

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д. В. Топольский
«___» _____ 2022 г.

РАЗРАБОТКА ГИБРИДНОГО ИНТЕРНЕТ-МАГАЗИНА ОДЕЖДЫ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2022.202 ПЗ ВКР

Руководитель работы,
к.п.н., доцент каф. ЭВМ
_____ М. А. Алтухова
«___» _____ 2022 г.

Автор работы,
студент группы КЭ-405
_____ Р. А. Абляимов
«___» _____ 2022 г.

Нормоконтролёр,
к.п.н., доцент каф. ЭВМ
_____ М. А. Алтухова
«___» _____ 2022 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Д. В. Топольский
«___» _____ 2022 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра
студенту группы КЭ-405
Аблялимову Рустему Арсеновичу,
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

Тема работы: «Разработка гибридного Интернет-магазина одежды»
утверждена приказом по университету от 12 декабря 2021 г. № 308/141.

Срок сдачи студентом законченной работы: 1 июня 2022 г.

Исходные данные к работе.

Разработать интернет-магазин со следующим функционалом:

- 1) для незарегистрированных и зарегистрированных пользователей:
 - а) просмотр каталога товаров;
 - б) добавление товара в корзину;
 - в) выбор из вариации атрибутов товара;
 - г) оформить заказ;
 - д) редактировать содержание корзины;

- е) выбор способа оплаты заказа;
 - ж) выбор способа доставки заказа;
- 2) для зарегистрированных пользователей:
- а) возможность авторизоваться;
 - б) редактирование пользователем персональной информации в личном кабинете;
 - в) возможность изменять пароль;
 - г) возможность оставлять отзыв и выставлять рейтинг в карточке товара;
 - д) возможность отслеживать статус заказа в личном кабинете;
- 3) для администратора:
- а) возможность добавлять/удалять товары в каталоге;
 - б) возможность добавлять/удалять категории в каталоге;
 - в) возможность добавлять/удалять атрибуты товара в каталоге;
 - г) возможность удалять отзывы в карточке товара;
 - д) возможность добавлять/удалять новые товары в каталоге;
 - е) возможность добавлять/удалять ссылки на социальные сети;
 - ж) возможность менять контактную информацию;
 - з) возможность добавлять/удалять способы доставки;
 - и) возможность добавлять/удалять способы оплаты;
 - к) возможность изменять информацию о запасах товара.

Перечень подлежащих разработке вопросов:

- 1) анализ предметной области, анализ аналогичных проектов;
- 2) подбор программных средств для создания веб интернет-магазина;
- 3) описание архитектуры проекта;
- 4) изложение и описание архитектуры веб интернет-магазина;
- 5) программная реализация веб интернет-магазина;
- 6) тестирование разработанного веб интернет-магазина.

Дата выдачи задания: 1 декабря 2021 г.

Руководитель работы _____ /М. А. Алтухова/

Студент _____ /Р. А. Абляимов/

КАЛЕНДАРНЫЙ ПЛАН

| Этап | Срок сдачи | Подпись руководителя |
|--|------------|-------------------------|
| Введение и обзор литературы. Анализ предметной области. Анализ технологических решений | 07.02.2022 | |
| Определение функциональных и нефункциональных требований. Проектирование серверной и пользовательской части реализуемого проекта, моделирование базы данных. | 07.03.2022 | |
| Реализация проекта | 04.04.2022 | |
| Тестирование проекта | 10.05.2022 | |
| Компоновка текста работы и сдача на нормоконтроль | 26.05.2022 | |
| Подготовка презентации и доклада | 12.06.2022 | |

Руководитель работы _____ /М. А. Алтухова/

Студент _____ /Р. А. Абляимов/

АННОТАЦИЯ

Р. А. Абляимов. Разработка интернет-магазина для одежды. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2022, 117 с., 91 ил., библиогр. список – 14 наим.

В рамках выпускной квалификационной работы выполнены все этапы, связанные с разработкой интернет-магазина для реально существующего Бренда одежды по заказу. Рассматриваются и анализируются решения аналогичных проектов с целью выявления положительных и отрицательных качеств для создания удобного интерфейса и решающего все задачи функционала. Произведен анализ современных технологий для создания Веб-приложений, а именно: языков программирования (JavaScript, PHP), языка разметки (HTML) и языка таблицы стилей (CSS), портативная серверная платформа (OpenServerPanel), Веб интерфейс для администрирования СУБД MySQL (phpMyAdmin) и СУБД MySQL. CMS WordPress и его плагин Woocommerce.

По завершении этапа разработки, выполнено альфа-тестирование разработанного приложения для работы над поставленными задачами. Для оценки результатов работы используются специально разработанные задачи, тестирующие корректность работы приложения во всех возможных вариантах использования.

СОДЕРЖАНИЕ

| | |
|---|-----------|
| ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ..... | 9 |
| ВВЕДЕНИЕ..... | 10 |
| 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ | 12 |
| 1.1 Обзор вариантов реализации | 12 |
| 1.2 Обзор аналогов..... | 15 |
| 1.3 Анализ технологических решений | 26 |
| 2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ..... | 37 |
| 2.1 Функциональные требования | 37 |
| 2.1.1 Требования к функционалу подсистемы авторизации и регистрации пользователя | 37 |
| 2.1.2 Требования к функционалу подсистемы взаимодействия с личным кабинетом..... | 38 |
| 2.1.3 Требования к функционалу подсистемы добавления товара в корзину | 39 |
| 2.1.4 Требования к функционалу подсистемы оформления заказа | 39 |
| 2.1.5 Требования к функционалу подсистемы создания отзывов и выставления рейтинга в карточке товара..... | 40 |
| 2.1.6 Требования к функционалу подсистемы администрирования приложения..... | 41 |
| 2.2 Нефункциональные требования..... | 42 |
| 2.2.1 Требования к пользователям | 42 |
| 2.2.2 Требования к объёму данных | 43 |
| 3 ПРОЕКТИРОВАНИЕ | 44 |
| 3.1 Архитектура предлагаемого решения | 44 |

| | | |
|------------|--|----|
| 3.2 | Интерфейс пользователя | 45 |
| 3.3 | Серверная часть | 45 |
| 3.4 | Моделирование базы данных MySQL | 46 |
| 4 | РЕАЛИЗАЦИЯ | 53 |
| 4.1 | Главная страница | 53 |
| 4.2 | Страница «О нас» | 58 |
| 4.3 | Страница «Контакты» | 60 |
| 4.4 | Каталог товаров | 61 |
| 4.5 | Карточка товара | 63 |
| 4.6 | Личный кабинет пользователя | 64 |
| 5 | ТЕСТИРОВАНИЕ | 66 |
| 5.1 | Тестирование форм ввода | 66 |
| 5.2 | Тестирование безопасности | 66 |
| 5.3 | Тестирование авторизации | 67 |
| 5.4 | Тестирование мультиязычности сайта | 68 |
| 5.5 | Тестирование написания отзыва и выставления рейтинга товару | 69 |
| 5.6 | Тестирование оформления заказа | 70 |
| 5.7 | Тестирование адаптивности интернет-магазина | 74 |
| | ЗАКЛЮЧЕНИЕ | 76 |
| | Список использованных источников | 77 |
| | ПРИЛОЖЕНИЕ А Интерфейсы сайтов, приведенных в обзоре | 79 |
| | ПРИЛОЖЕНИЕ Б Код программы | 97 |

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие сокращения и обозначения:

CMS (от англ. Content management system) – система управления контентом

IDE (от англ. Integrated development environment) – единая среда разработки

SQL (от англ. Structured Query Language) – язык структурированных запросов

SSL (от англ. Secure Sockets Layer) – уровень защищённых сокетов

ООП – объектно-ориентированное программирование

Сайдбар (от англ. Sidebar) – боковая панель сайта

Футер (от англ. Footer) – подвал сайта, блок в нижней части страницы

Хедер (от англ. Header) – шапка сайта, блок в верхней части страницы

ВВЕДЕНИЕ

Благодаря обширной интернетизации общества и всех сфер жизни, владельцы многих брендов осознали, что благодаря использованию интернет технологий, они могут существенно расширить географию своей торговли, сделав свой продукт доступным для покупателей из всех точек мира. По данным Ассоциации компаний интернет торговли (АКИТ), объём рынка e-commerce в России в 2021 году достиг 3.6 трлн. рублей, а доля продаж через интернет достигла 9.2% от всего объёма розничных продаж на российском рынке [1]. Это подтверждает тот факт, что Интернет-торговля – важный аспект ведения бизнеса любой продающей компании. В то же время, благодаря Интернет-магазинам, владельцы имеют возможность сэкономить деньги на аренде помещения и оплате труда сотрудников, занятых в точках розничной торговли.

