

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»

ИНТЕРНЕТ-МАГАЗИН ТОВАРОВ ДЛЯ РЕМОНТА

Выпускная квалификационная работа

по направлению подготовки 09.03.02 Информационные системы и
технологии

профилю подготовки «Информационные технологии в медиаиндустрии»

Идентификационный код ВКР: 502

Екатеринбург 2019

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ
Заведующий кафедрой ИС
_____ И. А. Сулова
«__» _____ 2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ИНТЕРНЕТ-МАГАЗИН ТОВАРОВ ДЛЯ РЕМОНТА

Исполнитель:

обучающийся группы ИТМ-403

П. А. Куноф

Руководитель:

канд. пед. наук, доцент кафедры ИС

Н. С. Власова

Нормоконтролер:

ст. преподаватель каф. ИС

Н. В. Хохлова

АННОТАЦИЯ

Выпускная квалификационная работа состоит из интернет-магазина по продаже товаров для ремонта и пояснительной записки на 60 страницах, содержащей 41 рисунок, 2 таблицы, 33 источника литературы, а также 1 приложение на 2 страницах.

Ключевые слова: ИНТЕРНЕТ-МАГАЗИН, OPENCART, CMS.

Куноф П. А., Интернет-магазин товаров для ремонта: выпускная квалификационная работа / П. А. Куноф; Рос. гос. проф.-пед. ун-т, ин-т инж.-пед. образования, каф. информ. систем и технологий. — Екатеринбург, 2019. — 60 с.

В работе приведен анализ популярных решений для создания интернет-магазинов. Обоснован выбор системы управления сайтом OpenCart и рассмотрены вопросы разработки интернет-магазина товаров для ремонта.

Цель выпускной квалификационной работы — разработать интернет-магазин для компании «Ремонтим» по продаже товаров для ремонта.

Для достижения цели были проанализированы современные решения в разработке интернет-магазинов. Рассмотрены вопросы программирования, верстки и тестирования интернет-магазина.

Принимая во внимание динамичное развитие интернет-торговли в России, интернет-магазин, осуществляющий продажу товаров через Интернет, становится одним из ключевых и перспективных торговых каналов для ведения бизнеса.

СОДЕРЖАНИЕ

Введение.....	4
1 Аналитическая часть.....	6
1.1 Роль интернет-магазина в современном мире	6
1.2 Сравнение существующих и самостоятельно созданных разработок.....	8
1.3 Наиболее популярные бесплатные системы управления для интернет-магазина.....	10
1.4 Обоснование выбора систем.....	17
1.5 Общий алгоритм реализации интернет-магазина.....	21
2 Проектная часть.....	23
2.1 Характеристика потенциальной аудитории проекта.....	23
2.2 Постановка задачи проекта	23
2.2.1 Актуальность проекта.....	23
2.2.2 Цель и назначение проекта	24
2.2.3 Функционал проекта, интерфейс проекта, эргономические и системные требования и другие требования к проекту	24
2.2.4 Входные данные.....	26
2.2.5 Характеристики оборудования для реализации проекта.....	29
2.3 Жизненный цикл проекта.....	29
2.3.1 Этап разработки элементов дизайна проекта.....	29
2.3.2 Этап разработки программного кода проекта.....	36
2.3.3 Этап тестирования, апробирования, внедрения.....	51
2.4 Технические требования к проекту	52
2.5 Калькуляция проекта	53
Заключение	54
Список использованных источников	56
Приложение	59

ВВЕДЕНИЕ

Наличие сайта в Интернете является неотъемлемой частью любой крупной компании, организации или частного предпринимателя. А наличие собственного интернет-магазина — это выгодное вложение денежных средств для развития своего бизнеса. На сегодняшний день большинство людей предпочитает купить товар через Интернет, не тратя время на походы по магазинам. Наличие интернет-магазина имеет ряд преимуществ:

1. Не требует затрат на аренду площади, найма персонала.
2. Любой покупатель, не зависимо от места проживания, имеет доступ ко всему, что представлено на сайте.
3. Гораздо ниже стоимость и сроки создания интернет-магазина, чем традиционного.
4. Покупатель может получить всю необходимую ему информацию без посредников, в то же время, имеет возможность связаться с компанией для уточнения каких-либо вопросов.
5. Нет ограничений по количеству товаров или информации для размещения на сайте.
6. Возможность совершать покупки, не выходя из дома. При наличии своего действующего интернет-магазина компания делает свой ассортимент более доступным для широкого круга целевой аудитории.

Объект выпускной квалификационной работы — средства создания, поддержки, разработки и администрирования интернет-магазина.

Предмет выпускной квалификационной работы — интернет-магазин, осуществляющий продажу товаров для ремонта.

Цель выпускной квалификационной работы — разработать интернет-магазин для компании «Ремонтим» по продаже товаров для ремонта.

Задачи:

- провести анализ предметной области;

- провести анализ существующих систем управления сайтом для создания интернет-магазина;
- выбрать технологию реализации интернет-магазина;
- разработать структуру интернет-магазина;
- реализовать разработанную структуру средствами web-технологий;
- наполнить сайт контентом;
- протестировать интернет-магазин.

В процессе выполнения проекта необходимо создать интернет-магазин по продаже товаров для ремонта «Ремонтим». В нем должна присутствовать возможность размещения, просмотра, оформления товаров и понятный интерфейс.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Роль интернет-магазина в современном мире

Электронный магазин (виртуальный магазин, интернет-магазин) можно рассматривать как прикладную систему, которая построена как элемент технологии системы электронной коммерции. Подобно обычному магазину, электронный магазин реализует следующие основные функции: представление товаров (услуг) покупателю, обработку заказов, продажу и доставку товаров [16].

В Интернете часто встречается другое определение этого понятия, более примитивное — сайт, торгующий товарами и услугами в сети Интернет. Электронный магазин позволяет пользователям сформировать заказ на покупку, выбрать способ оплаты и доставки заказа в сети Интернет.

Главным отличием интернет-магазина от обычного магазина является его расположение и организация взаимодействия с покупателем, использование сети Интернет для осуществления всех возможных операций.

Существуют три основные системы интернет-торговли:

1. Web-витрина — оформленный при помощи средств web-дизайна прайс-лист коммерческой компании с возможностью только приема заказов. Такой тип системы лучше всего подходит для организаций малого и среднего бизнеса. Web-витрины не имеют интерактивных интерфейсов управления и никак не связаны ни с какими приложениями компании [7]. Обмен информации с web-витриной делается менеджером вручную, при помощи файлообменников.

2. Интернет-магазин — полнофункциональная система ведения интернет-торговли с индивидуально разработанными бизнес-схемами взаимодействия с покупателями. Интернет-магазин имеет, помимо web-витрины, всю необходимую структуру и алгоритмы для управления бизнес-процессом.

Интернет-магазин подходит для предприятий, обладающих большим ассортиментом и набором товаров и услуг, компаниям среднего и крупного бизнеса [14].

3. Торговая интернет-система (ТИС) — сложная и мощная система ведения торговых операций через Интернет, в которой полностью интегрированы торговые процессы offline и системы online бизнеса компании. Достигается это путем интеграции внешнего интернет-магазина в бизнес-приложения, имеющиеся в компании, с системами складского учета, торговочетными и бухгалтерскими системами [6].

Из вышеперечисленных видов систем интернет-торговли, в России практически нет ТИС, крайне мало интернет-магазинов, при этом большое количество web-витрин.

Функции интернет-магазина:

- детальная демонстрация покупателям всех характеристик и преимуществ продаваемых товаров и услуг, возможностей их применения и различных вариантов приобретения;
- проведение консультаций потребителям;
- прием заявок и обработка заказов, подготовка и оформление документов;
- доставка потребителю необходимых товаров и услуг;
- ведение детальной статистики по деятельности компании;
- ведение базы данных;
- анализ деятельности компании, составление аналитических отчетов и прогнозов;
- возможность расширения деятельности компании, выстраивание торговой сети;
- поиск и выход на новые рынки сбыта.

Преимущество интернет-коммерции и интернет-технологий (ИТ) в сравнении с традиционными методами ведения коммерции заключается в том, что большая часть процессов торговли протекают в интерактивном ре-

жине и не сопровождаются оформлением различных документов на бумажных и цифровых носителях, а заменяются электронным документооборотом. Что в свою очередь позволяет увеличивать скорость и соответственно объемы торговли [16].

Появляется возможность сравнить цены на товар сразу в нескольких интернет-магазинах, при этом никуда не надо ходить.

Для жителей малых городов и деревень с небольшим количеством магазинов и скудным выбором товара интернет-магазин позволяет заказать именно тот товар, который им нужен, всего в пару действий и в несколько дней ожидания доставки [14].

1.2 Сравнение существующих и самостоятельно созданных разработок

На сегодняшний день способ создания сайта на базе Content Management System (CMS) активно используется многими разработчиками. Количество CMS перевалило за сотню, они могут отличаться по структуре, ориентации на определенный функционал (например, система для создания блога или новостных ресурсов, система для интернет-магазинов), а также по языку, на котором они написаны.

В зависимости от задачи, обусловленной бизнес-процессом, можно подобрать оптимальную CMS, в которой уже реализована большая часть функционала, чтобы сократить время на разработку.

Одним из преимуществ, которым обладают CMS, заключается в возможности максимально эффективно и за короткие сроки сделать рабочий продукт.

Это обусловлено тем, что, используя готовый функционал CMS и дополнительные модули, разработчик, концентрируется на конечном результате, который должна выполнять система, а не на её решении.

В зависимости от задачи, обусловленной бизнес-процессом, можно подобрать оптимальную CMS, в которой уже реализована большая часть функционала, чтобы сократить время на разработку [20].

К тому же общеизвестные CMS тестировались на протяжении многих лет и до сих пор обновляются. За это время исправлены все критические важные ошибки, что ведет к надежной и стабильной работе продукта.

На начальном этапе разработки web-сайта возникает вопрос использовать существующую систему управления контентом или разработать свою под конкретную задачу. Для того, чтобы определиться, были рассмотрены преимущества и недостатки каждой из сторон, представленные в таблице 1.

Таблица 1 — Сравнение готовых и самостоятельно разработанных систем

CMS	Достоинства	Недостатки
Существующие готовые системы	<ul style="list-style-type: none"> • в свободном доступе можно найти ответы на многие вопросы в сообществе разработчиков; • установить и настроить готовую систему гораздо быстрее, чем разработка системы; • для популярных CMS существует множество разнообразных шаблонов и дополнительных модулей, расширяющих функциональность. 	<ul style="list-style-type: none"> • уязвимость. Если пропустить обновление безопасности, то вероятнее всего сайт будет взломан; • еще один недостаток — это наличие похожих сайтов; • не все дополнительные модули бесплатны.
Самостоятельно разработанная система	<ul style="list-style-type: none"> • возможность доработать отдельные ключевые модули под свой проект, оптимизировав работу; • самостоятельное независимое обновление; • бесплатно; • стабильно работающую систему можно продать или сдавать в аренду. 	<ul style="list-style-type: none"> • отсутствие готовых модулей; • отсутствие документации; • понадобится очень много времени и усилий на разработку системы.

Исходя из всего вышеперечисленного можно сделать вывод, что для создания интернет-магазина с наименьшими финансовыми и временными затратами лучше воспользоваться существующей системой управления.

1.3 Наиболее популярные бесплатные системы управления для интернет-магазина

Обозначим термины системы управления сайтом и плагина для дальнейшего сравнения существующих разработок.

Content Management System (CMS) — система управления сайтом, которая используется для организации и обеспечения процесса по совместному созданию, управлению и редактированию содержимого сайта.

Плагин — независимый программный продукт, подключаемый к CMS, используется для расширения недостающих возможностей системы.

В зависимости от задачи, обусловленной бизнес-процессом, можно подобрать оптимальную CMS, в которой уже реализована большая часть функционала, чтобы сократить время на разработку.

Различные CMS были созданы для того, чтобы упростить процесс управления контентом на web-площадке. На сегодняшний день совсем не обязательно разбираться в сложных кодах и языках программирования, чтобы создать свой сайт. Достаточно иметь представление о системе управления, и уметь правильно настроить панель администрирования системы, а также знать, как прописать стили, чтобы в результате получился красиво оформленный и удобный в использовании web-сайт.

Основные функции CMS (системы управления контентом):

Преимущества системы управления контентом:

- огромный выбор решений и гибкий подход к каждой задаче;
- сокращение средств и времени на разработку web-сайта — многие решения уже готовы, остается только подключить и настроить;
- позволяет управлять контентом владельцу сайта без помощи программистов;
- стабильная работа системы, обеспечение регулярных обновлений и адаптация к новшествам в автоматическом режиме.

На данный момент существует огромное количество как платных, так и бесплатных CMS, обладающих различным набором функций.

У каждой CMS есть свой набор инструментов и функциональных возможностей, свой вид интерфейса и набор шаблонов. Существует множество CMS, которые можно условно разделить на три категории: коммерческие, Open-source CMS и индивидуальные (студийные) CMS. Рейтинг наиболее популярных CMS представлены на рисунке 1.

Тип распространения:		Тип CMS:		Тип сайта:	
Все		Бесплатные (1)		Интернет-магазин (19)	
Отрасль (тематика):					
Все					
сравнить	CMS	ТИЦ сайта	Кол-во работ	Яндекс X	средний ТИЦ
<input type="checkbox"/>	1. Opencart	650	3 142	118 232	38
<input type="checkbox"/>	2. Joomla!	1 000	1 756	76 486	44
<input type="checkbox"/>	3. WordPress	3 700	1 473	33 660	23
<input type="checkbox"/>	4. MODX	400	1 187	50 417	42
<input type="checkbox"/>	5. Drupal	300	871	42 637	49
<input type="checkbox"/>	6. PrestaShop	450	427	17 581	41
<input type="checkbox"/>	7. Moguta.CMS	450	315	9 660	31
<input type="checkbox"/>	8. ExpertPlus CMS	90	310	11 670	38
<input type="checkbox"/>	9. Magento	0	293	72 950	249
<input type="checkbox"/>	10. Evolution CMS	70	287	18 693	65

Рисунок 1 — Самые популярные системы управления сайтом в России, используемые для создания интернет-магазинов, по рейтингам «CMS Magazine»

Рассмотрим несколько наиболее популярных бесплатных систем управления для интернет-магазинов. Основные требования, а также достоинства и недостатки каждой из систем.

OpenCart — CMS с открытым исходным кодом, созданная специально для создания интернет-магазинов (рисунок 2). Предлагает расширение базовой функциональности за счёт платных и бесплатных модулей от сторонних разработчиков. Обладает большим пользовательским сообществом, которое помогает развивать проект и формирует базу знаний, разбирая возникающие вопросы на тематических форумах и сайтах [4]. Система пользуется популярностью, поэтому многие разработчики с ней хорошо знакомы.

