть комфортными в носке при температуре от 0 до + 30 градусов.

В корзину
4000Р

Комментарии к товару ELKLAND

Оставьте свой отзыв на товар

Отправить

Рисунок 3.5 - Карточка товара

Эта страница отображает изображение, описание и комментарии выбранного товара. Также присутствует информирование пользователя о цене с возможностью добавления товара в корзину.

На этой странице присутствует возможность комментирования товара, представленная на рисунке 3.6.

Комментарии к товару ELKLAND

Оставьте свой отзыв на товар

Отправить

Sergey

Не пожалел!

Daniil

Взял всей семье!

Рисунок 3.6 - Комментирование товара

Изм.	Лист	№ документа	Подпись	Дата

ИФСТ. 466452.011 ПЗ

Лист

40

Выбрав категорию товара, каталог отфильтрует обувь и предоставит возможность просмотра необходимых пользователю товаров. Категория обуви «термостойкая» представлена на **рисунке 3.7.**

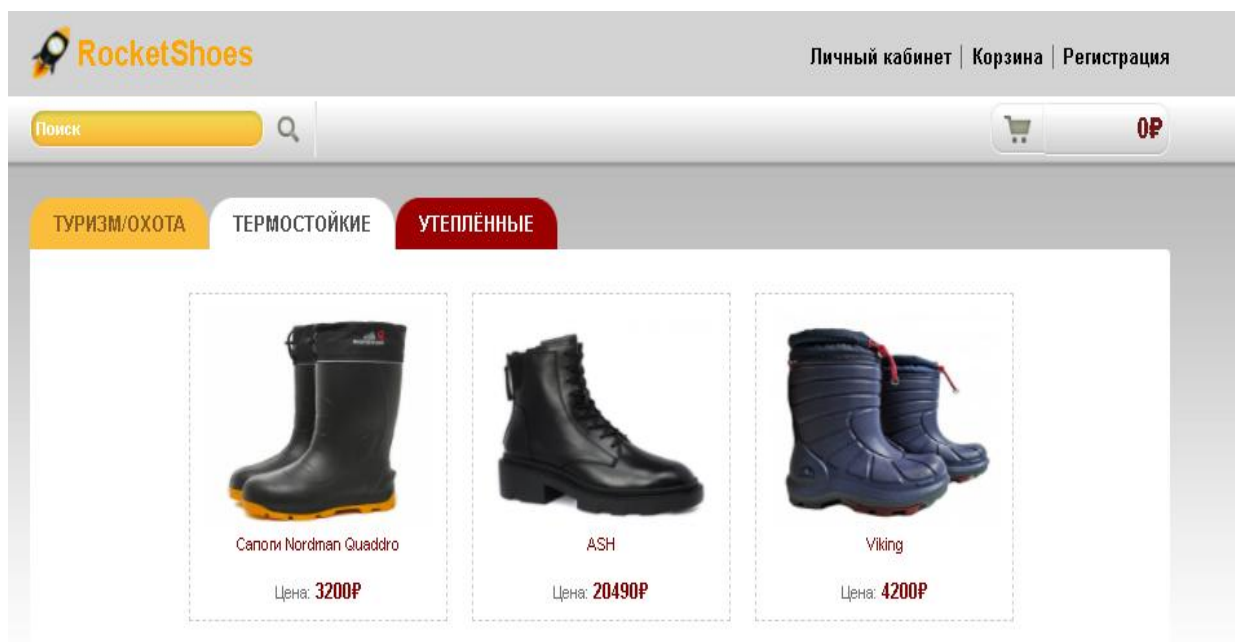


Рисунок 3.7 - Категория обуви «термостойкие»

На **рисунке 3.8** изображен выбор следующей категории товаров.