Сейчас у заказчика, желающего обзавестись сайтом для ведения интернет торговли, есть несколько вариантов:

1) взять в аренду готовый интернет-магазин; при этом функционал и дизайн зачастую изменить невозможно, так как формально ресурс принадлежит третьим лицам;

2) разработать новый сайт на основе шаблона; при этом дизайн и функционал изменить возможно, но в ограниченных пределах;

3) приобрести готовое решение в собственное пользование; предложенные варианты дизайнов и сервисов не всегда могут удовлетворять требованиям заказчика, однако возможно сделать доработку или переработку сайта, но с привлечением дополнительных ресурсов;

4) разработать новый сайт с индивидуальным дизайном и функционалом – лучший вариант, для заказчика, который хочет получить уникальный продукт.

В данной работе рассматривается задача проектирования интернет-магазина для реализации товара фирмы «ШПИЛЬ». Это дизайнерский бренд одежды челябинского производителя. В данный момент основной площадкой торговли является группа в социальной сети «ВКонтакте». С целью расширения охвата аудитории потенциальных покупателей и продвижению бренда в сети «интернет», владельцами было принято решение о создании собственного интернет-магазина.

Являясь маленькой компанией, заказчик не имеет возможности вложить большие средства в разработку, поэтому данный проект осуществляется в рамках дипломной работы.

Целью данной выпускной квалификационной работы является разработка гибридного интернет-магазина для бренда одежды «ШПИЛЬ» для осуществления торговли в сети Интернет.

Задачи работы:

- 1) провести анализ предметной области;
- 2) определить необходимый функционал приложения;
- 3) описать архитектуру проекта;
- 4) реализовать систему опираясь на поставленные задачи;
- 5) провести тестирование интернет-магазина.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

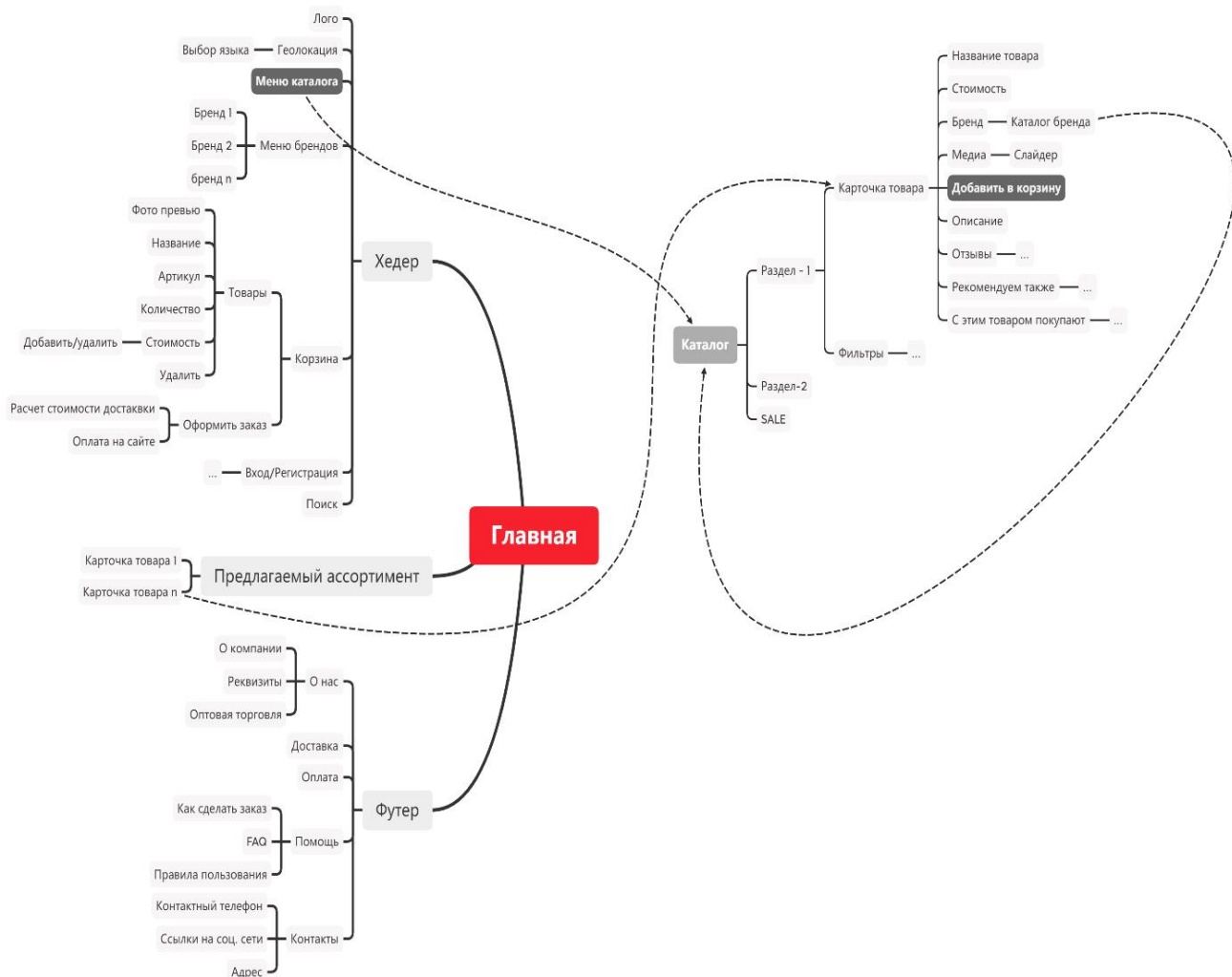
1.1 Обзор вариантов реализации

В настоящее время все интернет-магазины можно условно разделить на три категории. К первой категории отнесем самый компактный вид сайтов – посадочные страницы (лендинг пейдж – от англ. посадочная страница). Это одностраничный веб сайт с ограниченным функционалом. Структура лендинга показана на рисунке 1. Область страницы условно разделяется на экраны.



Рисунок 1 – Информационная архитектура лендинг страницы

Вторая категория – это многостраничные интернет-магазины с большим ассортиментом и широким функционалом. Информационная архитектура представлена на рисунке 2. Для данной категории характерно: большое количество контента, разнообразие элементов управления, (многоуровневое меню).