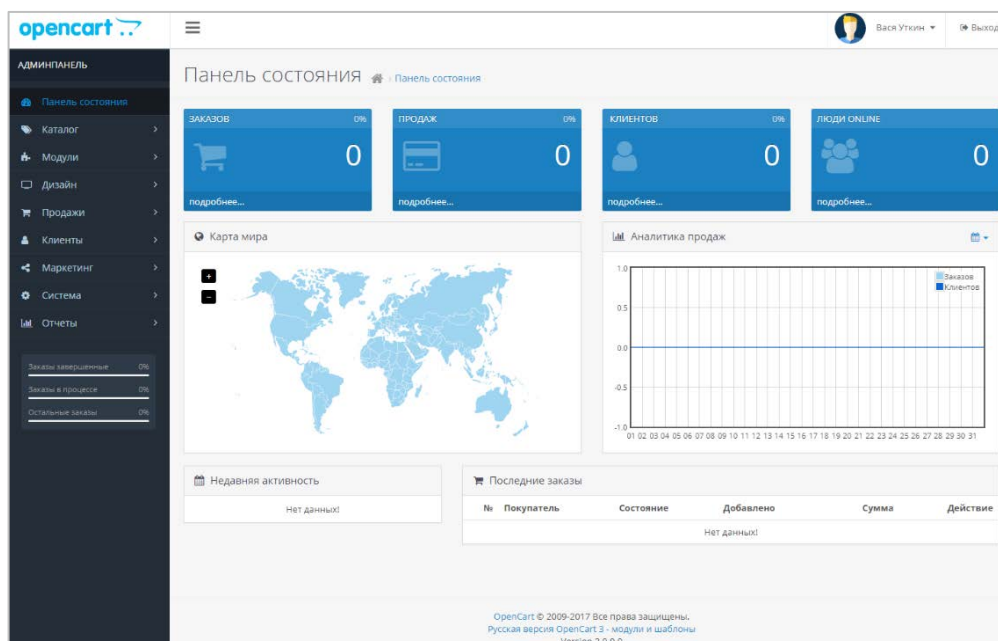


Рисунок 2 — Интерфейс системы OpenCart v.3.0

Его функциональность заточена строго под создание средних и крупных интернет-магазинов, и в этом главное достоинство системы: в базовой комплектации есть почти все необходимые инструменты для организации продаж и аналитики, а то, чего недостаёт, можно установить в виде модулей от сторонних разработчиков или же разработать самостоятельно.

Преимущества системы OpenCart:

- бесплатная в использовании;
- продвинутый функционал;
- встроенная подробная статистика продаж и посещений магазина;
- встроенная функция сравнения характеристик товаров;
- большое количество параметров и настроек для товаров;
- большой выбор способов оплаты и доставки;
- инструменты резервного копирования и восстановления данных;
- отчёты об ошибках;
- регистрация ошибок;
- огромное количество материалов с обучением.

Недостатки системы OpenCart:

- минимальные настройки SEO без дополнительных расширений;

- отсутствует встроенный конструктор и шаблоны;
- редкие обновления системы;
- старые версии сильно подвержены вирусным атакам через сторонние модули;
- сравнительно низкие возможности встроенного текстового редактора;
- нельзя загружать одновременно несколько изображений.

Анализируя CMS OpenCart, можно сделать вывод, что в этой системе реализован большой набор инструментов как для взаимодействия интернет-магазина с покупателями, так и для поискового продвижения. А такие дополнения, как личный кабинет, бонусные баллы, подарочные сертификаты, почтовая рассылка и комбинации товаров, позволяют использовать систему в реализации крупных проектов.

CMS WordPress + плагин WooCommerce

Изначально CMS WordPress создавалась как система, предназначенная для создания и ведения блогов. Для создания интернет-магазина на WordPress, нужно установить плагин WooCommerce. Доработок и настроек после установки плагина будет достаточно много [15].

Плагин WooCommerce позволяет создавать сколько угодно категорий любой вложенности. Можно задавать скидки, вариации товаров. Есть фильтр по параметрам товара, по цене.

Один товар можно поместить в разные категории, адрес будет один. Сравнения товаров нет. Есть отзывы к товарам с премодерацией. Заказы приходят в панель администратора. Уведомления о них приходят покупателю и администратору магазина.

Существует личный кабинет, но регистрация в нем доступна только в момент оформления заказа (в OpenCart, PrestaShop есть отдельная страница регистрации). Личный кабинет имеет стандартный функционал, как и в других системах.

Анализируя плагин WooCommerce системы WordPress, представленного на рисунке 3, следует отметить, что он включает только базовый функционал. Для профессионального развития интернет-магазина придется устанавливать и настраивать немалое количество дополнительных модулей.

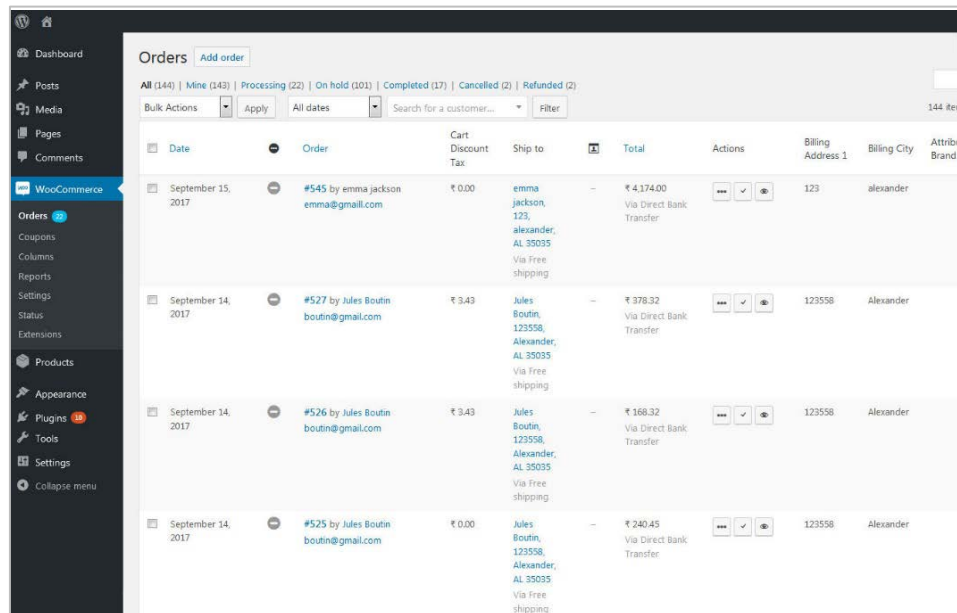


Рисунок 3 — Интерфейс плагина WooCommerce в системе управления сайтом WordPress

Преимущества системы WordPress:

- быстрая установка, настройка и администрирование;
- возможность бесплатного использования;
- простой, интуитивный и удобный интерфейс;
- управление на любом устройстве с доступом к Интернету;
- более 10000 дизайнерских шаблонов, имеются бесплатные;
- более 38000 плагинов под различные задачи;
- встроенный конструктор;
- регулярные и автоматические обновления системы;
- обширная интеграция с различными сервисами;
- огромное количество материалов с обучением.

Недостатки системы WordPress:

- низкая функциональность на базовой версии;
- низкая скорость работы;

- возможна перенагрузка сервера от обилия плагинов;
- минимальная безопасность, если использовать не официальные или бесплатные шаблоны.

Анализируя плагин WooCommerce системы WordPress, следует отметить, что он включает только базовый функционал. Для профессионального развития интернет-магазина придется устанавливать и настраивать немалое количество дополнительных модулей [32].

CMS Joomla + плагин VirtueMart

Joomla — это полноценная CMS система, которая имеет понятную для новичка панель администрирования. Joomla имеет большие возможности, такие как: менеджеры архивов, поиска, почтовых рассылок, шаблонов и редактор текста. Обилие различных форумов, сайтов и документации на русском языке позволяют получить необходимую информацию без особых усилий. Имеется множество модулей, плагинов и расширений для Joomla, позволяющие делать на ее основе различные сайты: от простого сайта-визитки до интернет-магазина.

VirtueMart — один из популярных плагинов системы Joomla, позволяющий организовать на сайте интернет-магазин. Он легко устанавливается с помощью автоматического инсталлятора компонентов и модулей. VirtueMart ориентирован на создание небольших и средних интернет-магазинов. Это свободно распространяемый продукт, позволяющий любому желающему организовать свой онлайн-магазин. На рисунке 4 представлен интерфейс плагина VirtueMart в системе управления сайтом Joomla.

VirtueMart поддерживает неограниченное количество продукции и категорий, неограниченное количество валют для одного продукта, назначение продукции множеству категорий, возможность продавать продукцию, предназначенную для скачивания, а также предоставляет возможность отключить функцию продажи и использовать VirtueMart в режиме каталога. В VirtueMart реализована система скидок, купонов, а также большой выбор систем оплаты и способов доставки.

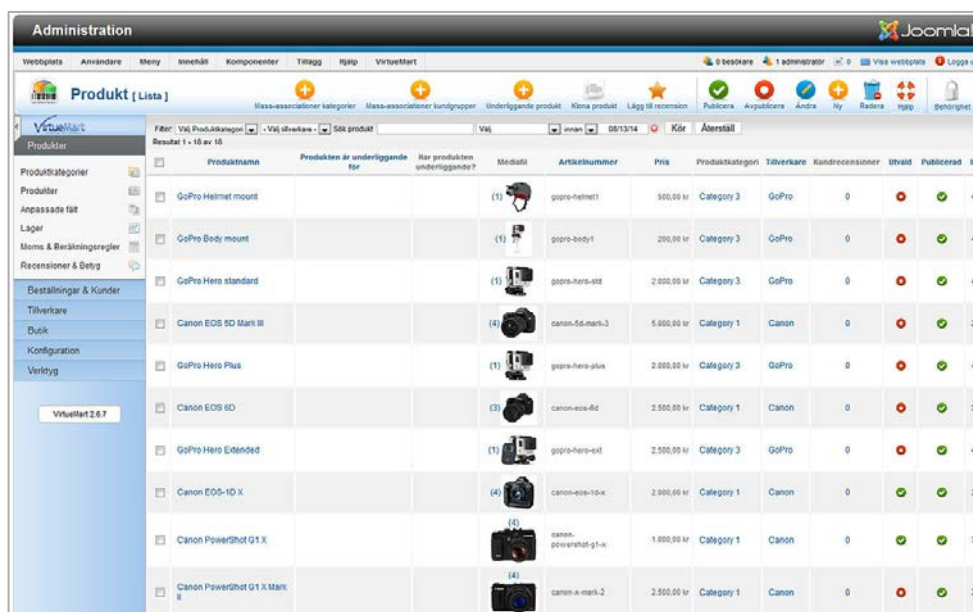


Рисунок 4 — Интерфейс плагина VirtueMart в системе управления сайтом Joomla

Категории создаются с любой вложенностью. Можно показывать один и тот же товар в разных категориях, притом карточка товара будет доступна по двум разным адресам, и у дублей страниц будут автоматически созданы канонические ссылки (поэтому поисковики не будут считать их дублированным контентом).

Можно сделать товар доступным только для определенных групп покупателей. Также можно задавать свою цену товара для каждой группы. Можно добавить характеристики к товару. Можно сделать, чтобы они влияли на цену товара. Есть цены со скидкой, купоны на скидку. К товарам можно добавлять отзывы [11].

Заказы поступают в панель администратора. Уведомления приходят покупателю и администратору магазина. Функционал личного кабинета стандартный.

Анализируя плагин VirtueMart для системы Joomla, можно сказать, что этот плагин хорошо дополняет систему, делая из нее полноценный интернет-магазин со всем необходимым функционалом. Автоматическое создание канонических ссылок, возможность прописывать meta-теги делает интернет-магазин более дружелюбным к поисковым системам.

1.4 Обоснование выбора систем

Подобрать систему, предоставляющую все необходимые функции невозможно. В связи с чем был проведен анализ всех преимуществ и недостатков различных CMS, и выбрана система, наиболее подходящая требованиям, которым должен удовлетворять интернет-магазин.

Требования к интернет-магазину:

1. Понятный и легко усваиваемый интерфейс.
2. Интерфейс должен быть модульным.
3. Неограниченная глубина вложенности каталогов.
4. Сортировка продукции по категориям.
5. Наличие контактной информации с возможностью обратной связи.
6. Наличие функции поиска.
7. Панель администратора с широким набором функций управления магазином.
8. Удобная и быстрая система заказа.
9. Обеспечивать конфиденциальность информации о покупателях и заказах.

В данном проекте выбрана система управления OpenCart, которая максимально удовлетворяет требованиям к интернет-магазину.

Рассмотрим эту бесплатную и современную CMS. Система OpenCart проста в использовании, многофункциональна, дружелюбна к поисковым системам и имеет простой и привлекательный интерфейс.

Преимущества системы OpenCart среди аналогов:

- бесплатная;
- быстрая и простая установка;
- небольшой вес CMS;
- высокая производительность;
- гибкость настроек из панели администратора;
- имеет огромное сообщество, как русскоязычное, так и зарубежное;

- большой выбор модулей и шаблонов;
- простота настройки и администрирования магазина.

Недостатки OpenCart при базовой установке:

- большое количество шагов при оформлении заказа;
- скудный набор фильтров для товаров;
- нет встроенных функций для ведения блога или новостей;
- нет возможности вывести товары всех категорий на одной странице;
- нет модуля обратного звонка.

Исходя из условий выбранной системы управления, был определен стек технологий разработки, необходимый для работы в CMS OpenCart: HyperText Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, Hypertext Preprocessor (PHP), MySQL.

CSS же является примитивным языком. В нём невозможно создать функцию, переиспользовать определение или использовать наследование. Для больших проектов или сложных систем, поддержка кода на CSS становится большой проблемой. Однако, web-технологии быстро развиваются, новые спецификации вводятся как в HTML, так и в CSS. Web-браузеры применяют эти спецификации, но оставляют свои специальные вендор-префиксы. Приходится добавлять всевозможные префиксы разных версий браузеров, чтобы достичь одинакового результата [19].

Чтобы код на CSS получался лучше, программисты чаще всего используют различные подходы, например, разделение определений стилей в небольшие файлы и их импортирование в один большой главный файл. Такой подход помогал разделять стили по компонентам, но не решал проблему с повторениями кода и не облегчал его поддержку. Другим подходом являлись попытки внедрения объектно-ориентированного программирования в CSS. В этом случае, к элементу применялись два или более определений классов. Каждый класс добавлял один тип стиля к этому элементу. Создание множе-

ства классов увеличивало возможность наследования кода, но уменьшало возможность эффективной поддержки кода [25].

Таким образом, в первую очередь, для комфортного программирования стилей для web-клиента, разработчику необходим CSS-препроцессор. CSS-препроцессоры расширяют использование CSS переменными, операторами, интерполяциями, функциями, «примесями» (mixins) и остальными полезными средствами. Наиболее известными CSS-препроцессорами являются SASS, LESS и Stylus. Как и любой язык программирования, препроцессоры имеют различные синтаксисы, однако, они довольно похожи друг на друга. Все препроцессоры поддерживают «классическое» программирование на CSS и их синтаксисы в целом схожи с CSS.

Препроцессоры, в свою очередь, помогают достичь написания масштабируемого и легко поддерживаемого кода в CSS. Используя препроцессор, программист может легким образом увеличить свою продуктивность и уменьшить объём кода в своём проекте.

Рассмотрим, чем отличаются синтаксисы CSS-препроцессоров на примере примесей (рисунок 5).



```
Sass
@mixin bordered($width) {
  border: $width solid #ddd;

  &:hover {
    border-color: #999;
  }
}

h1 {
  @include bordered(5px);
}

Less
.bordered (@width) {
  border: @width solid #ddd;

  &:hover {
    border-color: #999;
  }
}

h1 {
  .bordered(5px);
}

Stylus
bordered(w)
  border: n solid #ddd
  &:hover
    border-color: #999

h1
  bordered(5px)

Css
h1 { border: 5px solid #ddd; }
h1:hover { border-color: #999; }
```

Рисунок 5 — Описание примесей в различных препроцессорах

Примеси — это набор условий, который компилируется в соответствии с некоторыми параметрами или статическими правилами. С помощью них можно создавать что-то наподобие функций и переиспользовать правила стилей в других местах в коде. Как видно из рисунка 5, наиболее привычным

синтаксисом, похожим на CSS, является синтаксис SASS и LESS, однако в SASS более конкретное, подробное описание использования примесей (@mixin, @include), чем в LESS (символ «.»), что дублирует селектор класса, который тоже начинается с символа «.»), поэтому при разработке будет использоваться CSS-препроцессор SASS [22].