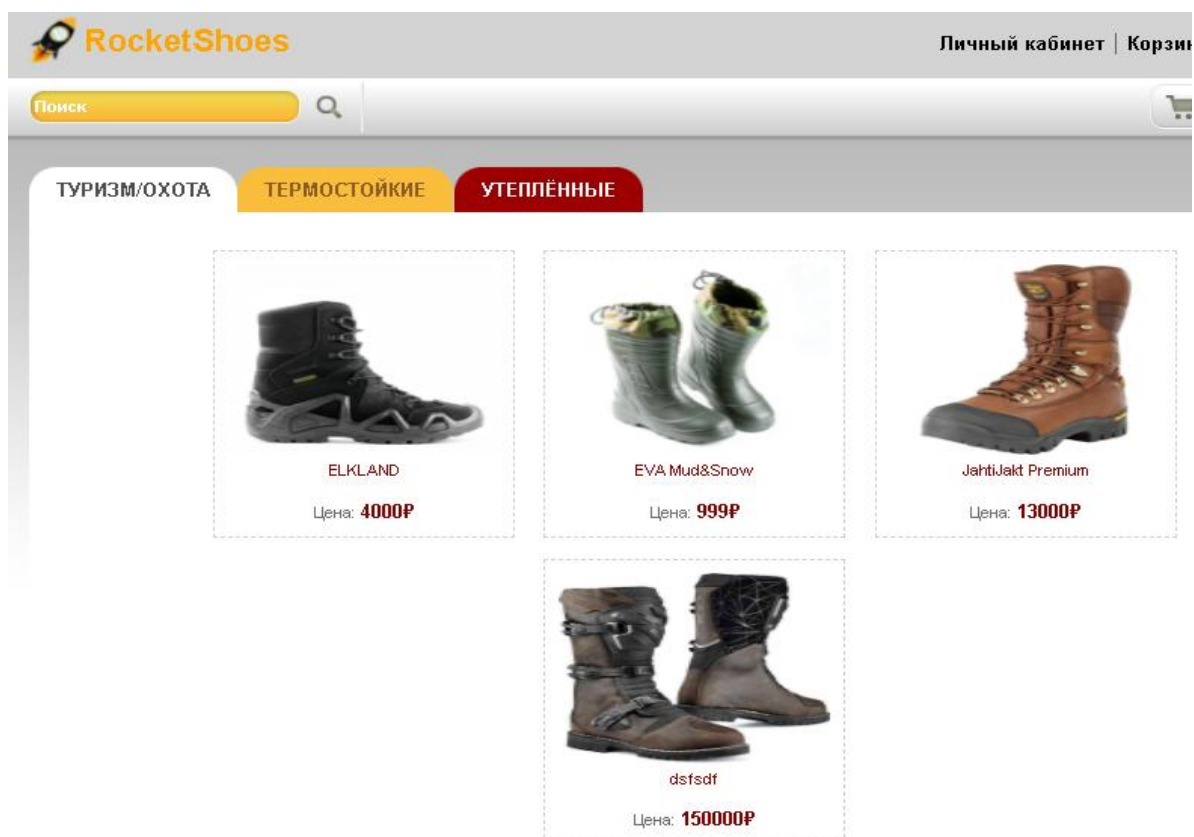


Рисунок 3.8 - Категория обуви «для туризма и охоты»

Изм.	Лист	№ документа	Подпись	Дата

ИФСТ. 466452.011 ПЗ

Лист

41

Следующая категория изображена на рисунке 3.9.

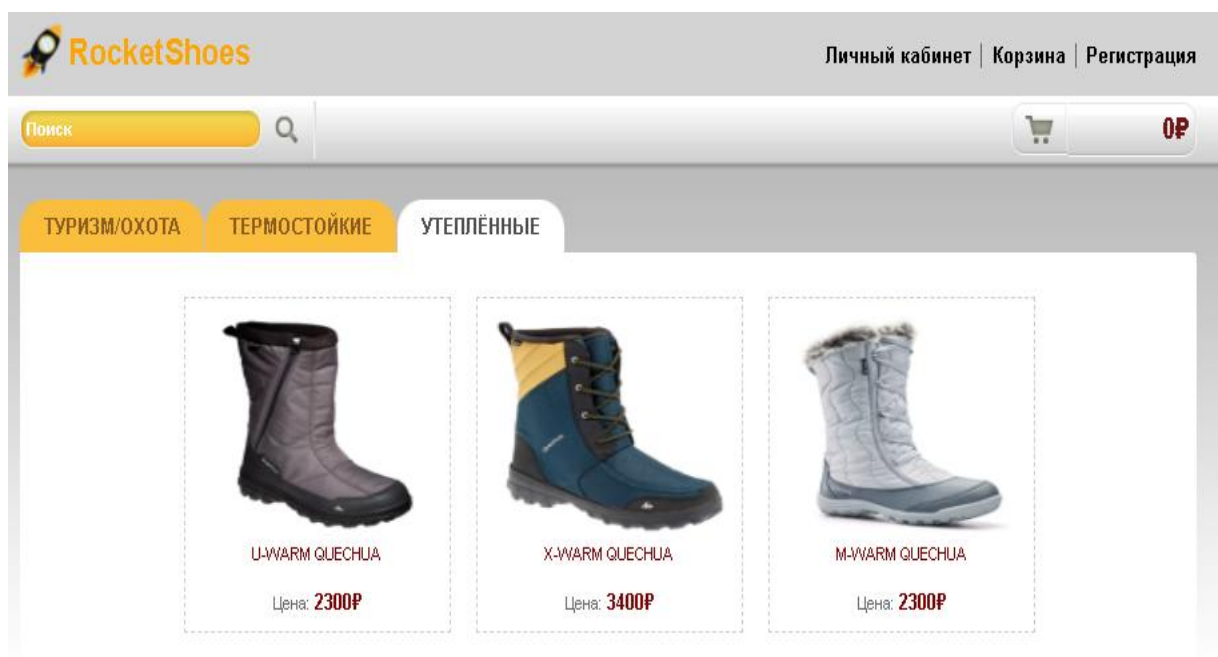


Рисунок 3.9 - Категория обуви «утепленные»

Если авторизоваться под логином админа, то система выдаст административную панель, которая представлена на рисунке 3.10.