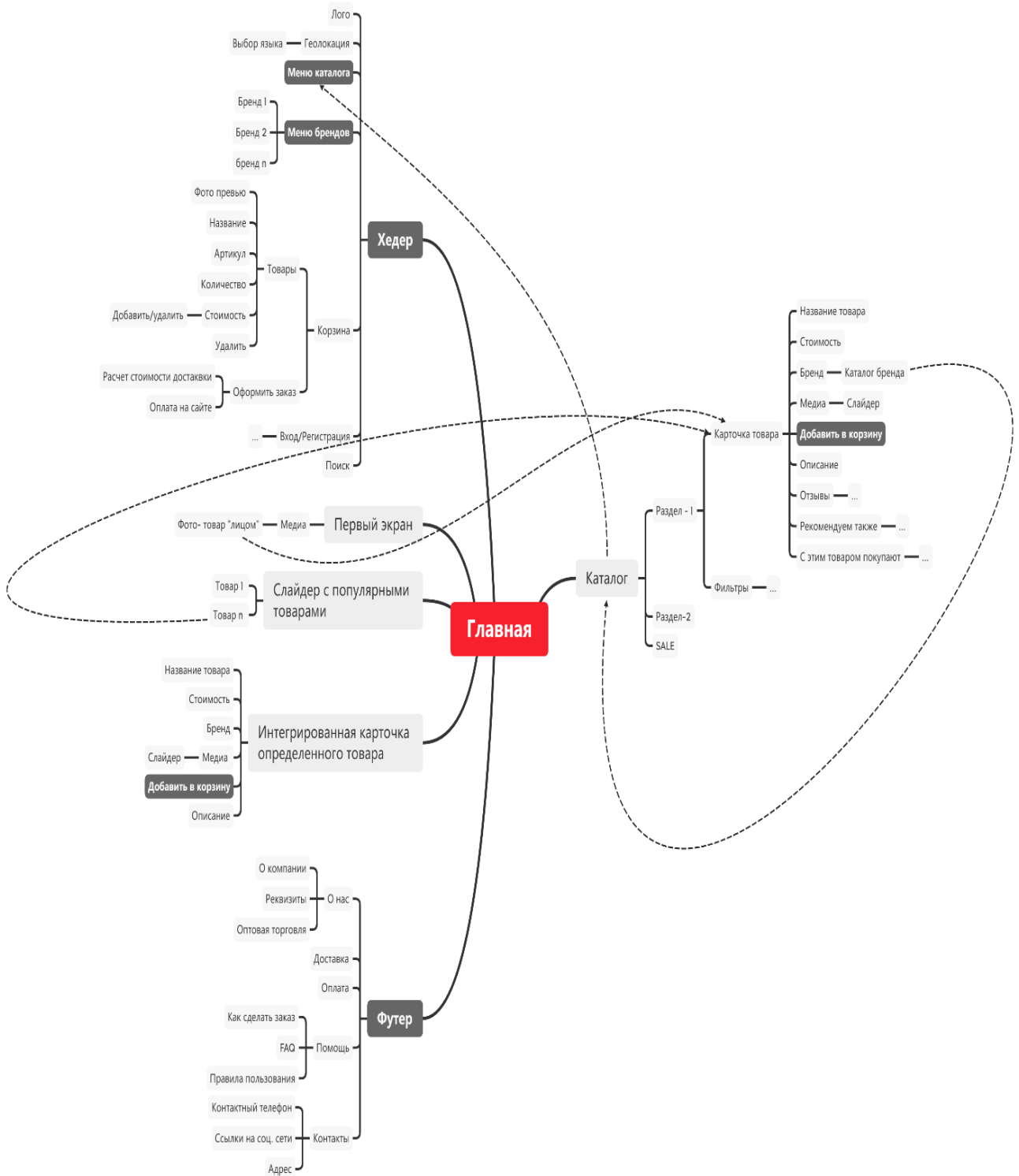


Presented with XMind

Рисунок 2 – Информационная архитектура многостраничного интернет-магазина

Третий вариант занимает промежуточное положение между первым и вторым. Поэтому назовем его гибридным.

Гибридный интернет-магазин представляет собой многостраничный проект, в составе которого есть лендинг. Благодаря этому можно совместить достоинства посадочной страницы с многостраничным сайтом, информация представлена на рисунке 3.



Presented with XMind

Рисунок 3 – Информационная архитектура гибридного интернет-магазина

У бренда «ШПИЛЬ» на данный момент всего 7 товаров на продажу, что говорит в пользу того, чтобы делать посадочную страницу, однако реализация гибридного интернет-магазина позволит в будущем при увеличении ассортимента расширить каталог не меняя структуру сайта. Также будет возможно добавление в ассортимент продукции других брендов с производителями которых планируется сотрудничество.

1.2 Обзор аналогов

Важной и необходимой информацией для разработки интернет-магазина помимо программной части являются архитектура создаваемого интернет-магазина, принцип её работы, оформление меню навигации, также важно узнать информацию о каталогах товаров и услуг, расположение, отображение и так далее, а так же новости связанные с товарами компании, посвященные определенной тематике, наличие определенных онлайн-сервисов (например онлайн оплата и заказ доставки товаров и услуг), ссылки на контактную информацию, информацию о компании и методах её работы. Помимо всего вышеперечисленного в интернет-магазинах отображаются статьи и новости, касающиеся продаваемых товаров и услуг.

Ripndip [2]. Данный интернет-магазин посвящен продаже одежды и аксессуаров одноименного бренда, дислоцирующегося в штате Калифорния, США. На рисунке 4 представлена его информационно-функциональная архитектура.



Рисунок 4 – Информационно-функциональная архитектура

Рассматривая сайт, в первую очередь обратим внимание на шапку сайта, где мы можем увидеть форму для поиска, кнопку позволяющую сменить язык и валюту, которая будет отображаться рядом с товаром, далее перечислены категории товаров, нажатие на которые перенаправит вас на страницу с товарами из данной категории, а также кнопка «CART», перенаправляющая пользователя в корзину с товарами (рисунок А.1).

В теле главной страницы (рисунок А.2) расположены видеоролик, повествующий о том, что в ассортименте появились скейтборды, динамический новостной слайдер, повествующий об акциях и ивентах проводимых брендом, далее идут товарные новинки различных категорий вплоть до футера сайта.

В подвале сайта, как и принято располагаются кнопки, перенаправляющие пользователя на страницы с контактной информацией, ссылки на соц. сети бренда, страницы с формой для обратной связи, форма для подписки на новостную рассылку. Также там располагается кнопка входа в аккаунт, что является нетипичным расположением для данного объекта на страницах интернет-магазинов (рисунок А.3).

Выбрав какую-либо категорию товара, например footwear, мы попадаем страницу, где представлен весь ассортимент выбранного категории. Слева располагаются фильтры, где можно выбрать желаемые размер и цвет (рисунок А.4).

В карточке товара доступны функции выбора размера, добавления товара в корзину, выбора количества единиц товара, а также кнопка отправки уведомления в случае поступления нужного размера, отсутствующего в данный момент в наличие (рисунок А.5).

В целом функционал данного интернет-магазина достаточно обширный, из недостающего можно упомянуть отсутствие возможности оставления отзыва в карточке товара, однако архитектура не удовлетворяет требованиям гибридного интернет-магазина.

Ricewear [3]. Данный интернет-магазин, как пример того, на что в плане функционала следует ориентироваться прислал заказчик. На рисунке 5 отображена его информационно-функциональная архитектура.

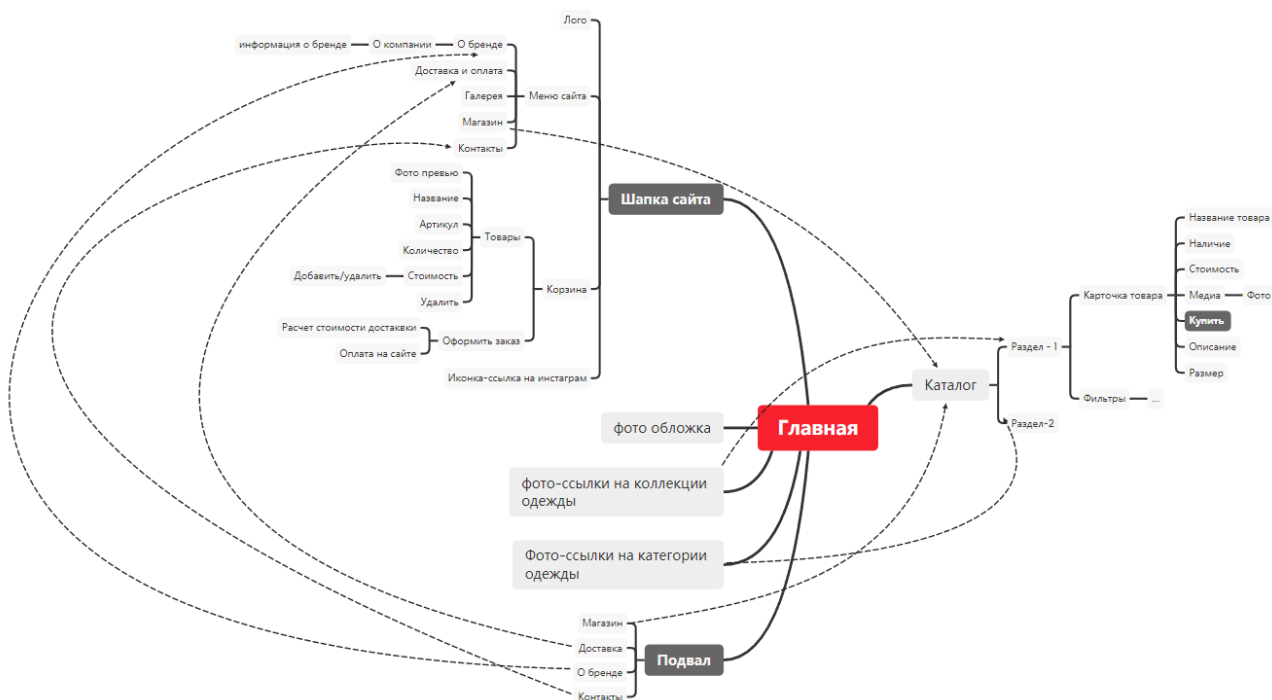


Рисунок 5 – Информационно-функциональная архитектура ricewear.com

В хедере сайта находится меню, содержащее 5 ссылок, перенаправляющие пользователя при нажатии на новые страницы, содержащие контент и информацию, соответствующие названиям, в левом верхнем углу расположена логотип-ссылка на инстаграм бренда (рисунок А.6).