В качестве дополнительных инструментов использовались такие средства как:

1. Photoshop CC — многофункциональный графический редактор, разработанный и распространяемый фирмой Adobe Systems. В основном работает с растровыми изображениями.

2. PhpStorm — интегрированная среда разработки, предоставляющая собой интеллектуальный и богатый возможностями редактор кода с расширенной конфигурацией форматирования кода, проверкой на наличие ошибок на лету и умным автодополнением.

3. Webpack — это инструмент, позволяющий автоматизировать рутинные задачи web-разработки, собирая и компилируя JavaScript, Sass и т.д. модули в отдельные статичные ассеты.

4. GIT — распределённая система контроля версий, которая даёт возможность разработчикам отслеживать изменения в файлах и работать совместно с другими разработчиками.

5. Node.js — программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения, добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой Application Programming Interface (API), написанный на C++ и подключать другие внешние библиотеки, написанные на разных языках.

6. OpenServer — это портативная серверная платформа и программная среда, созданная специально для web-разработчиков.

7. NPM — это менеджер пакетов, используемый Node.js-приложениями, управляющий библиотеками для web-разработки.

8. Bitbucket — это сервис, централизованно хранящий репозитории (файлов, доступных для дальнейшего распространения по сети) Git (с октября 2011 года) и Mercurial (с момента запуска).

9. GitKraken — кроссплатформенный визуальный клиент системы управления версиями Git.

1.5 Общий алгоритм реализации интернет-магазина

Первый этап — составление технического задания, полностью описывающего поставленные перед разработчиком задачи и отображающего конечную цель проекта в понимании заказчика.

Одним из важнейших условий подготовки качественного технического задания на сайт является тесный контакт с заказчиком. В каждом случае необходимо составить техническое задание для сайта индивидуально, с учетом особенностей конкретного ресурса, а также компании, которой он будет принадлежать. При этом очень важен системный подход, доскональность — техническое задание на сайт должно максимально четко и подробно описывать будущий сайт [1].

Грамотно составленное техническое задание на сайт должно содержать следующие моменты:

- требования к технической части — это основа, базис, то, от чего будут отталкиваться разработчики. Эта составляющая технического задания описывает все ключевые нюансы разработки будущего проекта: от выбора CMS до внедрения конкретных функций;
- маркетинговая часть — от нее зависит, будет ли ресурс коммерчески успешным. Маркетинговые требования в техническом задании на сайт описываются с учетом сферы деятельности компании, ее преимуществ, ситуации на рынке, уровня и характера конкуренции, особенностей целевой аудитории и т. п.;

- дизайн, юзабилити, контент — эта часть технического задания (ТЗ) на разработку сайта направлена на то, чтобы сделать сайт максимально удобным, интересным и полезным для пользователя. В этой части технического задания требуется учесть все пожелания заказчика, и так же зафиксировать все предложения по дизайну.

Второй этап — стадия проектирования.

Проектирование — ключевой этап создания интернет-сайта, отвечающий на следующие вопросы:

1. Каковы наши цели — зачем мы делаем сайт? Как мы реализуем поставленные цели?
2. Как сайт будет выглядеть и работать?

Третий этап — непосредственная разработка и работа с программным кодом:

- вёрстка всех страниц сайта средствами CSS и HTML;
- подключение плагинов, библиотек и шрифтов;
- установка и настройка CMS OpenCart;
- доработка существующего функционала, изменение поведения под требования технического задания;
- разработка собственных модулей;
- написание скриптов на JavaScript.

Этап разработки — самый долгий и трудоёмкий. Он требует больших трудозатрат относительно остальных этапов, и объем этих работ напрямую зависит от качества выполнения этапов составления ТЗ и проектирования [21].

Четвёртый этап — тестирование продукта на корректность отображения в современных браузерах, таких как Google Chrome, Safari, Microsoft Edge, Internet Explorer 11, Mozilla Firefox.

На данном шаге важно найти и исправить все недочёты, связанные с неправильным отображением сайта, когда браузеры по-разному интерпретируют комбинации различных свойств стилей.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Характеристика потенциальной аудитории проекта

Целевая аудитория интернет-магазина — это люди, выполняющие ремонт в своем доме или квартире. А также люди, которые совсем недавно приобрели жилье. Сайт предназначен для жителей Санкт-Петербурга и Москвы со средним доходом.

Для посетителей сайта не существует возрастных ограничений по использованию товара, представленного на сайте, однако, учитывая категорию изделий, оптимальный возраст покупателей начинается от 25 лет.

2.2 Постановка задачи проекта

2.2.1 Актуальность проекта

Актуальность создания интернет-магазина заключается в удобстве поиска, подбора и просмотра товаров через Интернет. На сайте можно найти всю необходимую информацию о товаре, о его стоимости и комплектации, об адресах магазинов, о реквизитах, о дополнительных услугах и об истории компании.

Разрабатываемый проект обладает удобным фильтром по цене, цвету, по производителям, категориям и странам, производящим продукцию.

Несомненным преимуществом такого вида продаж является то, что жители разных городов могут ознакомиться с товарами компании, узнать характеристики и цену интересующего их товара. Это особенно важно, поскольку продаваемый товар не представляется возможным выставить для демонстрации как в обычном магазине из-за определенных особенностей.

2.2.2 Цель и назначение проекта

Создание интернет-магазина обусловлено его обоснованностью и актуальностью и включает в себя следующие цели:

1. Создание интернет-магазина, способствующего покупке товаров со стороны пользователей, продаже товаров со стороны компании.
2. Обеспечение круглосуточного доступа к ресурсу для информирования пользователей напрямую.
3. Ресурс будет способствовать ускорению и экономии времени в процессе покупки, продаже, поиске товаров.
4. Увеличение охвата аудитории компании.
5. Способствование оказанию помощи в поиске товаров по заданным характеристикам.

2.2.3 Функционал проекта, интерфейс проекта, эргономические и системные требования и другие требования к проекту

Качество разрабатываемого продукта определяется его соответствием требованиям, которые были заложены на стадии проектирования системы [1]. Требования к проекту разделяются на нефункциональные и функциональные.

Нефункциональные требования представляют из себя характеристики, параметры, фотографии продуктов, важных для конечного пользователя системы.

Функциональные требования диктуют функциональность системы, которую должен реализовать разработчик для выполнения пользователями заданных в рамках данного проекта бизнес-процессов.

Функциональность проекта можно описать такой концепцией: размещение товара — связь с менеджерами компании — продажа. Были выделены следующие функции:

1. Возможность добавления товаров, статей, услуг.

2. Просмотр товаров с их характеристиками и фотографиями.
3. Возможность фильтрации по заданным параметрам.
4. Возможность связаться с продавцом и оформить заказ.

Первоочередная функция интернет-магазина — оформление заказа. Это позволит идентифицировать пользователя и поспособствует установлению связи с покупателем [5]. На этапе оформления заказа предполагается внесение пользователем базовой информации о себе, позволяющей связаться с ним.

Добавление товаров позволяет поместить изображения, текстовое описание и технические характеристики. Просмотр товаров, пожалуй, самая основная функция интернет-магазина. Однако помимо просмотра определенное удобство составит и поиск товаров по фильтру.

Возможность связи с менеджерами компании заключается в возможности оформить заказ, просмотра номера телефона на странице контактов или в нижней части сайта, а также в специальной форме обратной связи.

Заказчиком были поставлены следующие задачи:

1. Разработка интернет-магазина для продажи товаров для ремонта.
2. Разработка отдельно раздела с услугами магазина.
3. Возможность фильтрации товаров по различным параметрам.
4. Возможность сравнения товаров по характеристикам.
5. Разработка страницы с акционными позициями товаров.
6. Возможность совершать поиск товаров на сайте по названию и описанию.
7. Добавление информационных страниц о компании, о контактных данных магазина, о подробностях доставки и оплаты.
8. Возможность оформления заказа с получением контактных данных покупателя.
9. Соблюдение 152-ФЗ Федерального закона «О персональных данных».

Заказчиком были поставлены сроки выполнения работ: готовность продукта к маю 2019 года.

Для посетителей сайта заказчик выделил следующий функционал:

1. Функция «Заказать звонок». Служит для быстрой связи с пользователем для уточнения, его интересующей информации.
2. Функция оформления заказа — одна из самых востребованных функций на сайте интернет-магазина, данные (заполняемые пользователем) из которой отправляются на почту владельцу магазина и в административную панель.
3. Корзина пользователя — это индивидуальная страница пользователя, в которой временно хранятся все выбранные товары, и считается общая сумма покупки данных товаров.
4. Добавление товара в сравнение — это страница, на которой пользователь может сравнивать товары по их отличительным характеристикам.
5. Фильтрация и поиск товаров по различным категориям, ценам, характеристикам и ключевым словам.

Необходимо создать удобную, простую форму заказа товара, сделать проверку на корректность введенной в поля оформления заказа информации.

Общие требования к сервису: дизайн при просмотре с мобильных устройств должен выглядеть в точности так же, как и на персональном компьютере.

2.2.4 Входные данные

К данному проекту заказчиком были предоставлены дизайн-макеты разрабатываемого интернет-магазина в количестве 15 штук в формате *.psd.

А именно для таких страниц как:

- главная страница в трёх вариантах состояния;
- страница товара в двух разных вариантах;
- категория товара со списком товаров в четырёх разных состояниях;

- страница контактных данных магазина;
- страница оформления заказа;
- страница с таблицей цен на определенные услуги;
- страница с информацией о компании;
- страница с подробным описанием услуги;
- сравнение товаров.

Для примера дизайна, рассмотрим дизайн-макет страницы товара, представленного на рисунке 6.

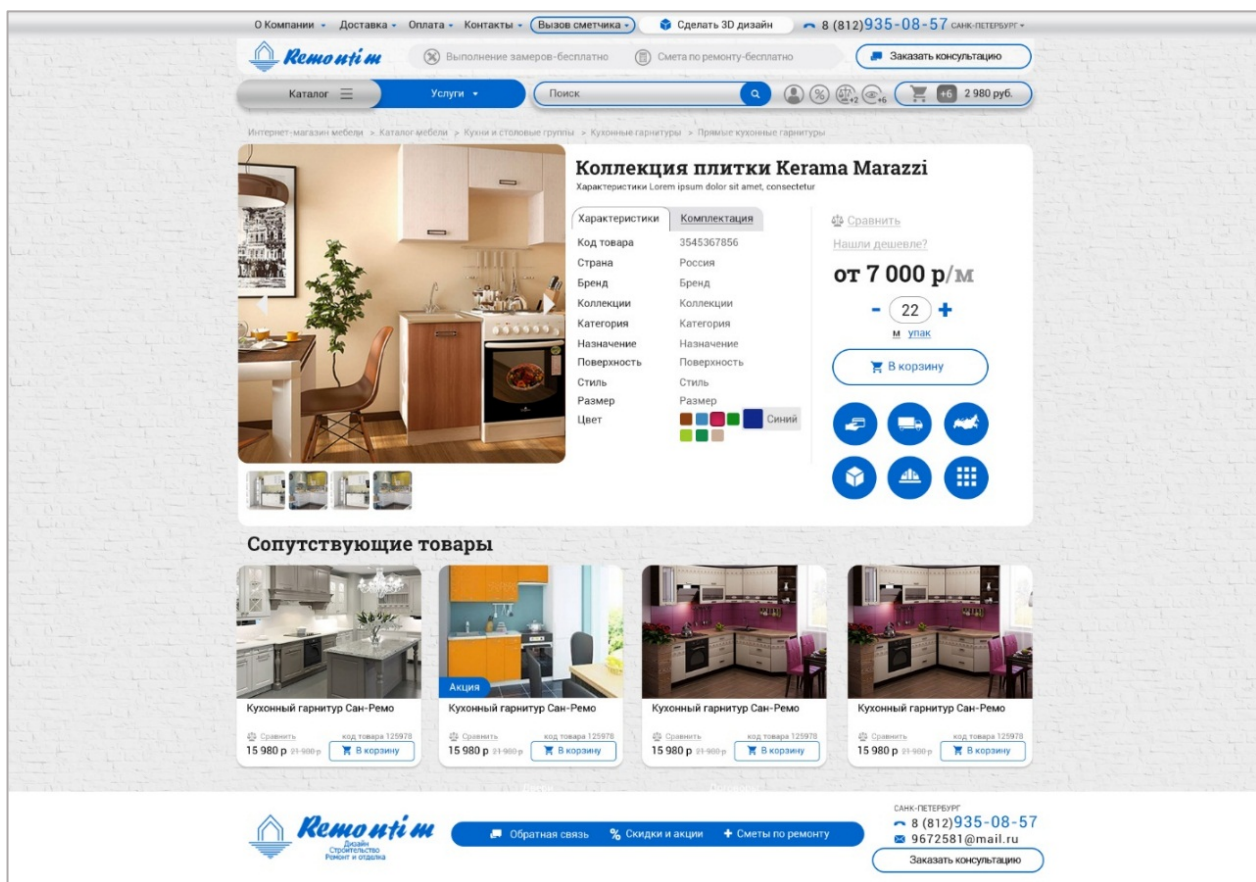


Рисунок 6 — Дизайн-макет страницы товара

Дополнительно к дизайну были установлены специальные шрифты и символы, которые можно использовать в графическом редакторе Adobe Photoshop.

На рисунке 7 представлены цвета, выбранные заказчиком и используемые в дизайн-макетах.



Рисунок 7 — Цвета сайта

На рисунке 8 продемонстрирован логотип компании «Ремонтим». Именно под этим логотипом будет компания будет представлять себя в сети Интернет.



Рисунок 8 — Логотип компании «Ремонтим»

Рассмотрим структуру главной страницы:

- шапка сайта (header):
 - логотип компании;
 - навигационное меню по информационным страницам сайта;
 - меню каталога товаров и услуг;
 - форма поиска по сайту;
 - кнопка для заказа обратного звонка;
 - корзина покупок;
 - телефон для связи;
 - кнопки персональных пользовательских функций;
- keyvisual — первый визуальный элемент, попадающий в глаза посетителю интернет-магазина;
- блок с главными категориями;
- тексты о компании;
- подвал сайта (footer):
 - логотип компании;
 - кнопки для перехода на страницы акций, сравнения товаров и для вызова формы обратной связи;
 - контактные данные.

2.2.5 Характеристики оборудования для реализации проекта

Для реализации проекта использовался портативный компьютер (ноутбук) ASUS N550J. В таблице 2 приведена характеристика используемого оборудования.

Таблица 2 — Характеристика используемого оборудования.

Операционная система	Windows 10
Тип системы	64-разрядная
Процессор	Core i7 4700HQ
Частота процессора	2.4 ГГц
Объем оперативной памяти	12 Гб
Тип оперативной памяти	DDR3
Объем постоянной памяти	256 Гб
Графический процессор	GeForce GT 750M
Объем графической памяти	4 Гб

2.3 Жизненный цикл проекта

2.3.1 Этап разработки элементов дизайна проекта

В ходе работы над сайтом, дизайнером от заказчика было создано 15 макетов, используемых для дальнейшей разработки проекта: верхняя и нижняя части сайта, главная страница, категории, карточка товара, оформление заказа, страница услуг, контакты, о компании, сравнение товаров.

Самым первым этапом стал разбор макетов на составные части: на иконки, шрифты и изображения.

На рисунке 9 представлен макет верхней части сайта, в которой содержится основной навигационный функционал.