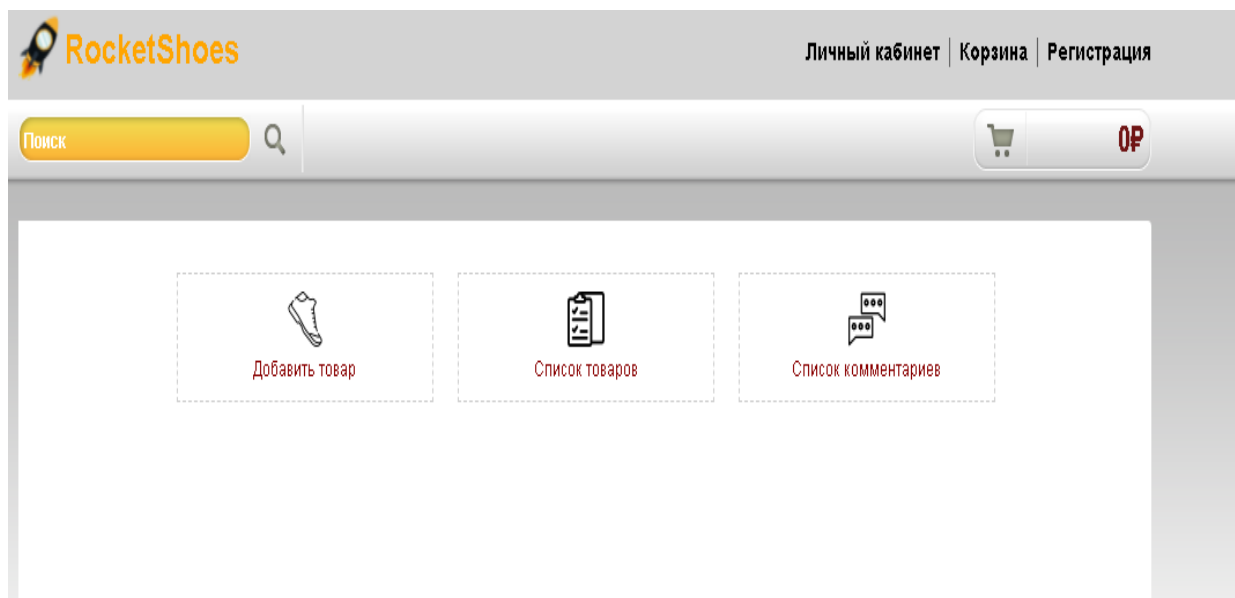


Рисунок 3.10 - Административная панель

Если перейти во меню списка товаров, представленное на рисунке 3.11, то появится страница со всеми добавленными товарами на сайт вместе с их ценами и названиями. На этой вкладке присутствуют возможности редактирования и удаления товаров.

Изм.	Лист	№ документа	Подпись	Дата

ИФСТ. 466452.011 ПЗ

Лист

42

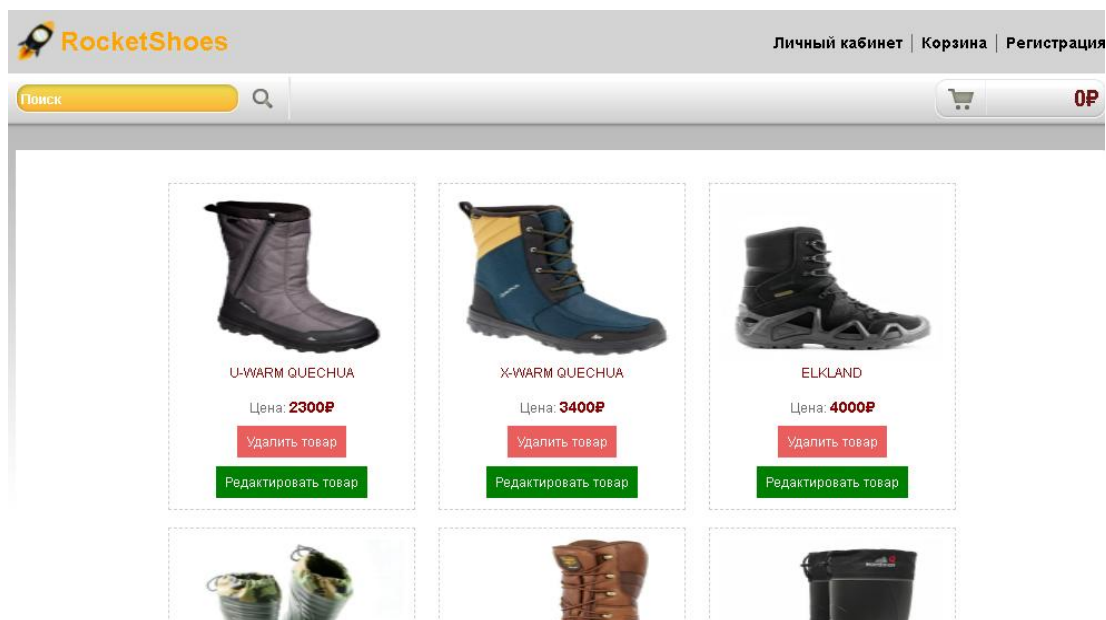


Рисунок 3.11 - Список товаров в административной панели

Если администратор захочет добавить товар, то система выдаст ему страницу, представленную на рисунке 3.12.