Главная страница представляет собой лендинг пейдж, а именно фото занимающее все пространство экрана (рисунок А.7), следом располагаются фото-ссылки, перенаправляющие пользователей при нажатии на страницы с ассортиментом новых коллекций (рисунок А.8), далее, также размещены фото-ссылки, перенаправляющие пользователя на страницы с соответствующим картинке ассортиментом товара, футболки слева и худи справа (рисунок А.9).

В футере располагаются меню дублирующее меню из хедера (рисунок А.10).

Нажав по ссылке «магазин» в хедере или футере, пользователь перенаправляется на новую страницу, где содержится весь ассортимент, имеющийся в наличии (рисунок А.11). Фильтры располагаются в сайдбаре,

который находится в левой части страницы, там пользователь может выбрать нужную ему категорию товара, ценовой диапазон, размер (рисунок А.12).

Карточка товара представляет собой страницу, на которой располагается вертикальный фото товара, описание товар, кнопка добавление товара в корзину, выпадающий список имеющихся размеров (рисунок А.13).

Мобильная версия интернет-магазина представлена на рисунке А.14.

Подводя итоги, можно сделать вывод что данный сайт обладает недостаточно обширным функционалом, в частности отсутствует возможность зарегистрироваться, а следовательно, и отследить заказ в личном кабинете, форма в карточке товара для написания или прочтения отзыва, возможность сменить язык.

Novaya [4]. Этот интернет-магазин, как и предыдущие посвящен продаже одноименного бренда одежды. На рисунке 6 представлена его информационно-функциональная архитектура.

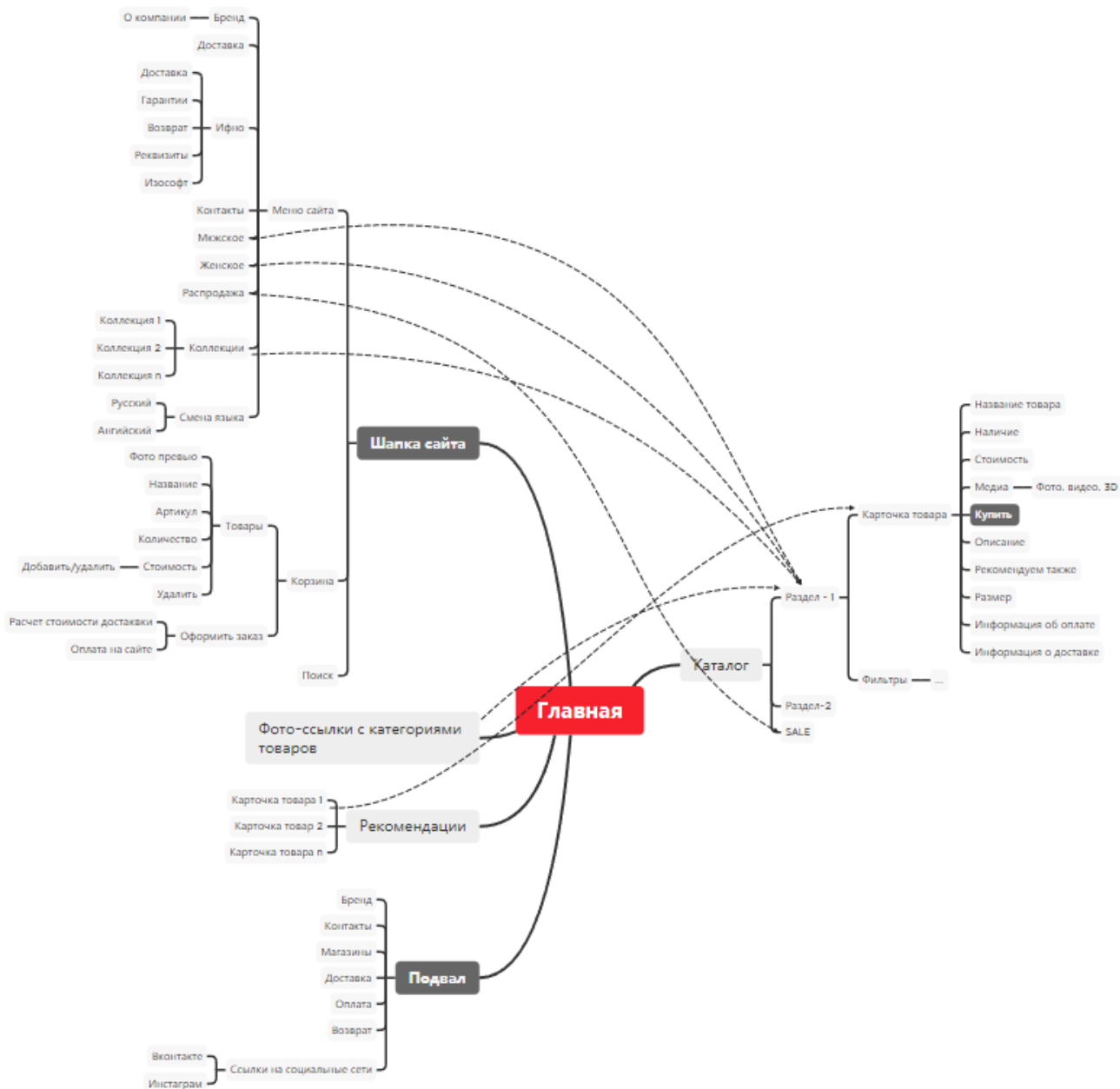


Рисунок 6 – Информационно-функциональная архитектура powayaawear.com

В хедере сайта находится меню (рисунок А.15), содержащее 9 ссылок, перенаправляющие пользователя при нажатии на новые страницы, содержащие контент и информацию, соответствующие названиям, кнопка «English» позволяет перевести все содержимое сайта на английский язык.

Главная страница представляет собой лендинг пейдж, в начале пользователь видит две фото-ссылки, которые при нажатии перенаправляют на страницы с соответствующим ассортиментом (рисунок А.16), следом располагаются карточки товаров, а именно новые поступления в женском (рисунок А.17) и мужском ассортименте (рисунок А.18).

В подвале сайта располагаются меню с ссылками от части дублирующее меню в шапке сайта (рисунок А.19).

Выбрав определенную категорию в выпадающем меню «Men», а именно футболки, пользователь перенаправляется на новую страницу с выбранным ассортиментом (рисунок А.20).

Карточка товара представляет собой страницу, на которой располагается галерея фото товара, описание товара, кнопка добавление товара в корзину, переключатель размеров (рисунок А.21).

Интернет-магазин оптимизирован под мобильные устройства (рисунок А.22).

Подводя итоги, можно сделать вывод что данный сайт обладает недостаточно обширным функционалом, в частности отсутствует возможность зарегистрироваться, форма в карточке товара для написания или прочтения отзыва, возможность выбора язык ограничена лишь русским и английским.

Volchok [5]. Этот интернет-магазин, как и предыдущие посвящен продаже одноименного бренда одежды. На рисунке 7 отображена его информационно-функциональная архитектура.