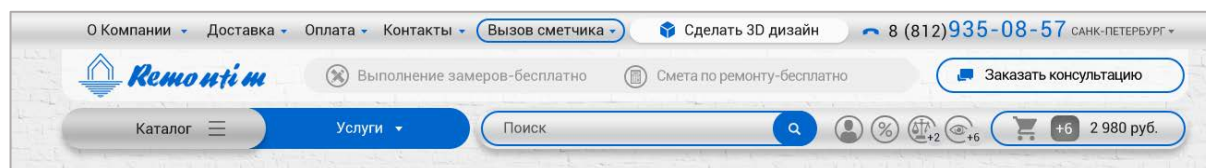


Рисунок 9 — Макет верхней части сайта

Верхняя часть сайта отвечает за размещение элементов навигации по интернет-магазину. В главном меню содержатся ссылки для перехода на главную страницу, категории товаров, услуги, информационные статьи о компании, а также доступны функции поиска по сайту, ссылки для перехода в корзину и в сравнение товаров.

Кнопка «Заказать консультацию» в панели навигации вызывает всплывающую форму обратной связи с полями для ввода номера телефона и имени, и кнопку для отправки сообщения администратору сайта.

Категории товаров на рисунке 10 представлены в виде двух уровней вложенности. Категории первого уровня при наведении открывают список второго уровня. Так же у каждой категории первого уровня имеется иконка, отображающая суть категории.

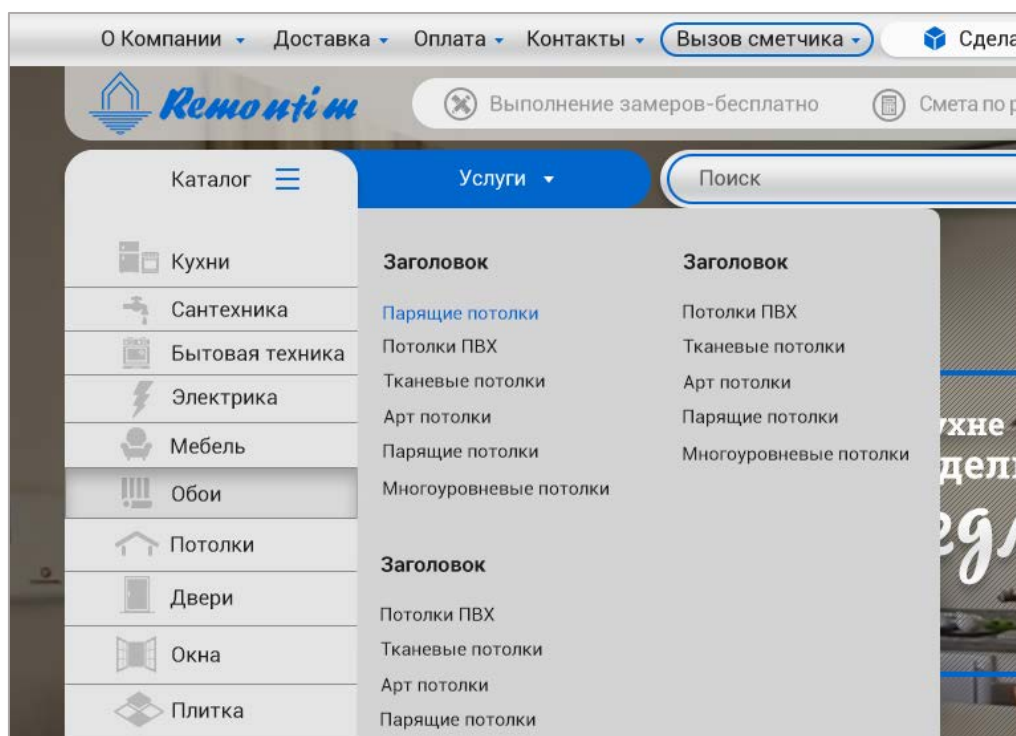


Рисунок 10 — Макет развернутого меню категорий в навигационной панели

На рисунке 11 представлен макет нижней части сайта, в которой указаны: номер телефона, адрес электронной почты, город, ссылки на страницы и форму связи, логотип.

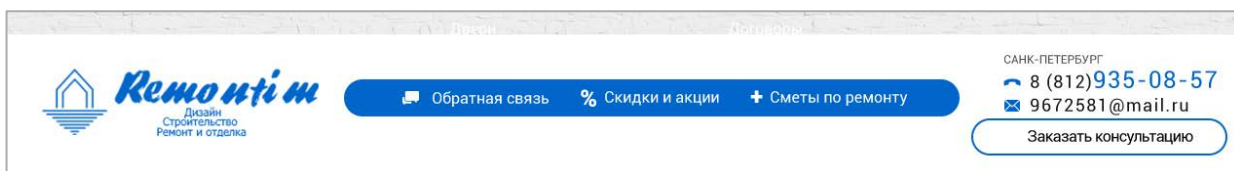


Рисунок 11 — Макет нижней части сайта

При переходе в любую из категорий, пользователю открывается список товаров интернет-магазина, где пользователь может сортировать товары по популярности, по алфавиту, по цене, как показано на рисунке 12. В левой части сайта пользователю представляется возможность применить фильтр к списку товаров. Параметры, по которым пользователь может фильтровать товары, устанавливаются администратором сайта.

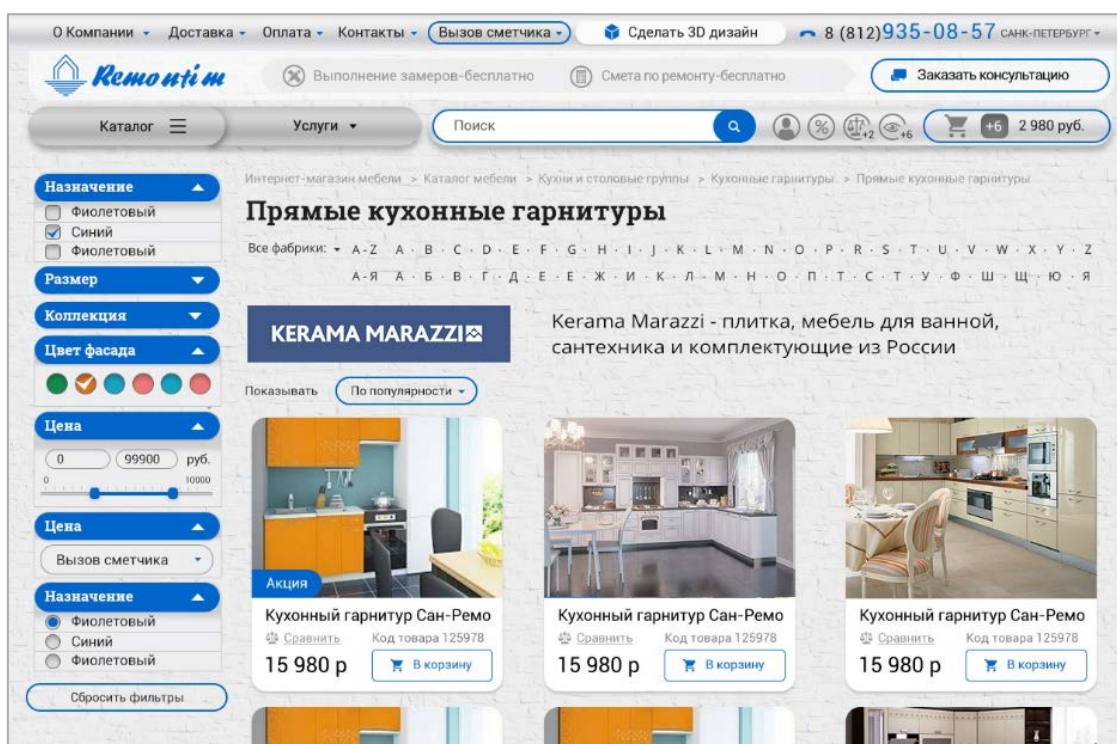


Рисунок 12 — Макет страницы категории со списком товаров и фильтрацией

Также из страницы категории пользователь может просмотреть изображения товара, добавить товар в корзину или в сравнение.

При переходе на страницу товара (представленную на рисунке 13), пользователю открывается возможность ознакомиться с подробной информацией о выбранном товаре, комплектацией, сопутствующими товарами и всеми изображениями.

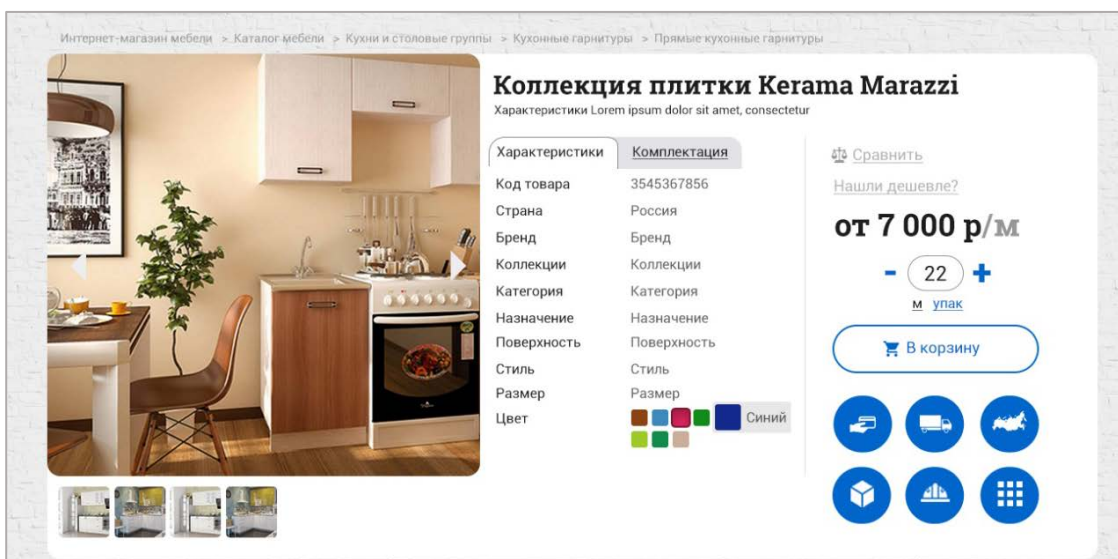


Рисунок 13 — Макет страницы товара

Есть очень хороший способ увеличить эффективность интернет-магазина — предложить покупателю дополняющие товары к основному. Если человек купил кухню, ему можно предложить дополнительную кухонную мебель и услуги по её сборке. Пакетная продажа сразу нескольких товаров и услуг.

Модуль сопутствующих товаров представлен на рисунке 14.

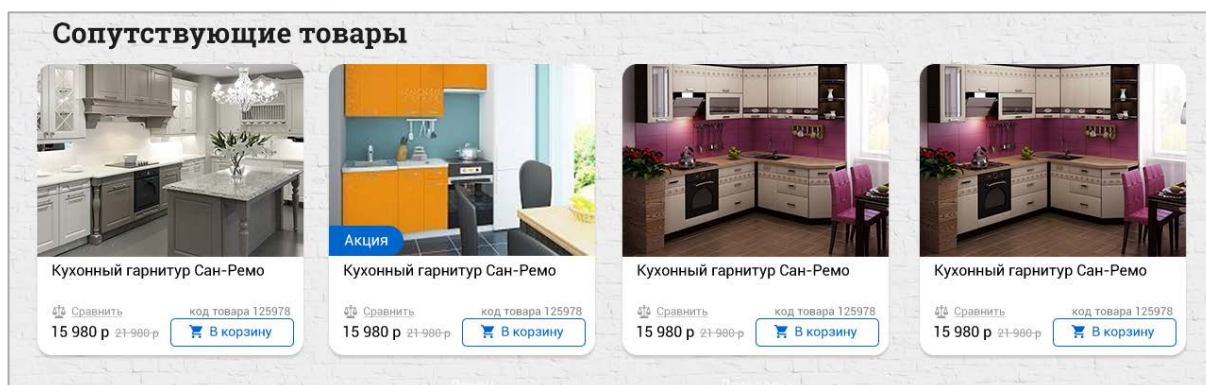


Рисунок 14 — Макет модуля сопутствующих товаров

Дополняющих товаров может быть довольно много, все их показать на странице товара будет невозможно. Для этого у администратора должна быть возможность выбора товара, который необходимо отображать.

Со страницы товара пользователь может добавить товар как в корзину покупок, так и в сравнение с другим товаром.

При нажатии на кнопку «В корзину», пользователь может перейти к странице оформления заказа, которая представлена на рисунке 15.

Главная > Корзина

Корзина

	Убрать
Кухонный гарнитур Сан-Ремо	Донор
Тип	15978543
Код товара	
- 22 + Шт	7 000 р

	Убрать
Кухонный гарнитур Сан-Ремо	Донор
Тип	15978543
Код товара	
- 22 + Шт	7 000 р

Продолжить покупки Очистить корзину

Контактная информация

Контактное лицо * Иван Иванович Иванов

Телефон * +7 987 654 32 10

Email адрес * Ivan@mail.ru

Доставка
 Самовывоз

Город Санкт-Петербург

Адрес Солдатова 24

Дата доставки ДД/ММ/ГГ

Время доставки ЧЧ/ММ

Дополнительные услуги

Бежевый
 Белый

Оплата

Способы оплаты Карта

Итого 14 000 р **Заказать**

Рисунок 15 — Макет страницы оформления заказа

На странице оформления заказа пользователь видит информацию о добавленных в корзину товарах и их количестве, а также может изменить количество заказываемых товаров.

В правой части страницы располагается форма для заполнения контактных данных покупателя и дополнительной информации.

При нажатии на кнопку «Заказать», администратор или менеджер магазина получает уведомление о новом заказе и связывается с клиентом по указанным им контактными данными.

Очень важная функция, которой пользуются многие покупатели, является сравнение товаров. Практически каждый товар имеет аналоги, который очень сложно различить между собой без подробного анализа. Для помощи покупателям в решении совершить ту или иную покупку и была придумана функция сравнения товаров, представленного на рисунке 16.

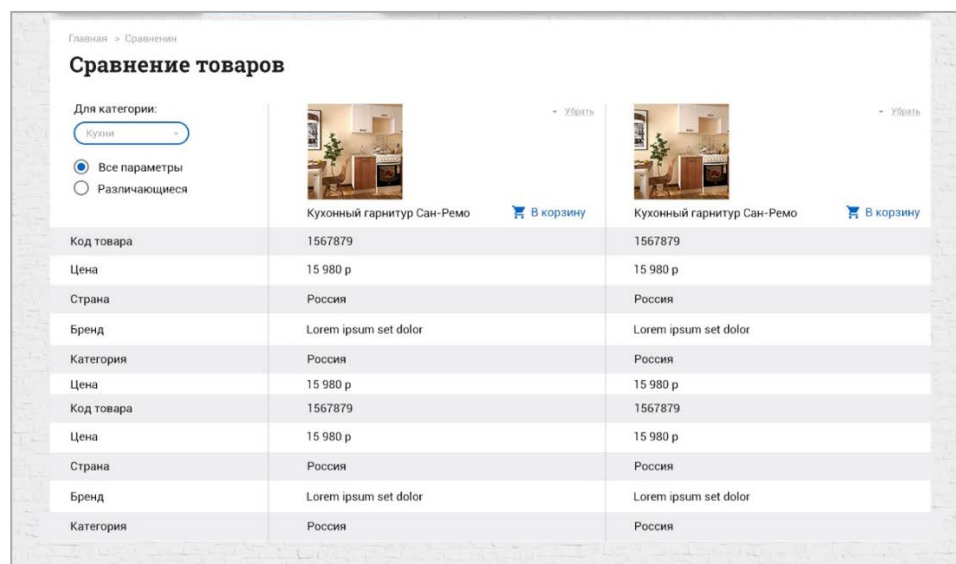


Рисунок 16 — Макет модуля сравнения товаров

Страница сравнения организована в виде таблицы, где по вертикали располагаются товары, а по горизонтали их характеристики. Параметров и характеристик у некоторых товаров может быть много, поэтому сделаны переключатели «Все параметры» и «Различающиеся». В интерфейсе все горизонтальные линии выделены чередующимся ненавязчивым цветом, чтобы облегчить визуальное восприятие таблицы. Фотография товара маленькая, покупатель прежде всего будет сравнивать характеристики товаров в таблице. Возле фотографии располагаются кнопки «В корзину» и «Убрать», которые пользователь использует при принятии решения о покупке.