Рисунок 3.12 - Добавление товара в административной панели

Эта страница содержит поля для ввода описания товара, его цены, наименования и типа категории.

Страница административной панели управления комментариями представлена на рисунке 3.13.

Изм.	Лист	№ документа	Подпись	Дата

ИФСТ. 466452.011 ПЗ

Лист

43

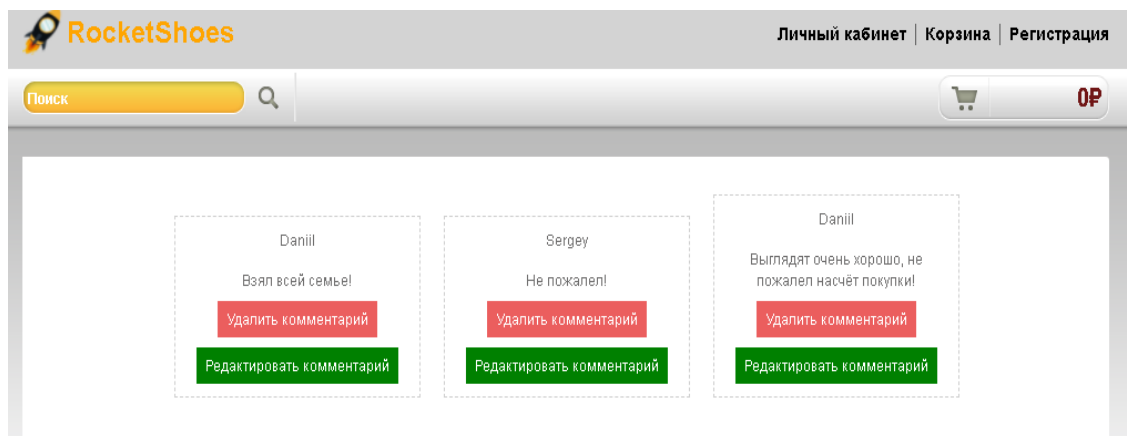


Рисунок 3.13 - Управление комментариями в административной панели

На этой странице отображаются все введенные пользователями комментарии. Помимо этого, присутствуют возможности удаления и редактирования комментариев.

3.2 Разработка программных модулей Интернет-ресурса

Корзина, **представленная на рисунке 3.14**, состоит из следующих структур:

- html;
- css;
- javascript и php.

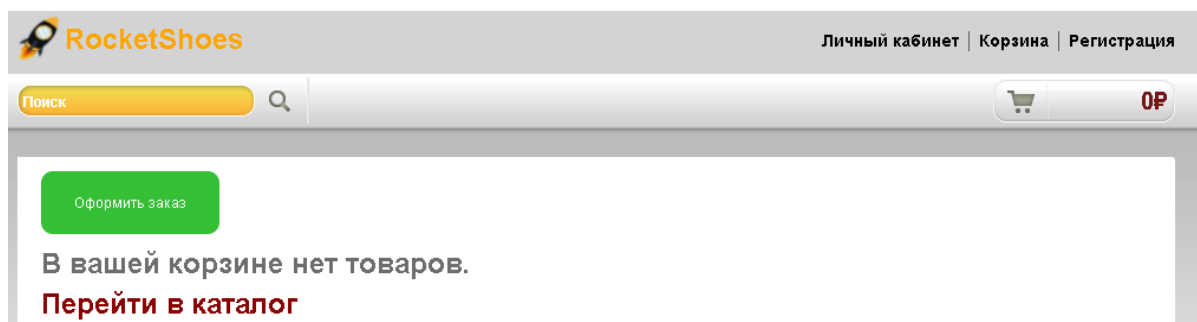


Рисунок 3.14 - Корзина товаров

Классический модуль корзины, как правило, размещают либо в шапке сайта, либо в правом верхнем углу. У такого модуля имеется индикатор, который показывает общую стоимость внесенных товаров.

Для создания класса и отображения страницы корзины используется язык HTML.

Код:

```
<div id="content">
<div id="container">
<div class="tabbed">
<a href="order.php" class="button">Оформить заказ</a>
<br /><br /><br />
```

Данный код находится в файле index.html

Далее этому классу добавлялись стили, чтобы оптимизировать и изменить ее внешний вид, посредством языка CSS.

```
.cart { float:left; position:relative; top:2px; }
.cart strong{ float:left; width:95px; height:36px; background:url(images/cart-price.gif); text-align: right; font-size:20px; color:#701616; line-height:36px; padding:0 7px 0 5px;}
.cart-ico{ float:left; width:47px; height:36px; background:url(images/cart.gif) no-repeat 0 0; font-size:0; line-height:0; text-indent: -4000px;}
```

Данный код находится в файле style.css.