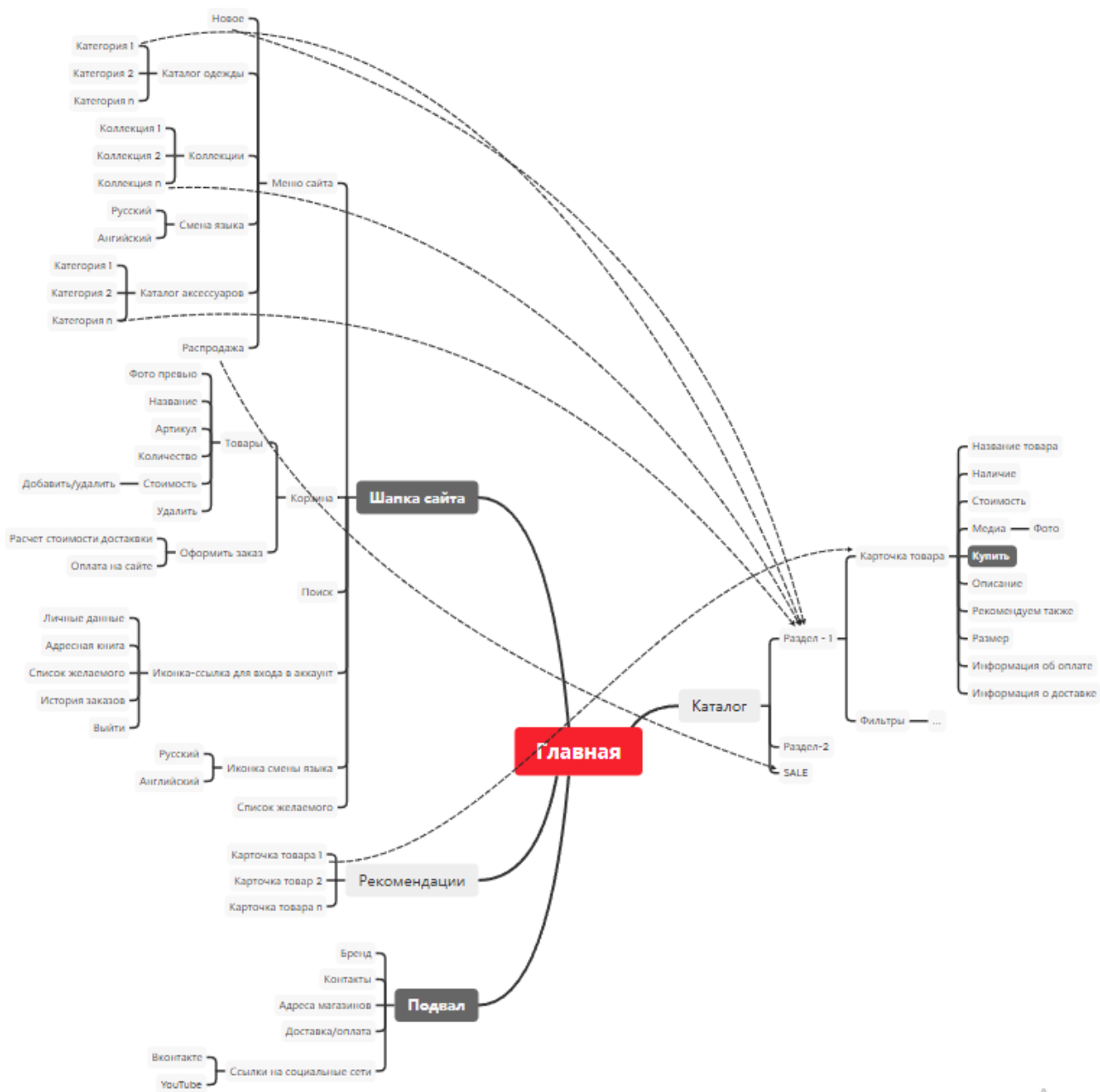


Рисунок 7 – Информационно-функциональная архитектура volchok.ru

В хедере сайта находится меню, содержащее три ссылки, перенаправляющие пользователя при нажатии на новые страницы, содержащие контент и информацию, соответствующие названиям, а также три выпадающих меню, с одеждой, где представлены все категории, аксессуарами и коллекциями (рисунок А.23). В правой части экрана располагаются иконки авторизации/регистрации пользователя, иконка формы поиска, кнопка смены

языка с русского на английский и наоборот, список понравившегося и корзина (рисунок А.24).

Главная страница представляет собой ассортимент товаров со скидкой, справа сверху расположены фильтры размеров и упорядочивания товара по новизне, возрастания и убывания цены (рисунок А.25).

В футере располагаются ссылки на страницы с информацией об оплате/доставке, партнерах, адресов магазинов, контакты, ссылки на социальные сети (рисунок А.26).

Карточка товара представляет собой страницу, на которой располагается фото товара, описание товар, описание доставки, состав из чего сделан продукт, кнопка добавление товара в корзину, кнопка добавления товара в список желаемого, переключатель размеров (рисунок А.27).

Личный кабинет пользователя выглядит следующим образом (рисунок А.28). Имеется возможность заполнить адресную книгу, посмотреть историю заказов, зайти в список понравившегося «закладки».

Мобильная версия интернет-магазина представлена на рисунке А.29.

Подводя итоги, можно сделать вывод что данный сайт обладает достаточно обширным функционалом, из недостатков можно отметить отсутствие возможности оставлять и смотреть отзывы в карточке товара.

Krakatau [6]. Этот интернет-магазин, как и предыдущие посвящен продаже одноименного бренда одежды. На рисунке 8 отображена его информационно-функциональная архитектура.

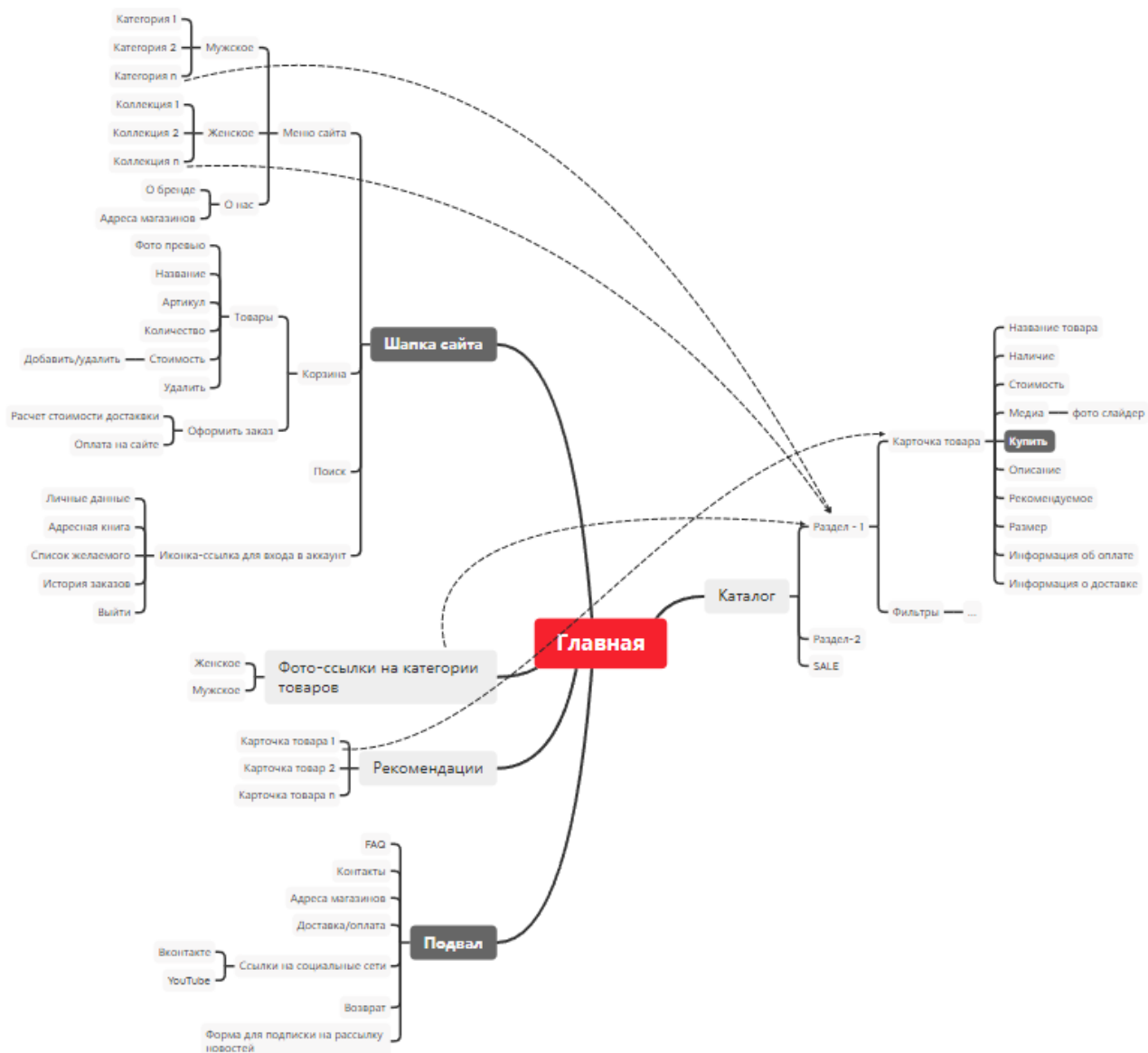


Рисунок 8 – Информационно-функциональная архитектура krakatauwear.com

В хедере сайта находится меню, содержащее два выпадающих меню: «мужское» и «женское», содержащие категории товаров, и ссылка, перенаправляющая на страницу с информацией о бренде. В правой части хедера находится форма поиска, иконка-ссылка, перенаправляющая пользователя на страницу с адресом торговой точки магазина, иконка-ссылка для авторизации/регистрации пользователя, корзина (рисунок А.30).

На главной странице в начале располагаются карточки некоторых товаров (рисунок А.31), далее идут две фото-ссылки, перенаправляющие пользователя

при нажатии на соответствующие страницы с товарами для мужчин или женщин (рисунок А.32).