После добавления товара в корзину, пользователь получает соответствующее сообщение об этом, представленное на рисунке 17.

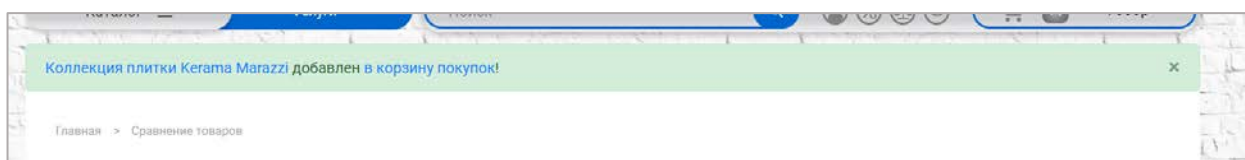


Рисунок 17 — Сообщение об успешном добавлении товара в корзину покупок

Через сообщение об успешном добавлении товара можно перейти в корзину покупок и на страницу с информацией о товаре.

На странице «Контакты», представленной на рисунке 18 размещается максимально возможное количество контактной информации. Это нужно для

того, чтобы человек мог связаться с магазином удобным ему способом, получить консультацию, помощь в выборе товара, а также карта проезда. Контактная информация должна редактироваться администратором сайта через панель управления.

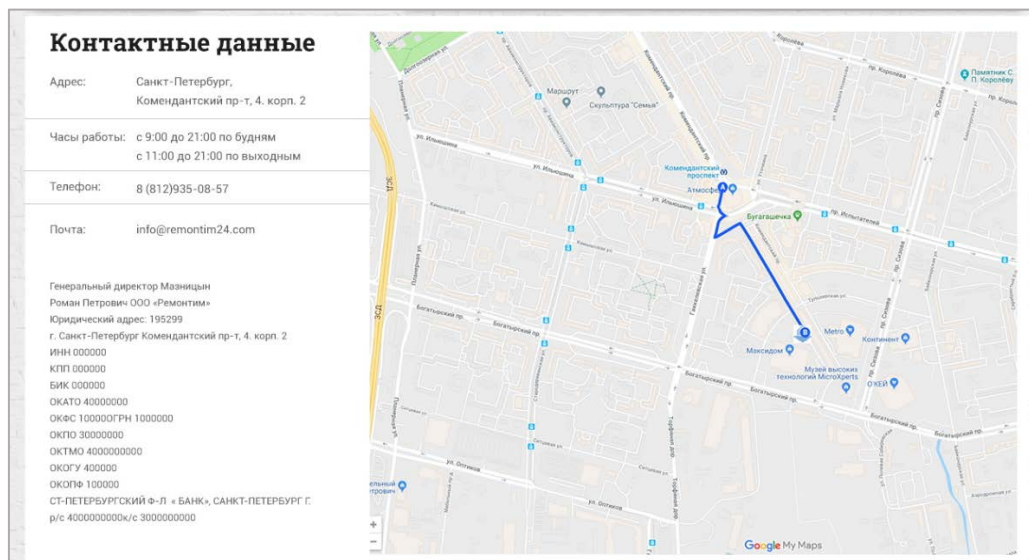


Рисунок 18 — Дизайн страницы с контактной информацией

Помимо товаров, сайт предлагает посетителям услуги по комплексному ремонту квартир, макет страницы с которыми представлен на рисунке 19.

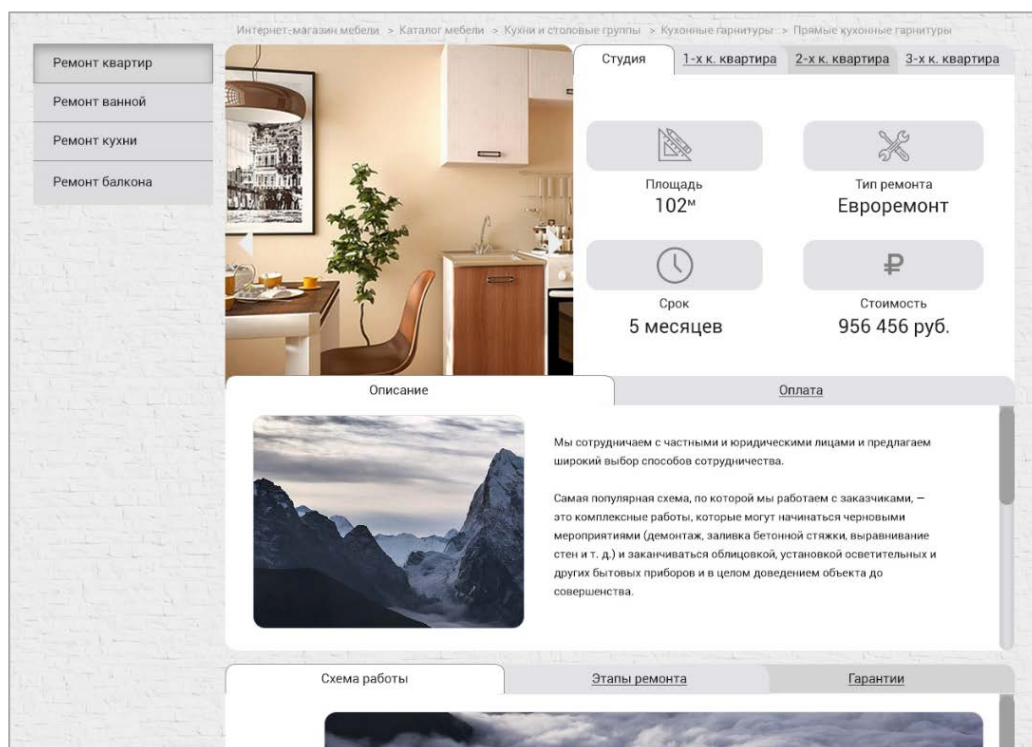


Рисунок 19 — Дизайн страницы комплексных услуг

Второй тип страниц с услугами включает в себя информацию об отдельных видах услуг, представленных в виде таблицы расценок. На рисунке 20 продемонстрирована основная информация выбранной услуги.

Стоимость и сроки дизайн-проекта		
<p>Стоимость выполнения дизайн-проекта интерьера составляет:</p> <ul style="list-style-type: none"> - Жилые интерьеры — от 3000 рублей за 1 м2. - Коммерческие интерьеры — от 2000 рублей за 1 м2. <p>Срок разработки дизайн-проекта — 30 - 60 дней</p>		
Наименование работ	Ед. изм.	Расценка за ед.
Демонтаж перегородок	кв. м.	340
Демонтаж балконной двери	шт.	1 200
Демонтаж перегородок	кв. м.	340
Демонтаж балконной двери	шт.	1 200
Демонтаж балконной двери	шт.	1 200
Демонтаж перегородок	кв. м.	340

Стоимость дизайна интерьера

- Рабочий дизайн-проект
- Планировочное решение
- Рабочий дизайн-проект
- Планировочное решение

Стоимость ремонта и отделки

- Общестроительные работы
- Монтажные работы
- Сантехнические работы
- Планировочное решение
- Монтажные работы
- Сантехнические работы

Стоимость согласования перепланировки

Стоимость приёмки квартиры

Стоимость технического надзора

Рисунок 20 — Дизайн страницы услуг

На странице «Услуги» в правой части экрана отображена навигация по остальным видам услуг.

2.3.2 Этап разработки программного кода проекта

Достаточно большим и объемным этапом в реализации проекта стало написание программного кода.

В качестве системы контроля версий была выбрана система Git, которая является распространенной среди web-разработчиков, а также имеет множество программ с графическим интерфейсом и функциональную командную консоль. В качестве сервера для системы контроля версий выбран Bitbucket, так как данный сервер дает возможность бесплатно хранить приватные репозитории кода (что является обязательным для коммерческих проектов), и позволяет работать в web-интерфейсе, продемонстрированном на рисунке 21.

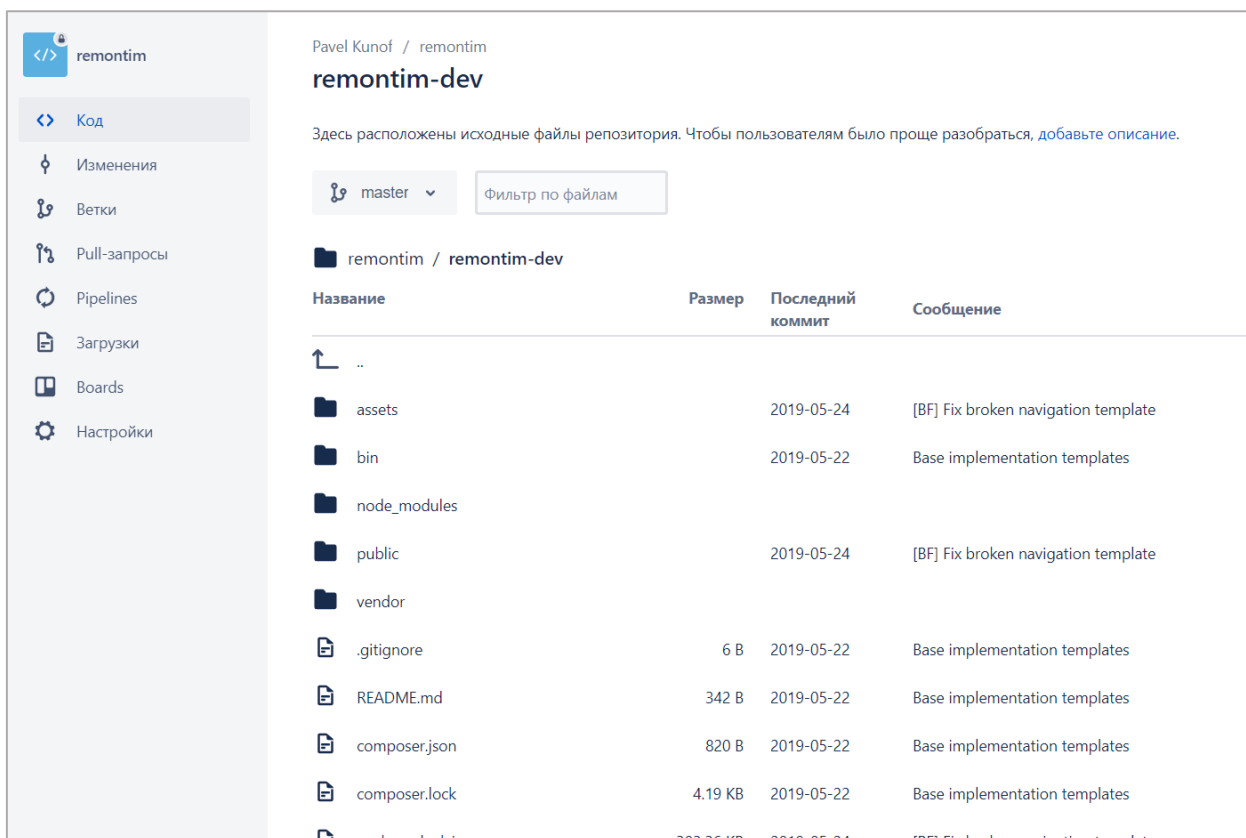


Рисунок 21 — Web-интерфейс сервиса Bitbucket

Первым этапом в разработке интернет-магазина является интеграция стека технологий. Для того, чтобы интегрировать большое количество зависимостей в проект, необходим пакетный менеджер. Без пакетного менеджера зависимости необходимо загружать с интернет-источников вручную. Самым популярным пакетным менеджером для JavaScript библиотек является «npm», который в свою очередь также является самым большим в мире программным реестром.

При использовании npm нет необходимости добавлять файлы зависимостей в сам проект, а достаточно добавить в корень проекта файл с наименованием «package.json» [27], который содержит соответствие названий JavaScript библиотек и их версий. Таким образом, в проекте всегда будут нужные версии пакетов, а загружать их можно перед запуском или сборкой проекта простой командой в консоли «npm install» (находясь в корне проекта), как показано на рисунке 22.

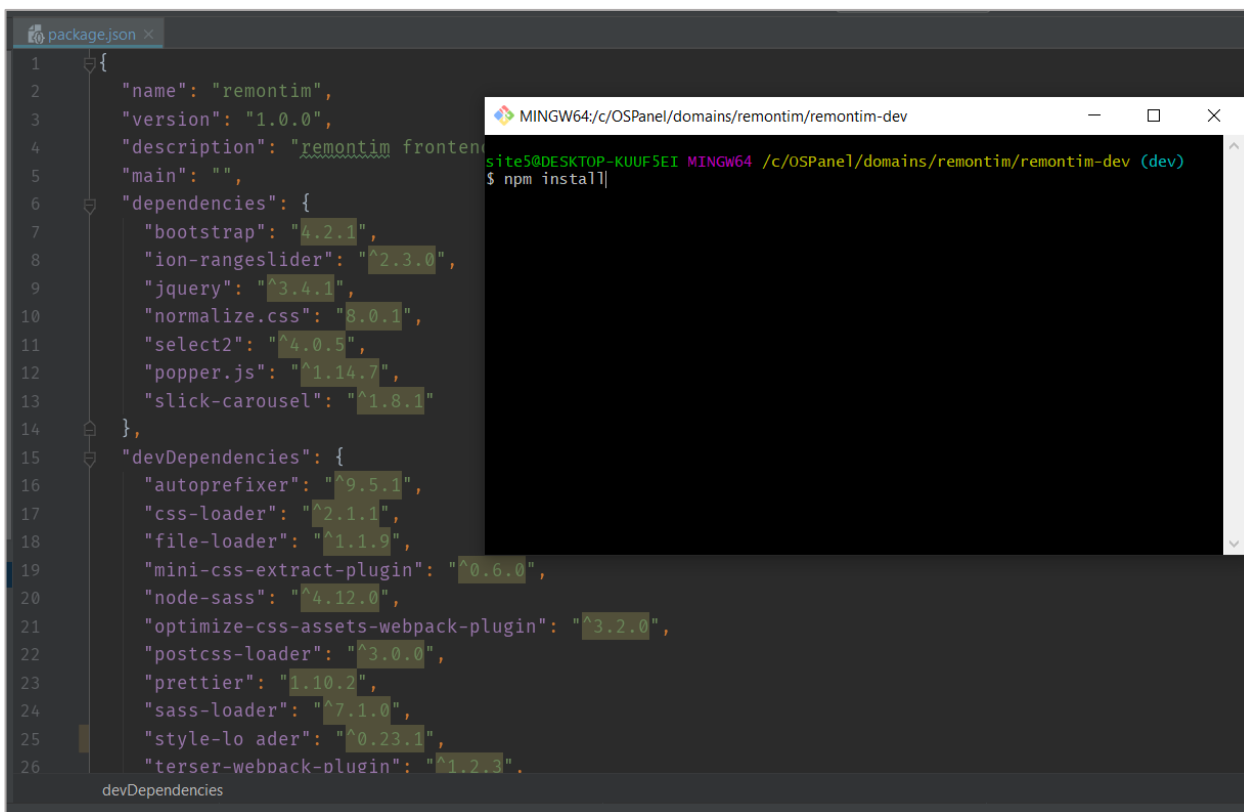


Рисунок 22 — Установка зависимостей проекта

Зависимости проекта, в свою очередь, могут также содержать свои зависимости, которые будут загружены при выполнении команды «`npm install`». Следовательно, кодовая база проекта будет содержать значительное количество кода, большая часть из которого не сжата, а некоторые части кода могут не использоваться в проекте и занимать лишнее место, что ведет к загрузке лишних данных в браузер при загрузке страницы и помещении этих данных в оперативную память, что может замедлить работу web-интерфейса.