Далее перейдем в системную структуру корзины и с помощью языков PHP добавим функционал удаления, дообновления и занесения карточек товара в корзину.

Код для добавления товара:

```
<?php
if (strpos ($_SERVER["HTTP_REFERER"], "remove_item.php")) {
    print "<h2 align='center' style='color:green;'>Товар успешно удалён из корзины</h2><br /><br />";
}
if (empty ($_SESSION["uid"])) {
    print "<h2>Для того, чтобы просмотреть свою корзину, вам необходимо <a href='auth.php'>войти</a> или <a href='register.php'>зарегистрироваться</a></h2>";
}
else {
```

Изм.	Лист	№ документа	Подпись	Дата

ИФСТ. 466452.011 ПЗ

Лист

45

```

if (mysqli_num_rows ($cart_data) == 0) {
    print "<h1>В вашей корзине нет товаров.<br /><br /><a href='/'>Перейти в
каталог</a></h1>"; }
else {
    print '<div class="tab-content">
<div class="items">
<div class="cl">&nbsp;</div>';
    while ($cart_item = mysqli_fetch_assoc ($cart_data)) {
        $product_item = mysqli_fetch_assoc (mysqli_query ($db, "SELECT * FROM
`product` WHERE `product_id`='{ $cart_item[\"product_id\"]}'"))
        ?>

```

Данный код позволяет обновлять данные, которые содержатся в модуле корзины. Информация о товарах в корзине содержится в массиве \$_SESSION, представляющий собой ассоциативный массив, в котором ключами являются ID продукции, а значением - количество товаров, которые добавлены в корзину.

3.3 Тестирование Интернет-ресурса

После разработки интернет ресурса перед сдачей его в эксплуатацию, необходимо протестировать его работу, а также рассмотреть его адаптивность.

Под адаптивностью понимается способность сайта, **представленная на рисунке 3.15**, которая позволяет подстроиться под размеры экрана пользовательского устройства.

Такой ресурс должен обеспечивать хорошее отображение на экране компьютера, планшета и телефона [29].



Рисунок 3.15 - Виды адаптаций ресурсов под гаджеты

На рисунке 3.16 изображена адаптация под планшет.

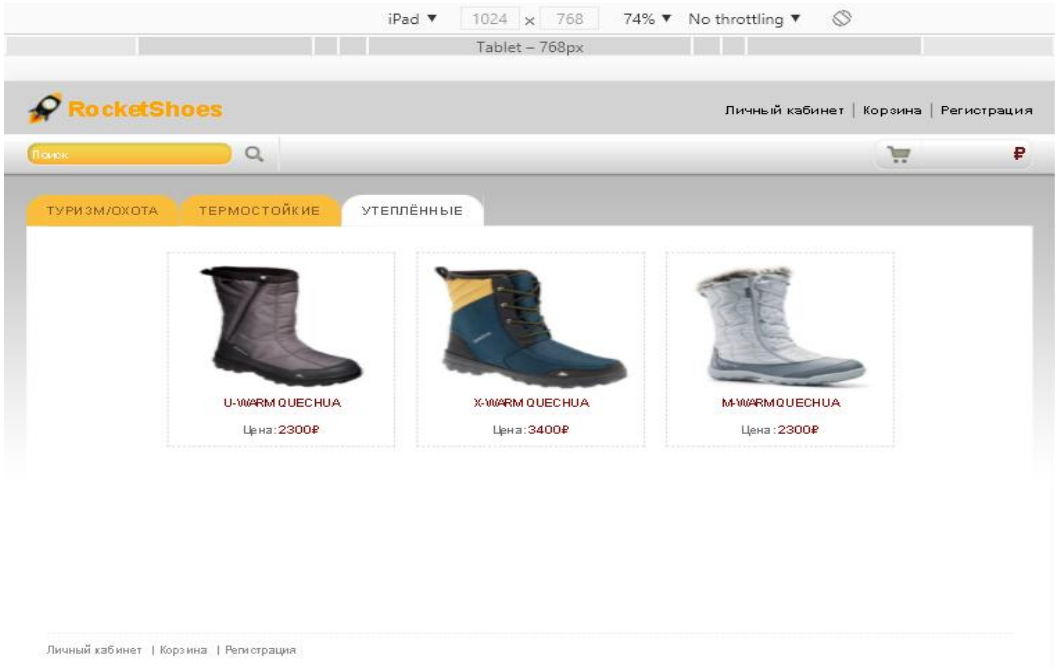


Рисунок 3.16 - Адаптация сайта под планшет

На рисунке 3.17 изображена адаптация под мобильное устройство.