В футере находятся иконки-ссылки на социальные сети бренда, также ссылки на страницы, содержащие информацию о доставке, возврате, оплате, контактах, пользовательских соглашениях (рисунок А.33).

нажав по ссылке «магазин» в хедере, выбрав определенную категорию пользователь перенаправляется на новую страницу, где содержится весь ассортимент, имеющийся в наличии (рисунок А.34). Фильтры располагаются в сайдбаре, который находится в верхней части страницы, там пользователь может выбрать нужную ему размер и цвет товара.

Карточка товара представляет собой страницу, на которой располагается галерея-слайдер с фото товара, кнопка добавление товара в корзину, переключатель размеров, таблица размеров (рисунок А.35).

В личном профиле, пользователь может в окне «адрес» заполнить свой адрес, который будет при оформлении заказа автоматически вписываться нужные поля, во вкладке «заказы» пользователь может отслеживать свои заказы (рисунок А.36).

Мобильная версия интернет-магазина представлена на рисунке А.37.

Рассмотрев вышеперечисленные интернет-магазины, можно сделать вывод, что все они обладают достаточным функционалом для удобного пользования. Стоит учесть недостатки каждого сайта и реализовать недостающий функционал в конечном проекте.

1.3 Анализ технологических решений

Для компилирования и отладки кода, понадобится интегрированная среда разработки. Microsoft Visual Studio – это многофункциональная программа, которая поддерживает многие аспекты разработки программного обеспечения. Интегрированная среда разработки Visual Studio – это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства авто завершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки.

PHP – это распространённый язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML, хороший язык для разработки проектов на CMS, в частности CMS WordPress написана на PHP.

Достоинства:

- 1) ориентация на Веб-разработку;
- 2) кроссплатформенность;
- 3) бесплатность;
- 4) универсальность;
- 5) динамичность.

PHP создавался, развивался и поддерживается как язык для создания Веб-сайтов. Многие конструкции и решения в нем созданы для удобства работы в Веб-среде.

PHP перенесен на все основные операционные системы. Можно разрабатывать сайт в Windows, Mac OS X, а эксплуатировать на Linux-сервере. Сложности переноса будут минимальны и нивелироваться языком.

PHP является разработкой из мира свободного программного обеспечения, не потребуется платить ни за сам язык, ни за большинство сопутствующих программ (редакторы, Веб-серверы, базы данных). Вдобавок большинство программных продуктов, с которыми придется иметь дело, будут иметь доступный для изучения и модификации исходный код. Вложения могут потребоваться при аренде доменного имени и сервера для публикации сайта в Интернете.

Универсальность PHP делает его популярным среди хостинг-провайдеров. Любой надежный провайдер поддерживает PHP и предоставляет хостинговые услуги для сайтов на нём [7]. Как правило, хостинговые пакеты услуг предоставляют поддержку PHP без дополнительной оплаты включая либо бесплатные тарифные планы, либо дешевые с неограниченным объемом и бесплатным доменным именем.

PHP изначально создавался для разработки динамических Веб-страниц, поэтому его скрипты справляются с этой задачей лучше, чем другие языки программирования. Его код легко встраивается в HTML, что позволяет легко конвертировать статические страницы в динамические. По сути, это лучший язык для создания полноценных сайтов на основе HTML стандарта.

Недостатки:

- 1) отсутствие лидера;
- 2) непоследовательный синтаксис;
- 3) PHP уже довольно долго живущий язык;

Многие технологии и языки имеют лидера, архитектора, который определяет облик технологии, задает вектор развития, принимает решение о

том, что должно быть обязательно, а чего не будет никогда (Linux, Python, Ruby и т. п). В PHP лидера нет, многие решения и конструкции – это компромисс заинтересованных групп и исторически сложившихся реалий.

При изучении языка PHP, особенно старой части, основанной на функциях, можно заметить, что часть функций имеет префиксы `array_`, `str_`, часть не имеет [8]. Параметры функций могут быть расположены не совсем логично и не так, как в другой функции этой же группы.

Когда язык только появляется, он довольно элегантный и внутренне согласованный. По мере жизненного цикла язык обрывает дополнительными ключевыми словами, артефактами, устаревшими конструкциями, которые есть, работают, но которыми не рекомендуется пользоваться.

Ruby – это динамический, объектно-ориентированный, рефлексивный, императивный, функциональный язык программирования. Активно используется в веб-разработке, в системном администрировании и в работе операционных систем (Mac OS X, Linux, BSD).

Достоинства:

- 1) Ruby достаточно прост в освоении;
- 2) ООП в стиле Smalltalk;
- 3) поддержку комьюнити (сообщества программистов);
- 4) простой и удобный синтаксис;
- 5) поддержку новейших перспективных подходов в программировании;
- 6) большое количество готовых решений.

Недостатки:

- 1) плохая поддержка для Windows, так как изначально Ruby разрабатывался для Unix;
- 2) мало документации на русском;
- 3) потребление памяти.

Python – высокоуровневый интерпретируемый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Достоинства:

- 1) низкий порог вхождения;
- 2) кроссплатформенный;
- 3) широкое применение;
- 4) мощная поддержка компаний-гигантов it-индустрии;
- 5) в мире Python много качественных библиотек.

Синтаксис Python в сравнении с другими языками программирования более понятный для новичка.

Python отлично работает на разных платформах: Linux, Windows.

Благодаря своим возможностям и оптимизации, Python используется для разработки веб-приложений, игр, удобен для автоматизации, математических вычислений, машинного обучения, в области интернета вещей. Существует реализация под названием Micro Python, оптимизированная для запуска на микроконтроллерах.

Такие компании, как Google, Facebook, Dropbox, Spotify, Quora, Netflix, на определенных этапах разработки использовали именно Python.

Недостатки:

- 1) скорость;
- 2) динамическая типизация данных.

Python не самый быстрый среди языков программирования. скорость выполнения программ на других языках программирования может быть выше;

Из-за гибкости типов данных потребление памяти python не минимальное.

JavaScript – встраиваемый язык для программного доступа к объектам приложений [9]. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Преимущества:

- 1) гибкость;
- 2) популярность.

JavaScript предоставляет большое количество возможностей для решения самых разнообразных задач. Гибкость языка позволяет использовать множество шаблонов программирования применительно к конкретным условиям.

JavaScript открывает перед программистом немалое количество готовых библиотек, которые позволяют значительно упростить написание кода и нивелировать несовершенства синтаксиса.

Недостатки:

- 1) язык компилируется в момент исполнения кода;
- 2) отсутствует типизация данных.

Каждый раз, при открытии сайта, JavaScript код начинает компилироваться, увеличивая время выполнения программы.

Проблема всех скриптовых языков. пока выполнение кода не дойдет до нужной строчки, не исключено появление ошибки.

Ruby – написан для людей, а не для машины из-за чего страдает скорость работы, а также крайне мало документации на русском. PHP и Python хорошие языки для написания серверной части приложения, но остановимся мы все же на PHP, он идеально подходит для работы с WordPress так как данная CMS была написана на нём, также для разработки нам, понадобится серверная платформа OpenServer, у которой сразу есть поддержка PHP. Так же будем использовать Javascript для работы с анимацией и переходами на странице.

HTML – стандартизированный язык разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора [10].

CSS – это формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML) [11].

OpenServer – это портативная серверная платформа и программная среда, созданная специально для Веб-разработчиков для запуска и тестирования Веб-приложения без необходимости пользоваться интернет хостингами и установки дополнительного софта.

В состав уже входят:

- 1) Apache версии 2.4.43;
- 2) MySQL версии 8.0.19;
- 3) PHP версии 7.4.05;
- 4) PHPMyAdmin версии 5.0.2. 21.

Возможности комплекса:

- 1) незаметная работа в трее Windows;
- 2) быстрые старт и остановка;
- 3) автостарт сервера при запуске программы;
- 4) несколько режимов управления доменами;
- 5) монтирование виртуального диска;
- 6) поддержка управления через командную строку;
- 7) поддержка профилей настроек;
- 8) удобный просмотр логов всех компонентов;
- 9) переключение HTTP, MySQL и PHP модулей;

- 10) подробная и понятная документация;
- 11) доступ к доменам в один клик;
- 12) быстрый доступ к шаблонам конфигурации;
- 13) мультязычный интерфейс;
- 14) автозапуск программ по списку.