Чтобы избежать данных проблем, необходимо отсеивать неиспользуемый код и сжимать его при сборке, однако наиболее эффективно делать это автоматически. Автоматически это можно сделать, если подходящим образом сконфигурировать сборку в `webpack`.

`Webpack` просматривает все файлы, распределенные по проекту, анализирует их, производит сборку файлов и сортирует их по соответствующим директориям. На рисунке 23 представлена схема, демонстрирующая работу сборщика. На ней видно, как сборщик просматривает все модули с зависимо-

стями и преобразует их в статические объекты, необходимые для работы приложения.

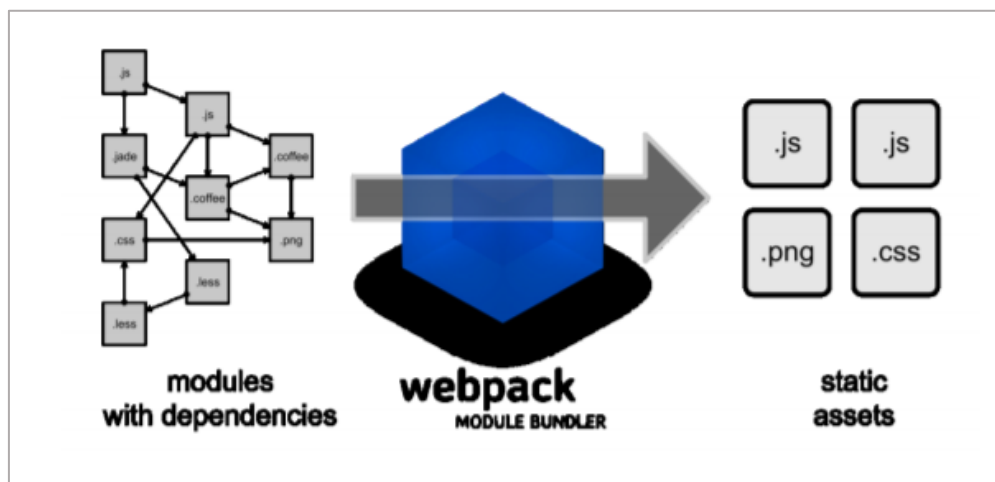


Рисунок 23 — Демонстрация работы webpack

Автоматизировать сборку проекта можно с помощью webpack [13], добавив в конфигурационный файл плагин «terser-webpack-plugin», который производит минификацию и обфускацию JavaScript-кода. Также, необходимо добавить плагин «webpack-watch-livereload-plugin», который отслеживает изменения в файлах и автоматически компилирует их с последующей перезагрузкой HTML страницы. Неиспользуемые файлы и библиотеки webpack не будет включать в результат сборки по умолчанию [31].

Для того чтобы автоматически использовать такие технологии, как CSS-препроцессоры, postCSS-обработка (autoprefixer) [26], современные стандарты ECMAScript, необходимо также добавить в конфигурационный файл системы сборки webpack так называемые «загрузчики» (loaders). Например, описанные в файле «package.json» зависимости css-loader, sass-loader, postcss-loader и autoprefixer позволяют разработчику использовать синтаксис CSS-препроцессора SASS и не писать специальные вендор-префиксы, т.к. добавление префиксов произойдет автоматически при сборке [30].

Основой работы для работы со сборщиком является файл конфигурации — достаточно мощный инструмент, но он может значительно отличаться

в разных проектах, а также может быть очень сложным, если идет разработка большого проекта.

На рисунке 24 приведен пример конфигурации сборщика проекта.

```
1  const path = require('path');
2  const MiniCssExtractPlugin = require("mini-css-extract-plugin");
3  const WatchLiveReloadPlugin = require('webpack-watch-livereload-plugin');
4  const autoprefixer = require('autoprefixer');
5  const OptimizeCssAssetsPlugin = require('optimize-css-assets-webpack-plugin');
6  const TerserPlugin = require('terser-webpack-plugin');
7  const isProd = process.env.NODE_ENV === 'production';
8
9  const devPlugins = [
10     new MiniCssExtractPlugin({
11         filename: 'css/[name].css'
12     }),
13     new WatchLiveReloadPlugin({
14         files: [
15             './public/{fonts,css,js,images}/*.{'css,png,svg,jp?eg,gif,js}'
16         ]
17     })
18 ];
19
20 const prodPlugins = [
21     new MiniCssExtractPlugin({
22         filename: 'css/[name].css'
23     }),
24     new OptimizeCssAssetsPlugin()
25 ];
26
27
```

Рисунок 24 — Пример конфигурации webpack

Набор зависимостей после загрузки в проект занимает 100 Мб на жёстком диске персонального компьютера (ПК) (рисунок 25).

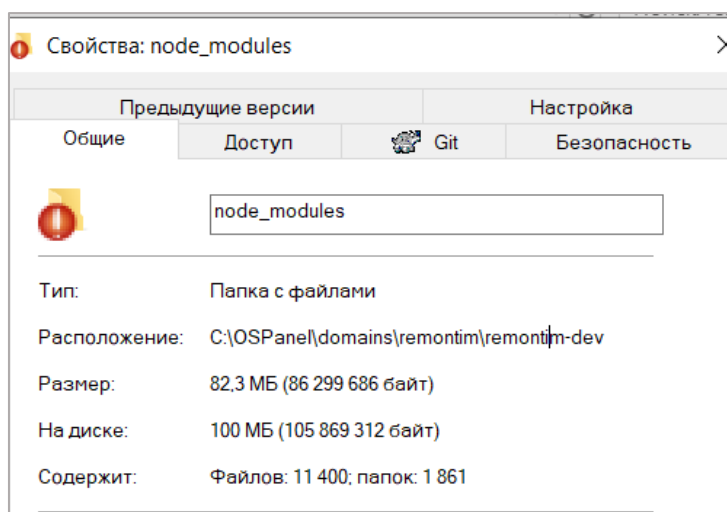


Рисунок 25 — Свойства папки «node_modules»

Размер всего остального кода в проекте составляет 1.1 Мб на жёстком диске персонального компьютера (рисунок 26).

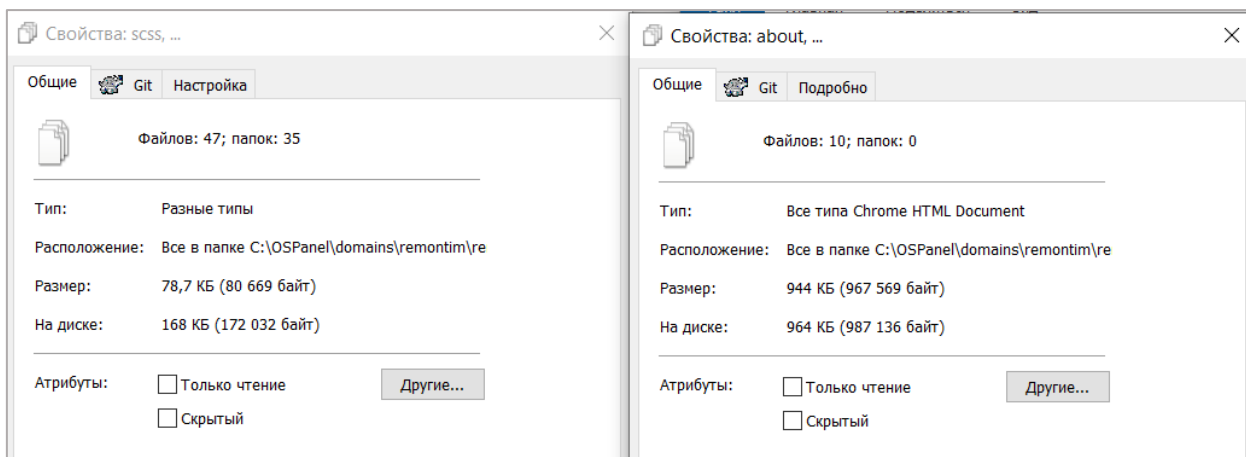


Рисунок 26 — Свойства исходных файлов проекта, кроме папки «node_modules»

Размер скомпилированного проекта со всеми подключенными в него библиотеками занимает 1.51 Мб, в соответствии с рисунком 27.

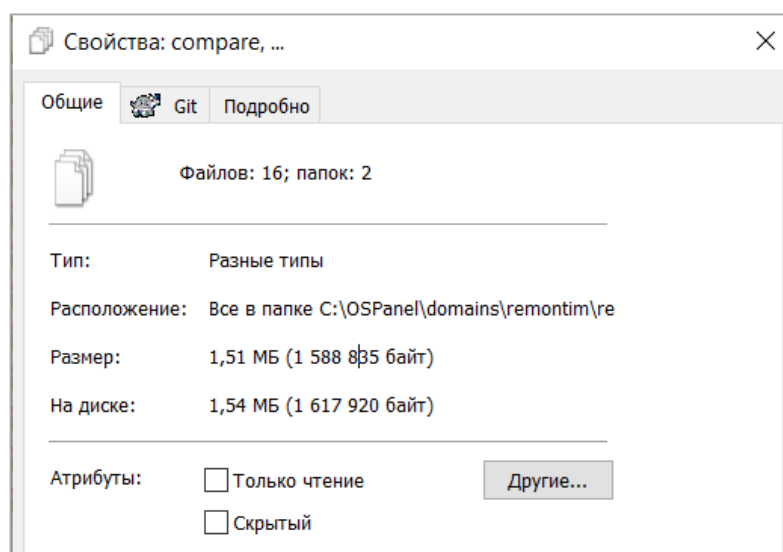


Рисунок 27 — Свойства скомпилированных файлов проекта

Для создания каскадных таблиц стилей на этапе анализа был выбран препроцессор SASS. На сегодняшний день SASS является самым мощным CSS-препроцессором, он обладает явными преимуществами в сравнении с другими препроцессорами.

Основными преимуществами SASS являются:

- совместимость с CSS;
- большое количество возможностей;

- многофункциональность библиотек.

SASS имеет очень важную и удобную для разработчика особенность: он позволяет сделать вложенную иерархию селекторов CSS. Эта особенность позволяет сделать код стилей более читабельным, а также избавит от написания лишнего кода. На рисунке 28 представлен код с использованием вложенности SASS и скомпилированный CSS код.

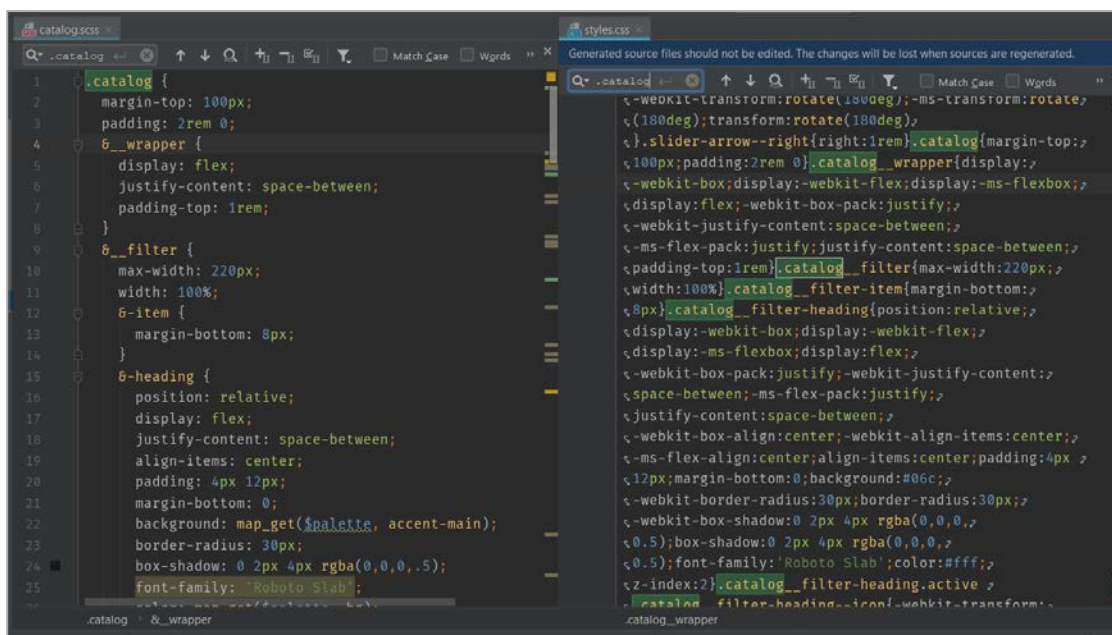


Рисунок 28 — Сравнение исходного и скомпилированного кода

Вложенность решает проблему повторного использования родительских селекторов, а также делает код более лаконичным и читабельным.

SASS поддерживает такой набор дополнительных расширений, как SaasScript. Этот набор позволяет использовать в SASS переменные, миксины, создавать собственные селекторы и имена для свойств. Самое распространенное применение переменных — кодировка цветов. Например, в большом проекте используется цвет с кодировкой #f8f8ff, и в какой-то момент разработчик решает изменить цвет. Переменные SASS позволяют изменить значение только в одном месте — там, где была объявлена переменная с цветом, - и цвет автоматически изменится во всем проекте [28].

SASS поддерживает все существующие @-правила CSS, остальные правила, которые поддерживает SASS, называются директивами. В разраба-

тываемом приложении используются такие директивы, как `@import` и `@media` [24].

Директива `@import` позволяет импортировать sass-файлы, которые объединяются в отдельном файле, а затем подключаются в главный файл со стилями. `Import` будет искать все файлы со стилями, имеющие расширение `*.sass` или `*.scss`, для их импорта.

Директива `@media` позволяет выявлять заданные в условии разрешения экрана и применять стили только к ним. Медиа-запросы ограничивают ширину макета, с которым ведется работа, и при достижении заданного условия применяют свойства, требующиеся на данном разрешении. Синтаксис пре-процессора позволяет вкладывать медиа-запросы друг в друга, в таком случае их объединяет оператор «and».

Также в SASS предусмотрено наследование, которое возможно благодаря директиве `@extend`. При использовании этой директивы свойства наследуются от одного селектора к другому, что позволяет избавиться от лишнего кода.

На рисунках 29 и 30 приведены примеры использования директив `@import` и `@media`.

```
1 //base styles
2 @import "scss/base";
3
4 //components
5 @import "components/controls/controls";
6 @import "components/main-navigation/main-navigation";
7 @import "components/main-header/main-header";
8 @import "components/main-menu/main-menu";
9 @import "components/menu-catalog/menu-catalog";
10 @import "components/menu-services/menu-services";
11 @import "components/main-tools/main-tools";
12 @import "components/main-search/main-search";
```

Рисунок 29 — Пример использования директивы `@import`

```

html, body {
  font-family: #{map-get($font, body)} !important;
  line-height: 1.5;
  font-size: 16px;
  // generate font size for body
  @each $breakpoint, $width in $grid-breakpoints {
    @media (max-width: $width) {
      @include font-size($font-size, $breakpoint)
    }
  }
}

```

Рисунок 30 — Пример использования директивы @media

Препроцессор SASS обладает большими возможностями, облегчающими жизнь разработчика и в разы ускоряющими его работу. С использованием SASS в разрабатываемом проекте код стал более структурированным и лаконичным [9].

Следующий этап включает в себя компонентную верстку проекта с использованием ранее перечисленных технологий. Итоговый результат сборки и конфигураций представлен на рисунке 31.