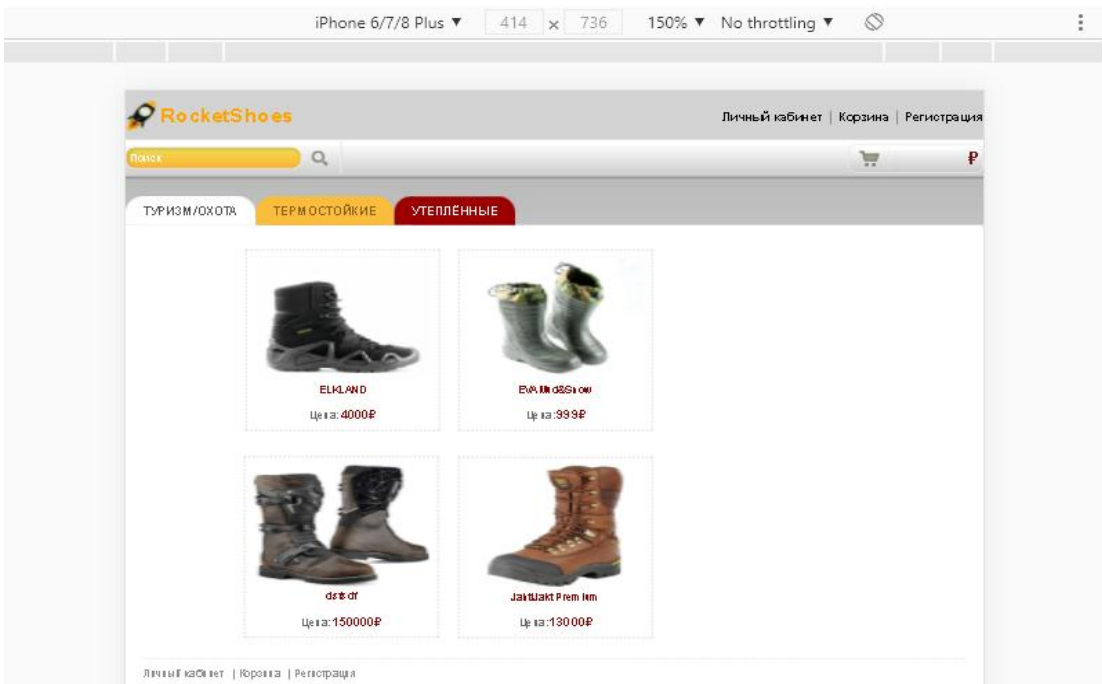


Рисунок 3.17 - Адаптация под мобильное устройство

Основная цель тестирования - проверить работу ресурса на соответствие требованиям:

- 1) обеспечение правильного поиска товаров;
- 2) наличие корзины для выбранных пользователем товаров;
- 3) возможность оформления заказа;

4) занесение чека в историю покупок.

Если ввести в поисковую строку название обуви, то система выдаст именно ту обувь, которую пользователь искал.

Результат этих действий изображен на рисунке 3.18.