Особенности комплекса:

- 1) не требует установки (портативность);
- 2) возможность работы с USB накопителя;
- 3) одновременная работа с Denwer, Хатрр и т.д.;
- 4) работа на локальном/сетевом/внешнем IP адресе;
- 5) поддержка SSL;
- 6) создание домена путем создания обычной папки;
- 7) поддержка кириллических доменов;
- 8) поддержка алиасов (доменных указателей);
- 9) защита сервера от внешнего доступа; -
- 10) runcode конвертер доменных имён;
- 11) набор популярных сторонних расширений PHP;
- 12) планировщик заданий (cron);
- 13) создание локального поддомена без потери видимости основного домена в сети интернет.

PHPMyAdmin – Веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой Веб-интерфейс для администрирования СУБД MySQL [12]. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у Веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд. Приложение

распространяется под лицензией GNU General Public License и поэтому многие другие разработчики интегрируют его в свои разработки, например XAMPP, Denwer, AppServ, Open Server.

Главные преимущества:

- 1) поддержка большинства функций MySQL;
- 2) просмотр и удаление баз данных, таблиц, вьюшек, полей и индексов;
- 3) создание, копирование, удаление, переименование и изменение баз данных, таблиц, полей и индексов;
- 4) управление сервером, базами данных и таблицами, с советами по настройке сервера;
- 5) выполнение, редакция и сохранение любого SQL-выражения, включая пакетные запросы;
- 6) управление пользователями MySQL и их привилегиями;
- 7) работа с хранимыми процедурами и триггерами;
- 8) поддержка импорта данных из CSV и SQL;
- 9) поддержка экспорта в различные форматы CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument текст и таблицы, Word, Excel, LATEX и другие – администрирование нескольких серверов;
- 10) генерирование наглядных схем баз данных в виде PDF;
- 11) создание комплексных запросов с помощью функции Запрос по шаблону;
- 12) глобальный или частичный поиск в базе данных.

WordPress – CMS с открытым исходным кодом; для написания так же использует язык программирования PHP; в качестве сервера базы данных использует – MySQL; выпущена под лицензией GNU GPL rev. 2. Данная CMS изначально создавалась, как CMS для создания исключительно интернет-блогов, но с прошествии временем многофункциональные возможности

существенно увеличились, позволив системе получить инструментарий для создания интернет-магазинов, новостных веб-сайтов и прочих веб-сайтов хоть какой трудности и сложности [13]. WordPress дает возможность с легкостью увеличивать функциональность системы с помощью готовых модулей, а также расширений, которые можно создавать самостоятельно.

Главные плюсы системы:

- 1) бесплатная лицензия с постоянными обновлениями;
- 2) неплохая документация и служба поддержки на русском языке;
- 3) легкость инсталляции системы;
- 4) удобный визуальный редактор;
- 5) возможность создания собственных тем;
- 6) доступно большое количество бесплатных тем и плагинов;
- 7) возможность разработки новых и редактирования уже имеющихся расширений и плагинов;
- 8) создание, управление правами пользователя.

WooCommerce – плагин, который используется для работы в CMS WordPress. В настоящее время, данный плагин является одним и наилучшего решение для юзеров, которым необходимо разработать интернет-магазин на CMS WordPress [14]. Количество загрузок плагина Woocommerce превышает один миллион. Плагин WooCommerce имеет очень хорошую гибкость и простоту использования, практически все юзеры имеют возможность с легкостью и без 24 труда понять процесс его работе, ещё один плюс — это огромное количество плагинов-дополнений.

Рассмотрим возможности данного плагина:

- 1) реализация абсолютно любых товаров и услуг;
- 2) гибкая и удобная настройка доставки товаров и услуг;
- 3) обилие способов оплаты товаров и услуг;

4) абсолютный контроль.

Для реализации абсолютно любых товаров и услуг используется плагин WooCommerce. Появляется возможность реализовывать физические, а также электронные товары имеющие различные формы и размеры, представлять варианты товаров и услуг, сложные конфигурации.

Плагин предоставляет абсолютное контролирование интернет-магазина, начиная расчётами налоговых ставок и количества запасов на складе заканчивая возможностью управлять аккаунтами юзеров. даёт возможность устанавливать и деинсталлировать расширения, изменять оформление и настройки, как необходимо для функционирования интернет-магазина.

WooCommerce предоставляет гибкие настройки способов доставки товаров и услуг. Есть множество вариантов: бесплатная доставка, доставку за определенную стоимость и тому подобные варианты. Есть право сделать ограничение для доставки в те или иные страны.

Плагин комплектуется возможностью использовать в качестве оплаты обширный список банковских карт.

Плагин предоставляет абсолютное контролирование интернет-магазина, начиная расчётами налоговых ставок и количества запасов на складе заканчивая возможностью управлять аккаунтами юзеров. даёт возможность устанавливать и деинсталлировать расширения, изменять оформление и настройки, как необходимо для функционирования интернет-магазина.

Все вышеперечисленные платформы являются одними из самых популярных решений для разработки в последнее время, это обеспечивается их гибкостью, простотой и свободной лицензией. Так же во время разработки продукт будет проходить тестирование в браузере Google Chrome так как он основан на самом популярном движке chromium на котором написано

большинство современных браузеров (Яндекс Браузер, Opera, Microsoft Edge, Epic Browser, Comodo Dragon и др.)

Перечислим все инструменты, которыми нужно будет пользоваться в процессе разработки:

- 1) HTML;
- 2) PHP;
- 3) CSS;
- 4) JavaScript;
- 5) OpenServer;
- 6) MySQL;
- 7) PHPMysqlAdmin;
- 8) Google Chrome;
- 9) CMS WordPress (WooCommerce).

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Определим основные требования реализации в итоговом проекте для выполнения поставленной цели.

2.1 Функциональные требования

В проекте реализованы следующие подсистемы:

- 1) система взаимодействия с личным кабинетом;
- 2) система добавления товара в корзину;
- 3) система оформления заказа;
- 4) система авторизации и регистрации пользователя;
- 5) система создания отзывов и выставления рейтинга в карточке товара;
- 6) система администрирования приложения.

2.1.1 Требования к функционалу подсистемы авторизации и регистрации пользователя

Подсистема имеет следующий функционал:

- 1) зарегистрировать пользователя в системе;
- 2) авторизовать пользователя в системе.

Подсистема имеет связь с хранилищем информации о пользователях.

Входные данные при регистрации:

- 1) имя пользователя (необязательно настоящее);
- 2) e-mail пользователя;
- 3) пароль для входа в систему.

Выходные данные: пользователь регистрируется в приложении.

Входные данные при авторизации:

- 1) имя пользователя;
- 2) пароль для входа в систему.

Выходные данные: пользователь авторизовывается в приложении.

2.1.2 Требования к функционалу подсистемы взаимодействия с личным кабинетом

Подсистема имеет следующий функционал:

- 1) изменение пароля;
- 2) просмотр статуса заказа;
- 3) просмотр истории заказов;
- 4) добавление адреса доставки.

Подсистема имеет связь с:

- 1) хранилищем информации о пользователях;
- 2) хранилищем информации о всех отзывах;
- 3) подсистемой регистрации и авторизации, поскольку пользователю сначала нужно авторизоваться, прежде чем перейти в личный кабинет.

Входные данные:

- 1) пароль пользователя;
- 2) имя пользователя.

Выходные данные: добавленная/измененная в хранилище данных информация о пользователе.

2.1.3 Требования к функционалу подсистемы добавления товара в корзину

Подсистема имеет следующий функционал:

1) добавить товар в корзину.

Подсистема имеет связь с:

1) хранилищем информации о товарах;

Входные данные при добавлении товара в корзину:

1) выбор цвета из вариации доступных (если имеется);

2) выбор размера из вариации доступных (если имеется).

Выходные данные: добавлен товар в корзину с выбранными атрибутами.

2.1.4 Требования к функционалу подсистемы оформления заказа

Подсистема имеет следующий функционал:

1) оформить заказ.

Подсистема имеет связь с:

1) хранилищем информации о пользователях;

2) хранилищем информации о товарах;

3) хранилищем информации о купонах;

4) хранилищем информации о заказах;

5) хранилищем информации о странах;

6) хранилищем информации о доставке;

7) хранилищем информации об оплате.

Входные данные при оформлении заказа:

1) имя;

2) фамилия;

- 3) адрес;
- 4) телефон;
- 5) e-mail;
- 6) вариант оплаты;
- 7) вариант доставки.

Выходные данные оформления заказа: новая позиция в хранилище информации о заказах.

2.1.5 Требования к функционалу подсистемы создания отзывов и выставления рейтинга в карточке товара

Подсистема имеет следующий функционал:

- 1) просмотр списка написанных ранее другими пользователями отзывов;
- 2) предоставить форму для написания отзыва;
- 3) отправить отзыв.