Имя	Дата изменения	Тип	Pa
.git	10.06.2019 11:50	Папка с файлами	
assets	09.06.2019 13:15	Папка с файлами	
bin	21.05.2019 11:20	Папка с файлами	
node_modules	09.06.2019 14:47	Папка с файлами	
public	09.06.2019 13:15	Папка с файлами	
vendor	21.05.2019 11:19	Папка с файлами	
	23.05.2019 7:22	Текстовый докум...	
composer.json	23.05.2019 7:22	Файл "JSON"	
composer.lock	23.05.2019 7:22	Файл "LOCK"	
package.json	09.06.2019 14:47	Файл "JSON"	
package-lock.json	09.06.2019 14:47	Файл "JSON"	
README.md	23.05.2019 7:22	Файл "MD"	
webpack.config	23.05.2019 7:22	JetBrains PhpStorm	

Рисунок 31 — Корневая папка проекта на этапе верстки дизайн-макетов

На данном этапе имеется готовая статичная верстка всех страниц по дизайн-макетам.

Был подключен и настроен модуль «Select2» для кастомизации стандартных браузерных селектов, которые используются на страницах с фильтрацией, на странице сравнения товаров и в оформлении заказа.

Пример результата работы плагина продемонстрирован на рисунке 32.

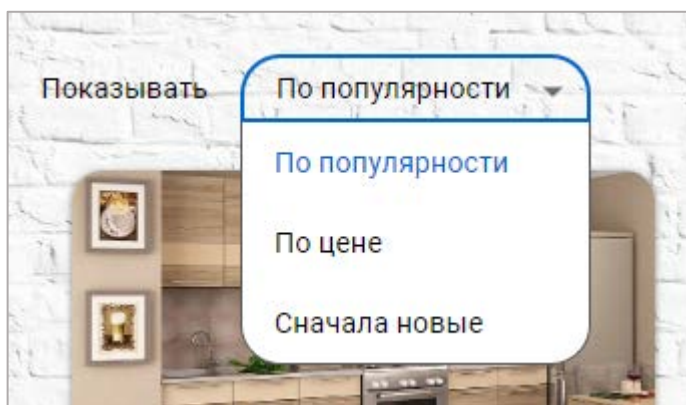


Рисунок 32 — Пример кастомного селективного поля

Пример опций плагина представлен на рисунке 33.

```
13 $(document).ready(function() {
14   const $select = $('.catalog__filter-select, .catalog__sorting-select');
15   $select.css('width', '100%');
16   $select.select2({
17     language: 'ru',
18     allowClear: !$select.prop('multiple') || !$select.attr('multiple'),
19     minimumResultsForSearch: -1
20   });
21 }
```

Рисунок 33 — Пример опций кастомного селективного поля

Установлен и настроен плагин «ionRangeSlider» для кастомизации поля ввода типа «range», представленного на рисунке 34.

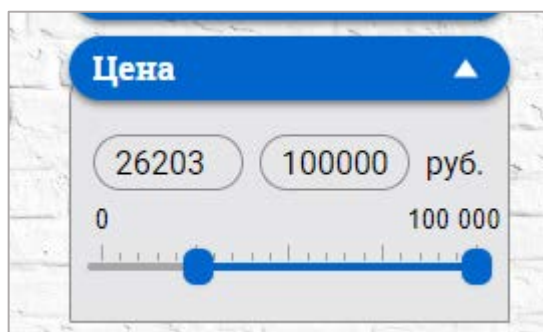


Рисунок 34 — Поле ввода типа «range»

Пример опций плагина «ionRangeSlider» представлен на рисунке 35.

```
$rangeInput.ionRangeSlider( options: {
  prettify_enabled: true,
  skin: "round",
  onChange: function (data) {
    $rangeInputMin.val(data.from);
    $rangeInputMax.val(data.to);
  },
  onUpdate: function (data) {
    $rangeInputMin.val(data.from);
    $rangeInputMax.val(data.to);
  }
});
```

Рисунок 35 — Опции плагина «ionRangeSlider»

Загружен и подключен плагин «Slick» для создания карусели изображений (рисунок 36), который используется на странице товаров.

```
1 $(document).ready(function() {
2   $('.product2__card-slider').slick({
3     slidesToShow: 1,
4     arrows: true,
5     nextArrow: '<svg class="slider-arrow slider-arrow--right"><use xlink:href="#arrow"></use></svg>',
6     prevArrow: '<svg class="slider-arrow slider-arrow--left"><use xlink:href="#arrow"></use></svg>',
7     dots: false,
8     fade: true,
9     asNavFor: '.product2__card-thumbnail'
10  });
11
12  const $sliderThumbnail2 = $('.product2__card-thumbnail');
13  const $sliderlength2 = $('.product2__card-thumbnail').children().length;
14  $sliderThumbnail2.slick({
15    slidesToShow: $sliderlength2 - 1,
16    asNavFor: '.product2__card-slider',
17    dots: false,
18    slide: '.thumbnail-link2',
19    focusOnSelect: true,
20    arrows: false,
21    draggable: false,
22    vertical: true
23  });
24  });
```

Рисунок 36 — Опции плагина «Slick»

После завершения этапа верстки, необходимо установить статичные страницы на систему управления сайтом OpenCart для дальнейшего администрирования проекта и создания функционала для использования возможностей сайта [10].

Для установки OpenCart необходимо скачать и распаковать архив с дистрибутивом с официального сайта, представленного на рисунке 37. Подробную инструкцию по установке можно найти в архиве с дистрибутивом.

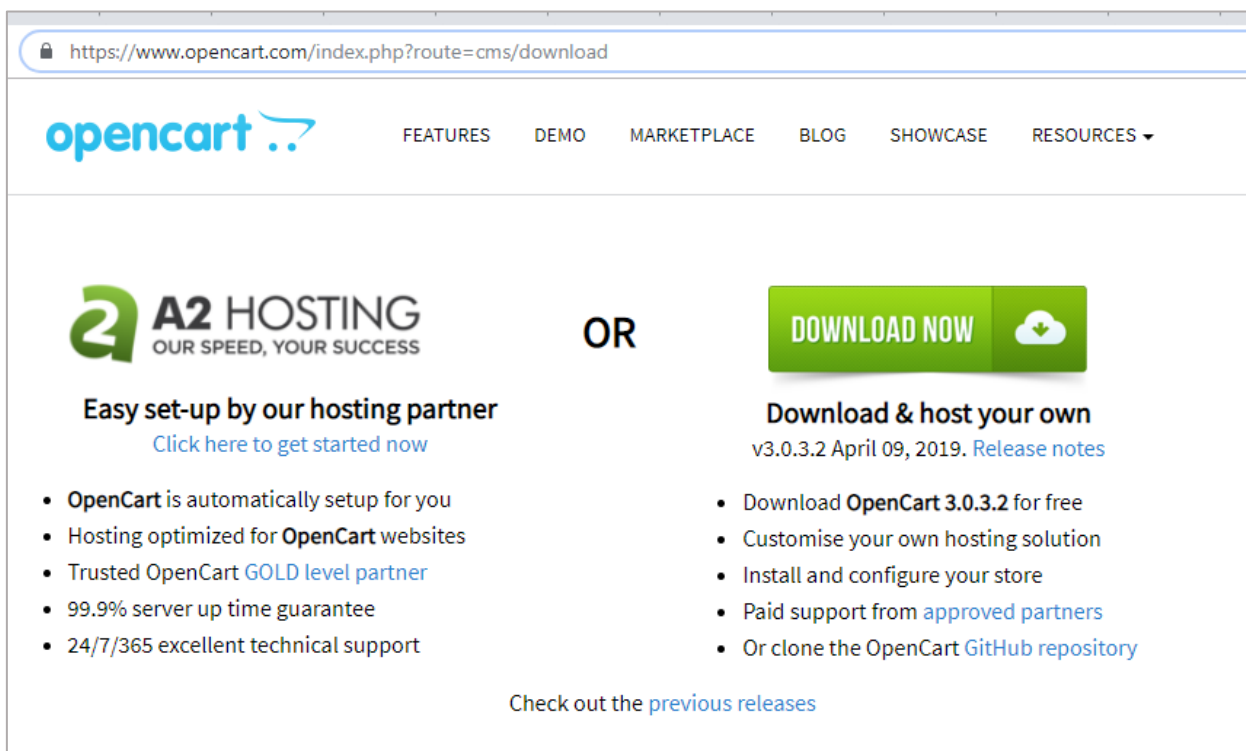


Рисунок 37 — Страница загрузки системы с официального сайта OpenCart

На данный момент полной и актуальной версией является версия 3.0.3.2. Все последние версии представлены на рисунке 38.

Previous & Release Notes		
Previous	Release Notes	
Version	Release Date	Action
3.0.3.2	April 9, 2019	Download
3.0.3.1	January 7, 2019	Download
3.0.3.0	January 2, 2019	Download
3.1.0.0_b	July 27, 2017	Download
3.0.2.0	July 18, 2017	Download
3.0.1.2	July 7, 2017	Download
3.0.1.1	July 4, 2017	Download
3.0.0.0	June 19, 2017	Download
2.3.0.2	August 1, 2016	Download
2.3.0.1	July 31, 2016	Download

Рисунок 38 — Страница со списком последних версий OpenCart

Следующий этап создания интернет-магазина на OpenCart это его установка на локальный web-сервер. В данном проекте использовался OpenServer. Для полноценной работы желательно использовать версию Premium.

Далее устанавливаем OpenServer в рабочую папку с названием интернет-магазина. Запускать OpenServer необходимо с соответствующими параметрами операционной системы, на которой он установлен. В настройках OpenServer необходимо выбрать параметры PHP и Apache и изменить версии на актуальные.

Следующим этапом будет создание базы данных. Для этого необходимо открыть OpenServer и зайти на панели задач в программу PhpMyAdmin.

Для добавления базы данных сайта необходимо зайти во вкладку «Базы данных» в главном окне и создать новую базу данных.

Далее необходимо перезапустить web-сервер. Открыть в браузере новую вкладку и перейти по адресу, который соответствует названию папки интернет-магазина.

На данном шаге выполняется поэтапная установка CMS OpenCart:

1. Принятие лицензионного соглашения.
2. Проверка системы на соответствие требований системы.
3. Ввести данные для доступа к базе данных, и настройки аккаунта администратора сайта (логин, пароль, почта для техподдержки клиентов).
4. Завершающий этап установки и переход в административную часть интернет-магазина. А также предупреждение об удалении папки инсталлятора из CMS.

На данном этапе установка CMS завершена.

При установке CMS OpenCart автоматически создаются 136 таблиц в базе данных. На рисунке 39 продемонстрирован фрагмент базы данных, после установки OpenCart.

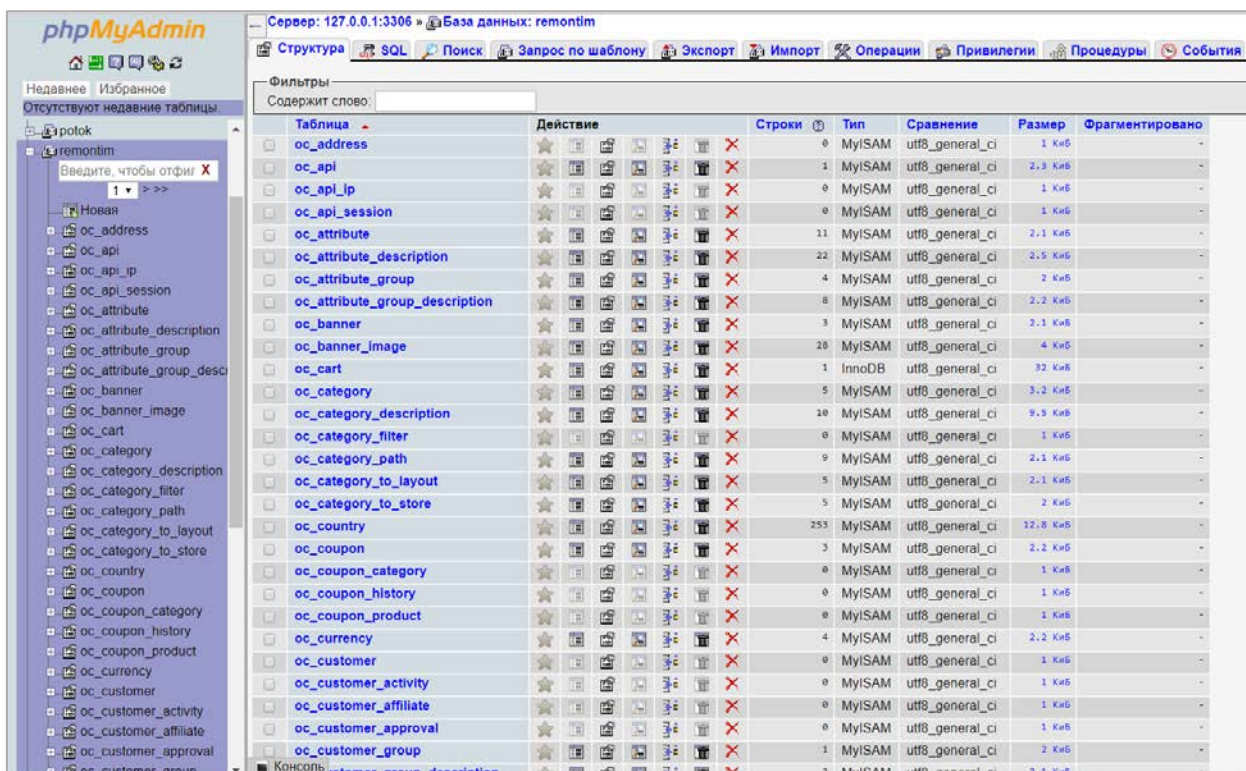


Рисунок 39 — Таблицы в базе данных после установки OpenCart

Чтобы пользоваться настройками интерфейса на русском языке, необходимо русифицировать систему управления. Для этого необходимо скачать архив с официального сайта OpenCart, выбрав соответствующую установленной системе версию — 3.0.3.2.

В данном проекте была установлена бесплатная версия русификатора, представленная на рисунке 40.

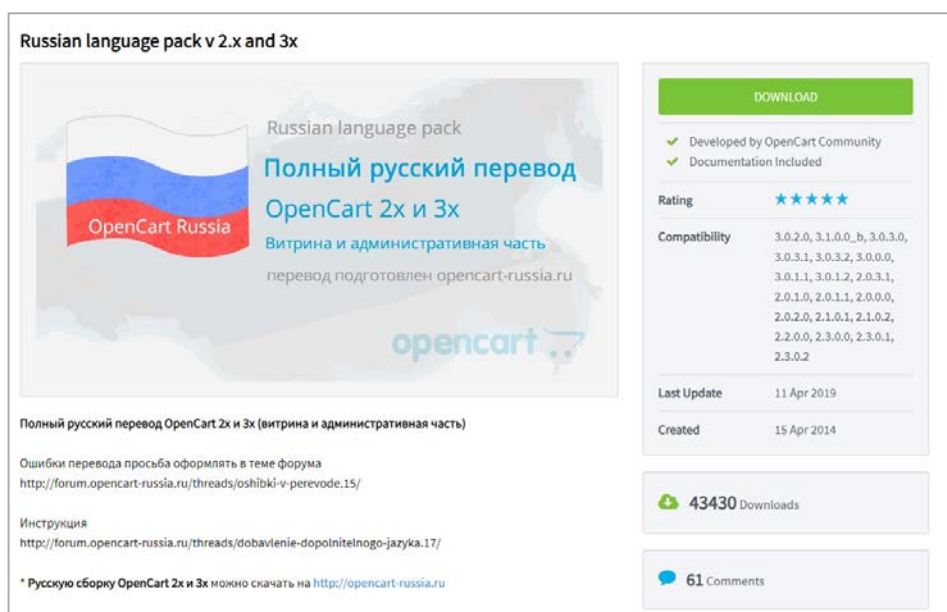


Рисунок 40 — Страница с русификатором на официальном сайте OpenCart

Следуя инструкции, настроить в панели администратора параметры языка. Далее можно настроить валюту и другие необходимые параметры для интернет-магазина.