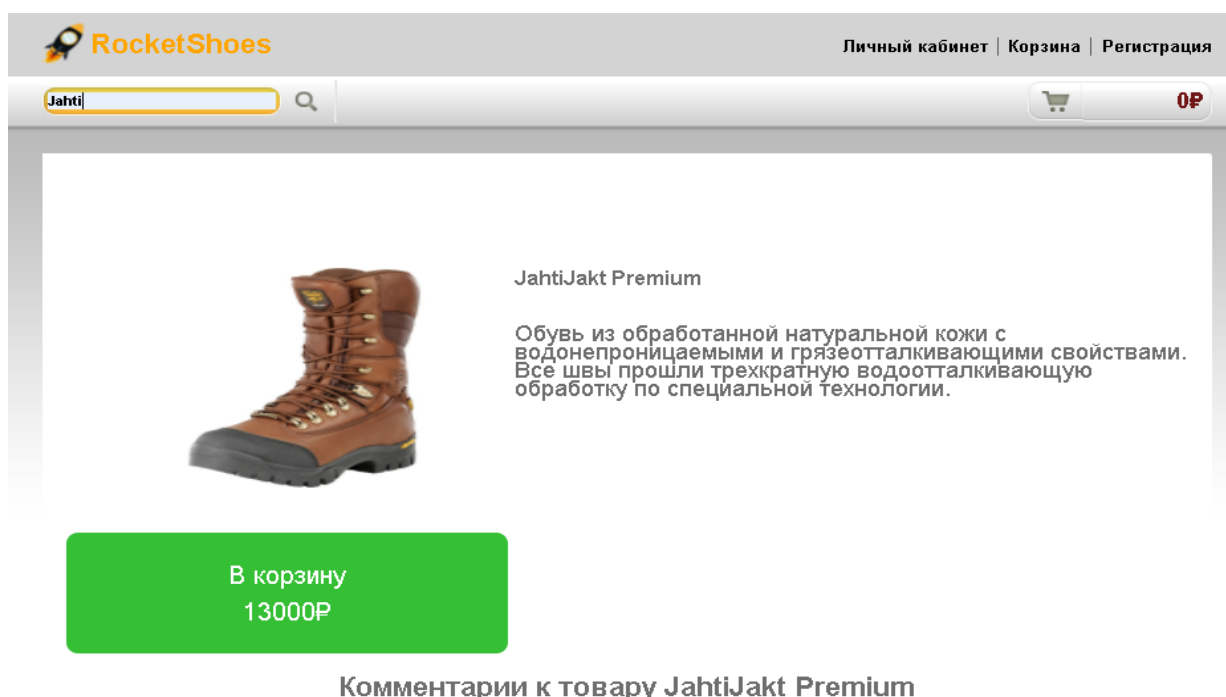


Рисунок 3.18 - Тестирование поиска товара

При занесении товара в корзину пользователем, система заносит этот товар и выводит краткую информацию по нему, в виде наименования товара и его цены.

На странице корзины пользователю предлагается либо оформить заказ, либо убрать товар из корзины. Тестирование корзины изображено на рисунке 3.19.

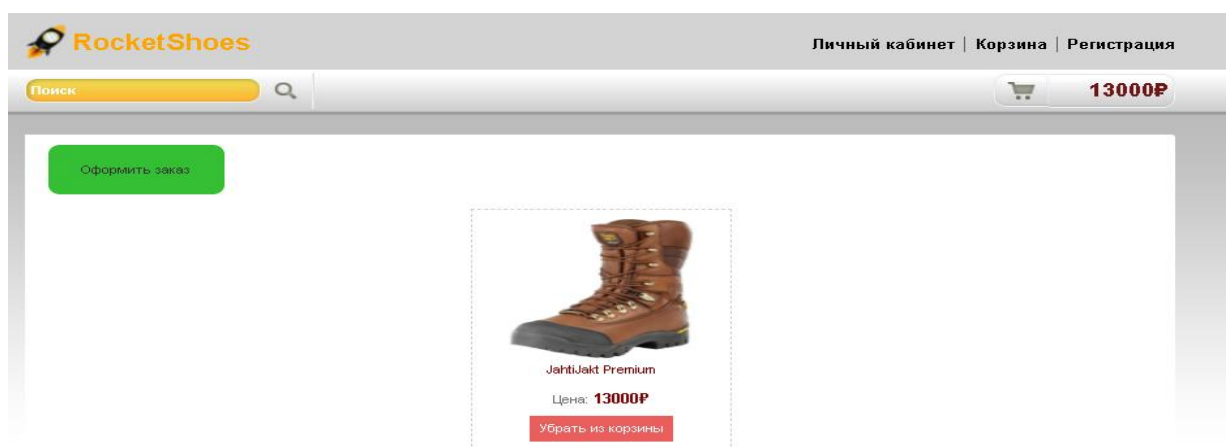


Рисунок 3.19 - Тестирование корзины

Изм.	Лист	№ документа	Подпись	Дата

ИФСТ. 466452.011 ПЗ

Лист
48

Данный чек нужен для оплаты товара на почте. В этом чеке указана следующая информация:

- ФИО заказчика;
- номер телефона;
- адрес доставки товара;
- почтовый индекс;
- e-mail;
- общая сумма заказа.

По итогам тестирования разрабатываемого интернет-ресурса можно сделать вывод, что сайт работает корректно и выполняет все поставленные требования.

					ИФСТ. 466452.011 ПЗ	Лист
						50
Изм.	Лист	№ документа	Подпись	Дата		

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были выполнены следующие задачи:

- был проведен анализ функций и требований к интернет-ресурсу;
- проведен обзор существующих методов создания интернет-ресурсов;
- была спроектирована навигационная структура интернет-ресурса;
- произведен обзор и выбор инструментальных средств;
- был разработан интернет-ресурс;
- был описан и протестирован интерфейс интернет-ресурса.