Для настройки необходимых параметров интернет-магазина необходимо зайти в панель администратора. Система OpenCart содержит огромное количество разных функций, с помощью которых можно максимально точно настроить интернет-магазин под необходимые параметры. Многие из того, что раньше прописывалось кодом вручную, учтено в параметрах данной системы управления. Это значительно сокращает время разработки.

После установки системы управления к шаблону подключаются все ассеты (шрифты, изображения, иконки, CSS-стили и JavaScript код), реализованные на этапе верстки интернет-магазина.

Следующим шагом статичные страницы приводятся в администрируемый вид. Как, например, меню каталога изменяется из повторяющихся HTML элементов в вызов элементов из массива данных [8, 12], как показано на рисунке 41.

```
</p>
<div class="menu-catalog__dropdown">
  {% for category in categories %}
  {% if category.children %}
  <li class="menu-catalog__dropdown-item">
    <a href="{{ category.href }}" class="menu-catalog__dropdown-link">{{ category.name }}</a>
    <div class="menu-catalog__submenu">
      {% for children in category.children|batch(category.children|length / category.column|round(1, 'ceil')) %}
      <div class="menu-catalog__submenu-child">
        <p class="menu-catalog__submenu-heading">
          {{ category.name }}
        </p>
        <ul class="menu-catalog__submenu-list">
          {% for child in children %}
          <li class="menu-catalog__submenu-item">
            <a class="menu-catalog__submenu-link" href="{{ child.href }}">{{ child.name }}</a>
          </li>
          {% endfor %}
        </ul>
      </div>
      {% endfor %}
    </div>
  </li>
  {% else %}
  <li class="menu-catalog__dropdown-item">
    <a href="{{ category.href }}" class="menu-catalog__dropdown-link">{{ category.name }}</a>
  </li>
  {% endif %}
  {% endfor %}
</div>
```

Рисунок 41 — Пример кода, используемого для вывода списка категорий

2.3.3 Этап тестирования, апробирования, внедрения

Не существует методов тестирования сайтов, которые позволили бы со стопроцентной вероятностью выявить ошибки в работе программного обеспечения (продукта). Существует лишь формальный процесс проведения тестирования. Согласно этому процессу, есть возможность доказать, что в программном обеспечении нет ошибок только, с точки зрения выбранного метода тестирования. Таким образом получается, что нет возможности установить и гарантировать, что в программе нет ошибок или дефектов, которые могут возникнуть ввиду человеческого фактора, присутствующего на всех этапах жизненного цикла программного обеспечения. На текущий момент существует большое количество подходов к тестированию программного обеспечения [18].

Процесс тестирования позволяет смоделировать ситуации, при которых программный продукт перестает полностью или частично функционировать или корректно отображаться.

Тестирование front-end части данного интернет-магазина проводилось для определения кроссбраузерности использованных при реализации решений, проверялось отсутствие возникновения консольных ошибок в различных ситуациях.

В ходе тестирования использовались последние версии популярных браузеров. Так как проект не предназначен для использования устаревших браузеров, их поддержка не была заявлена.

Тестирование проводилось в таких браузерах как:

- Google Chrome 74.0.3729;
- Microsoft Edge 44.17763.1.0;
- Internet Explorer 11;
- Mozilla Firefox 67.0.

Тестирование back-end части данного сайта заключалась в проверке работоспособности фильтров, работоспособности форм обратной связи, кор-

ректности работы административной панели сайта, а также на корректность вывода данных.

Стадия тестирования сайта проводилась в течение 7 дней, в результате которой были выявлены и устранены ошибки в работе проекта. В процессе тестирования определялись и устранялись ошибки и неточности, как в сценариях JavaScript, так и в таблице стилей проекта. Проверялось правильное отображение всех элементов интернет-магазина в различных браузерах и на разных мобильных устройствах. Были устранены проблемы с фильтрацией и поиском товаров.

2.4 Технические требования к проекту

Технические требования включают в себя требования как для клиента, так и для сервера.

Разрабатываемый проект доступен для всех популярных операционных систем и актуальных браузеров и ориентирован на широкую аудиторию.

Технические требования для пользователя компьютера заключаются в использовании Google Chrome 66 и выше, Mozilla Firefox 60 и выше, Internet Explorer 11.

Технические требования для пользователя мобильного устройства заключаются в использовании Android 4.0.3 и выше или iOS 8.0.

Такие требования обусловлены необходимостью поддержкой браузером последней версии стандарта HTML и CSS, а также браузер должен поддерживать JavaScript стандарт — ECMAScript 5 [23].

Технические требования к серверу, обеспечивающему работоспособность сайта, имеют иные параметры. Программное обеспечение сервера должно использовать web-сервер Apache или Nging + Apache.

OpenCart работает с интерпретатором PHP от версии 5.4 и выше. Работа на PHP 7.0, 7.1 не требует дополнительных настроек [29].

Скрипт работает с любой базой данных, но для беспроблемной работы скрипта, не требующей дополнительных настроек рекомендован сервер базы данных MySQLi.

Для корректной работы OpenCart необходимо включение следующих модулей PHP:

- Curl — служебная программа командной строки;
- ZIP — расширение, которое позволяет читать и записывать сжатые ZIP-архивы и файлы в них;
- Zlib — модуль возможности сжатия передаваемых страниц;
- GD Library — программная библиотека для динамической работы с изображениями;
- Mcrypt — библиотека шифрования;
- Mbstrings — заставляет работать PHP-приложение во много байтовом окружении. Не работает на PHP 7.2;
- Xml — модуль позволяющий совершать операции с элементами и атрибутами XML-документа.

2.5 Калькуляция проекта

К моменту завершения работы над проектом, можно привести следующую статистику по калькуляции проекта:

1. Проведено сравнение 3 различных CMS для разработки интернет-магазинов.
2. Написано 3404 строк SASS кода.
3. Написано 13547 строк HTML когда.
4. Создано более 30 компонентов сайта.
5. Свёрстано 10 страниц сайта.
6. Потрачено более 12 часов рабочего времени на дискуссии и утверждение с дизайнером и заказчиком.
7. Потрачено 135 часов на разработку продукта.

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе был выполнен ряд требований заказчика:

- разработан интернет-магазина для продажи товаров для ремонта;
- разработан отдельный раздел с услугами магазина;
- возможность фильтрации товаров по различным параметрам;
- возможность сравнения товаров по характеристикам;
- разработаны страницы с акционными позициями товаров;
- возможность совершать поиск товаров на сайте по названию и описанию;
- добавлены информационные страницы «О компании», «Контактные данные магазина» и подробности доставки и оплаты;
- возможность оформления заказа с получением контактных данных покупателя.

С помощью системы OpenCart, с наименьшими финансовыми и временными затратами разработан интернет-магазин, который содержит необходимый перечень товаров, исчерпывающую информацию, дополнительные модули для удобства заказа и связи с покупателем, а также интуитивно понятный пользователю интерфейс.

Использование разработанного программного продукта упрощает процессы продвижения компании, ведения и выдачи информации, что приводит к значительному снижению трудовых затрат, обеспечивает сокращение времени обработки информации, повышает достоверность выдаваемой информации.

В ходе выполнения выпускной квалификационной работы были решены следующие задачи:

- произведен анализ предметной области и существующих систем управления сайтом для создания интернет-магазина;

- выбрана технология реализации интернет-магазина;
- разработана структура интернет-магазина;
- реализована разработанная структура средствами web-технологий;
- произведено тестирование и наполнение контентом.

В заключительном этапе тестирования была проверена корректность отображения и работоспособность функционала сайта в различных браузерах, таких как Google Chrome, Mozilla Firefox, Opera, Internet Explorer. А также проверено корректное отображение с мобильных устройств.

Таким образом, следует считать, все задачи выпускной квалификационной работы полностью выполнены и цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вигерс К. И. Разработка требований к программному обеспечению [Текст] / К. И. Вигерс, Д. Битти. — Санкт-Петербург: БХВ-Петербург, 2015. — 784 с.
1. ГОСТ 19.101-77. Единая система программной документации. Виды программ и программных документов [Текст]. — Введ. 01.01.1980. — Москва: Стандартинформ, 2010. — 24 с.
2. ГОСТ 19.102-77. Единая система программной документации. Стадии разработки [Текст]. — Введ. 01.01.1980. — Москва: Стандартинформ, 2010. — 4 с.
3. ГОСТ 19.105-78. Единая система программной документации. Общие требования к программным документам [Текст]. — Введ. 01.01.1980. — Москва: Стандартинформ, 2010. — 4 с.
4. Документация OpenCart [Электронный ресурс]. — Режим доступа: <https://docs.ocstore.com/> (дата обращения: 26.05.2019).
5. Интерактивные элементы на сайте [Электронный ресурс]. — Режим доступа: <https://easy-it.ru/tag/interaktivnyye-elementy/> (дата обращения: 26.05.2019).
6. Интернет-магазин: общие сведения [Электронный ресурс]. — Режим доступа: <http://alpha-spb.ru/site-shop> (дата обращения: 19.05.2019).
7. Как открыть интернет-магазин: пошаговая инструкция [Электронный ресурс]. — Режим доступа: <https://texterra.ru/blog/kak-otkryt-internet-magazin-poshagovaya-instruktsiya.html> (дата обращения: 29.05.2019).
8. Котеров Д. PHP5 [Текст] / Д. Котеров, А. Костарев. — Санкт-Петербург: БХВ-Петербург, 2016. — 1104 с.
9. Мартин Р. К. Чистый код: создание, анализ и рефакторинг. Библиотека программиста [Текст] / Р. К. Мартин. — Санкт-Петербург: Питер, 2016. — 464 с.

10. Никсон Р. Создаем динамические web-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML [Текст] / Никсон Р. — Санкт-Петербург: Питер, 2016. — 768 с.
11. Описание VirtueMart: возможности и преимущества [Электронный ресурс]. — Режим доступа: <http://shine-design.com/cms-joomla-with-virtuemart.html> (дата обращения: 15.05.2019).
12. Подборка документации и справочников по PHP [Электронный ресурс]. — Режим доступа: <http://www.php.ru/> (дата обращения: 05.05.2019).
13. Прохоренок Н. HTML, JavaScript, PHP и MySQL. Джентльменский набор web-мастера [Текст] / Н. Прохоренок. — Санкт-Петербург: БХВ-Петербург, 2015. — 768 с.
14. Рынок интернет-торговли в России: проблемы и перспективы развития [Электронный ресурс]. — Режим доступа: <https://sibac.info/studconf/econom/xlviii/64721> (дата обращения: 15.05.2019).
15. Система управления сайтом WordPress [Электронный ресурс]. — Режим доступа: <https://ru.wordpress.com/> (дата обращения: 29.05.2019).
16. Цели и задачи создания сайта [Электронный ресурс]. — Режим доступа: <https://seovedenie.ru/sozдание-sajta> (дата обращения: 23.05.19).
17. Структура и оформление интернет-магазина [Электронный ресурс]. — Режим доступа: <https://kakzarabativat.ru/internet-magazin/struktura-i-oformlenie-internet-magazina-chto-obyazatelno-dolzno-byt-na-sajte-vashego-internet-magazina/> (дата обращения: 25.05.2019).
18. Теория тестирования программного обеспечения [Электронный ресурс]. — Режим доступа: <http://www.alexproger.narod.ru/Interv/test1.html/> (дата обращения: 08.05.2019).
19. Технологии web-разработки в 2018 году [Электронный ресурс]. — Режим доступа: <https://webformyself.com/tehnologii-veb-razrabotki-v-2018-godu/> (дата обращения: 07.05.2019).
20. Что такое CMS? Для чего она нужна? Какие бывают CMS? [Электронный ресурс]. — Режим доступа: <http://moolkin.ru/joomla/cms/chto-takoe-cms-dlya-chego-ona-nuzhna-kakie-byvayut-cms> (дата обращения: 03.05.2019).

21. Этапы разработки программного обеспечения [Электронный ресурс]. — Режим доступа: <http://ru.intechcore.com/stages-software-development/> (дата обращения: 28.04.2019).

22. An Introduction to CSS Pre-Processors: SASS, LESS and Stylus [Электронный ресурс]. — Режим доступа: <https://htmlmag.com/article/anintroduction-to-css-preprocessors-sass-less-stylus> (дата обращения: 29.05.2019).

23. ECMAScript 5 [Электронный ресурс]. — Режим доступа: http://www.w3schools.com/js/js_es5.asp (дата обращения: 29.05.2019).

24. Expressive, dynamic, robust CSS — expressive, robust, feature-rich CSS preprocessor [Электронный ресурс]. — Режим доступа: <http://styluslang.com/> (дата обращения: 29.05.2019).

25. Getting started | Less.js [Электронный ресурс]. — Режим доступа: <http://lesscss.org/> (дата обращения: 28.05.2019).

26. GitHub — postcss/autoprefixer: Parse CSS and add vendor prefixes to rules by Can I Use [Электронный ресурс]. — Режим доступа: <https://github.com/postcss/autoprefixer> (дата обращения: 29.05.2019).

27. npm [Электронный ресурс]. — Режим доступа: <https://www.npmjs.com/> (дата обращения: 29.05.2019).

28. Sass: Syntactically Awesome Style Sheets [Электронный ресурс]. — Режим доступа: <http://sass-lang.com/> (дата обращения: 29.05.2019).

29. System Requirements — OpenCart Documentation [Электронный ресурс]. — Режим доступа: <https://docs.opencart.com/requirements/> (дата обращения: 01.06.2019).

30. Using loaders [Электронный ресурс]. — Режим доступа: <https://webpack.github.io/docs/using-loaders.html> (дата обращения: 28.05.2019).

31. Webpack module bundler [Электронный ресурс]. — Режим доступа: <https://webpack.github.io/> (дата обращения: 30.05.2019).

32. WooCommerce [Электронный ресурс] / Плагин для WordPress. — Режим доступа: <https://woocommerce.com/> (дата обращения: 18.05.2019).

ПРИЛОЖЕНИЕ

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»**

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
Направление подготовки 09.03.02 Информационные системы и технологии
Профиль «Информационные технологии в медиаиндустрии»

УТВЕРЖДАЮ
Заведующий кафедрой
И. А. Сулова
и.о. фамилия

« ____ » _____ 201_ г.

ЗАДАНИЕ на выполнение выпускной квалификационной работы бакалавра

студента (ки) 4 курса группы ИТМ-403
Куноф Павла Александровича
фамилия, имя, отчество полностью

1. Тема Интернет-магазин товаров для ремонта

утверждена распоряжением по институту от « ____ » _____ 20 г. № ____

2. Руководитель Власова Наталья Сергеевна
фамилия, имя, отчество полностью

Канд. пед. наук доцент доцент РГППУ
ученая степень ученое звание должность место работы

3. Место преддипломной практики РГППУ каф. ИС

4. Исходные данные к ВКР Тексты и фотографии от заказчика

5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)

Анализ литературных и интернет-источников по разработке интернет-магазина.

Исследование средств для разработки интернет-магазина

Разработка структуры интернет-магазина

Реализация разработанной модели интернет-магазина

Наполнение интернет-магазина контентом

Тестирование интернет-магазина

6. Перечень демонстрационных материалов: Презентация в MS Power Point, промо-ролик