					ИФСТ. 466452.011 ПЗ	Лист
						51
Изм.	Лист	№ документа	Подпись	Дата		

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Никсон Р., Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. — СПб.: Питер - 2016. - 496 с.
2. Правовое регулирование интернет торговли. — Текст: электронный // upside — 2021. — URL: <https://upside.pro/lab/pravovoe-regulirovanie-internet-torgovli.html> (дата обращения 05.04.2021).
3. Способы разработки веб-сайта. — Текст: электронный // vc.ru — 2021. — URL: <https://vc.ru/dev/78714-3-sposoba-razrabotki-veb-sayta> (дата обращения 02.04.2021).
4. Язык программирования Java . — Текст: электронный // hexlet — 2021. — URL: <https://ru.hexlet.io/blog/posts/yazyk-programmirovaniya-java-osobennosti-populyarnost-situatsiya-na-rynke-truda> (дата обращения 06.04.2021).
5. Что такое Java Script. — Текст: электронный // developer.mozilla — 2021. — URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics (дата обращения 07.04.2021).
6. Зандстра М., PHP. Объекты, шаблоны и методики программирования. — М.: «Вильямс», 2016. - 560 с.
7. Разработка веб-сайтов. — Текст: электронный // sales-generator — 2021. — URL: <https://sales-generator.ru/blog/razrabotka-veb-saytov> (дата обращения 01.04.2021).
8. Характеристики успешного интерфейса — Текст: электронный // ruseller — 2021. — URL: <https://ruseller.com/lessons.php?rub=35&id=616> (дата обращения 12.05.2021).
9. Программирование на JavaScript — Текст: электронный // roi4cio — 2021. — URL: <https://roi4cio.com/categories/category/programmirovanie-na-javascript/> (дата обращения 15.05.2021).
10. Основы HTML — Текст: электронный // developer.mozilla — 2021. — URL:

https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics (дата обращения 11.05.2021).

11. Сравнение JavaScript и Java – Текст: электронный // bitdegree – 2021. — URL: <https://ru.bitdegree.org/rukovodstvo/java-i-javascript/#heading-3> (дата обращения 08.05.2021).

12. Статические и динамические сайты. – Текст: электронный // moolkin – 2021. — URL: <http://moolkin.ru/joomla/cms/staticheskie-i-dinamicheskie-web-sayty-v-chyom-raznitsa/> (дата обращения 25.04.2021).

13. Методы создания сайта. – Текст: электронный // ravechnost – 2021. — URL: <https://ravechnost.ru/metody-sozdaniya-sajta-3-sposoba-sozdat-sajt/> (дата обращения 04.05.2021).

14. Сравнение языков программирования. – Текст: электронный // proglib.io – 2021. — URL: <https://proglib.io/p/top-10-programming-languages-2019> (дата обращения 02.05.2021).

15. Сравнение реляционных СУБД. – Текст: электронный // tproger – 2021. — URL: <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/> (дата обращения 10.05.2021).

16. Плюсы и минусы разработки сайтов на конструкторах – Текст: электронный // uguide – 2021. — URL: <https://uguide.ru/sajty-na-konstrukto-re-pljusy-i-minusy> (дата обращения 11.05.2021).

17. Преимущества самостоятельной разработки сайтов – Текст: электронный // art-reklama – 2021. — URL: https://art-reklama.net/blog/samostoiatelnoe_sozdanie_saita_pliusy_i_minusy (дата обращения 15.05.2021).

18. Что такое CMS – Текст: электронный // hostiq.ua – 2021. — URL: <https://hostiq.ua/wiki/cms/> (дата обращения 07.04.2021).

19. Конструкторы сайтов – Текст: электронный // hostiq.ua – 2021. — URL <https://hostiq.ua/wiki/about-sitebuilder/> (дата обращения 07.04.2021).

20. Методы создания интернет-ресурсов. – Текст: электронный // webstudio2u – 2021. — URL: <https://webstudio2u.net/ru/webdesign/354-site-development-methods.html> (дата обращения 02.05.2021).

21. Классы UML. – Текст: электронный // flexberry – 2021. — URL: https://flexberry.github.io/ru/gpg_practical-guides-uml.html (дата обращения 20.04.2021)

22. Диаграммы состояний. – Текст: электронный // itteach – 2021. — URL: <https://itteach.ru/rational-rose/diagrammi-sostoyaniy> (дата обращения 20.04.2020)

23. Диаграммы компонентов. – Текст: электронный // sites – 2021. — URL: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema15/tema15_2 (дата обращения 20.04.2020)

24. Диаграмма развертывания. – Текст: электронный // flexberry – 2021. — URL: https://flexberry.github.io/ru/fd_deployment-diagram.html (дата обращения 20.04.2020)

25. Чаффер Д., Эффективная веб-разработка на JavaScript. Символ-плюс - М., 2017. - 391 с.

26. Диаграммы вариантов использования. – Текст: электронный // sites.google.com – 2021. — URL: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12/tema12_2

27. Яргер Р.Дж., MySQL. Базы данных для небольших предприятий и Интернета / Р.Дж. Яргер, Дж. Риз, Т. Кинг. - М.: СПб: Символ-Плюс, 2016. - 560 с.

28. Разработка статических сайтов – Текст: электронный // PUZZLEWEB: официальный сайт. – 2021. — URL: https://puzzleweb.ru/php/00_teacher.php (дата обращения: 23.05.2021).

29. Адаптивность сайта – Текст: электронный // UXРАРА: официальный сайт. – 2021. — URL: <https://uxrapa.ru/kak-sdelat-adaptivnyy-sayt.html> (дата обращения: 23.05.2021).

30. Минусы создания сайтов на CMS. – Текст: электронный // JINO: – 2021. — URL: <https://jino.ru/journal/articles/plyusy-minusy-sayty-cms/> (дата обращения 10.05.2021).