Подсистема имеет связь с:

- 1) хранилищем информации о пользователях;
- 2) хранилище информации о всех отзывах;
- 3) подсистемой регистрации и авторизации.

Входные данные:

- 1) имя пользователя;
- 2) отзыв;
- 3) рейтинг, выставленный пользователем.

Выходные данные: отзыв о товаре и рейтинг, добавленные в хранилище информации о всех отзывах.

2.1.6 Требования к функционалу подсистемы администрирования приложения

Подсистема должна иметь следующий функционал:

- 1) администрирование каталога товаров;
 - а) редактировать информацию о товаре;
 - б) добавить товар;
 - в) удалить товар;
 - г) добавить категорию товара;
 - д) удалить категорию товара;
 - е) удалить атрибут;
 - ж) добавить атрибут;
- 2) администрирование отзывов;
 - а) просмотр отзывов;
 - б) удалить отзыв;
- 3) администрирование скидочных купонов;
 - а) выставить процент скидки;
 - б) указать временные рамки действия купона;
 - в) указать возможное количество использования купона для 1 пользователя;
 - г) удалить купон;
 - д) добавить купон;
- 4) администрирование наличия товаров:
 - а) изменить статус наличия;
 - б) указать точное количество оставшихся товаров.

Подсистема имеет связь со всеми хранилищами данных.

Входные данные для редактирования: информация о товарах, категории товаров, купоны, отзывы, информация о наличии товара.

Входные данные для удаления:

- 1) определенный товар;
- 2) категория товара;
- 3) отзыв;
- 4) купон;
- 5) атрибут товара.

Входные данные для добавления:

- 1) товар;
- 2) купон;
- 3) категория товара.

2.2 Нефункциональные требования

Определены следующие нефункциональные требования для разрабатываемой системы:

- 1) требования к пользователям;
- 2) требования к объёму данных.

2.2.1 Требования к пользователям

Каждый пользователь, желающий иметь собственный кабинет на сайте, должен зарегистрироваться в системе и заполнить следующие обязательные поля в регистрационной форме:

- 1) имя пользователя;
- 2) e-mail пользователя;

3) пароль для входа в систему.

В системе должны быть выделены следующие типы пользователей:

- 1) неавторизованный пользователь;
- 2) авторизованный пользователь;
- 3) администратор.

Неавторизованный пользователь имеет возможность пользоваться всем функционалом сайта, кроме личного кабинета. Авторизованный пользователь должен иметь возможность воспользоваться любым функционалом системы, за исключением функций администрирования.

Администратор должен иметь возможность воспользоваться функционалом админ-панели.

2.2.2 Требования к объёму данных

Всего в системе имеются следующие хранилища данных:

- 1) хранилище информации о пользователях;
- 2) хранилище информации о заказах;
- 3) хранилище информации о купонах;
- 4) хранилище информации о товарах;
- 5) хранилище информации о странах;
- 6) хранилище информации об отзывах;
- 7) хранилище информации об оплате;
- 8) хранилище информации о доставке;
- 9) хранилище информации о категориях товаров.

3 ПРОЕКТИРОВАНИЕ

3.1 Архитектура предлагаемого решения

Разработанная архитектура интернет-магазина, который будет разработан с использованием CMS WordPress будет выглядеть как показано на рисунке 9 и будет рассматриваться как взаимосвязь операционной части т.е. среды разработки CMS WORDPRESS и серверной части.

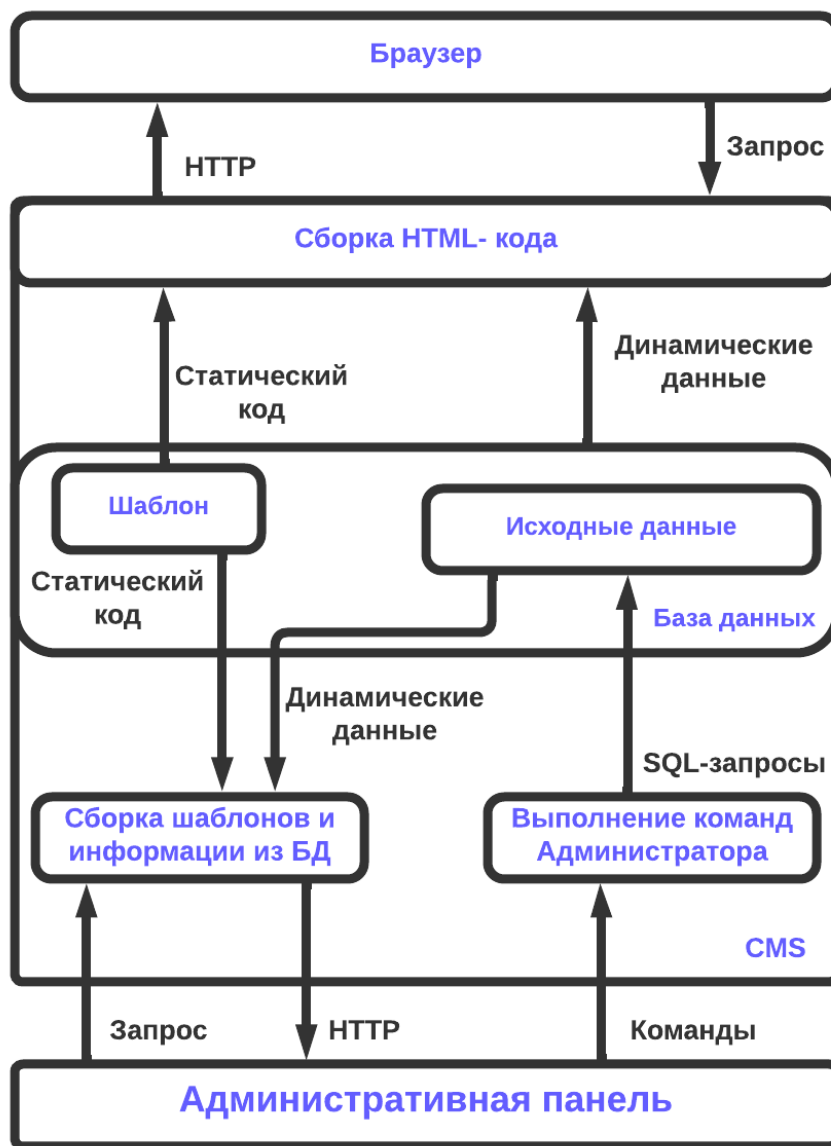


Рисунок 9 – Архитектура интернет-магазина на CMS

3.2 Интерфейс пользователя

Интерфейс пользователя представлен в виде языка разметки HTML, языка стилей CSS, когда пользователь совершает какое-нибудь действие, то оно обрабатывается либо внутри интерфейса, как например переход по страницам сайта, так и отправляется запрос на сервер, в случае если пользователю, посредством отправки формы, надо получить результат от БД (рисунок 10). Также есть JavaScript, который не входит в интерфейс пользователя, но он все же отвечает за некоторые анимации, в частности за открытие модальных окон.



Рисунок 10 – Интерфейс пользователя

3.3 Серверная часть

Серверная часть реализуется на языке PHP с вспомогательным JavaScript. С интерфейса пользователя приходит запрос, сервер его обрабатывает, если что-то неверно, возвращает его с ошибкой, если всё верно отправляет запрос в БД. БД получает запрос и выдаёт серверу ответ, сервер получает ответ и отправляет его в виде JSON интерфейсу пользователя (рисунок 11).

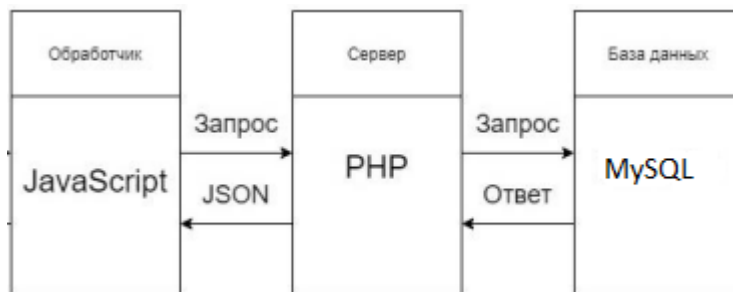


Рисунок 11 – Серверная часть

3.4 Моделирование базы данных MySQL

В системе управления контентом CSM WordPress применяется СУБД MySQL. База данных состоит из 8 основных таблиц (рисунок 12):

- 1) покупатели;
- 2) категории;
- 3) страны;
- 4) оплата;
- 5) заказы;
- 6) товар;
- 7) доставка;
- 8) скидки (купоны);
- 9) отзывы.

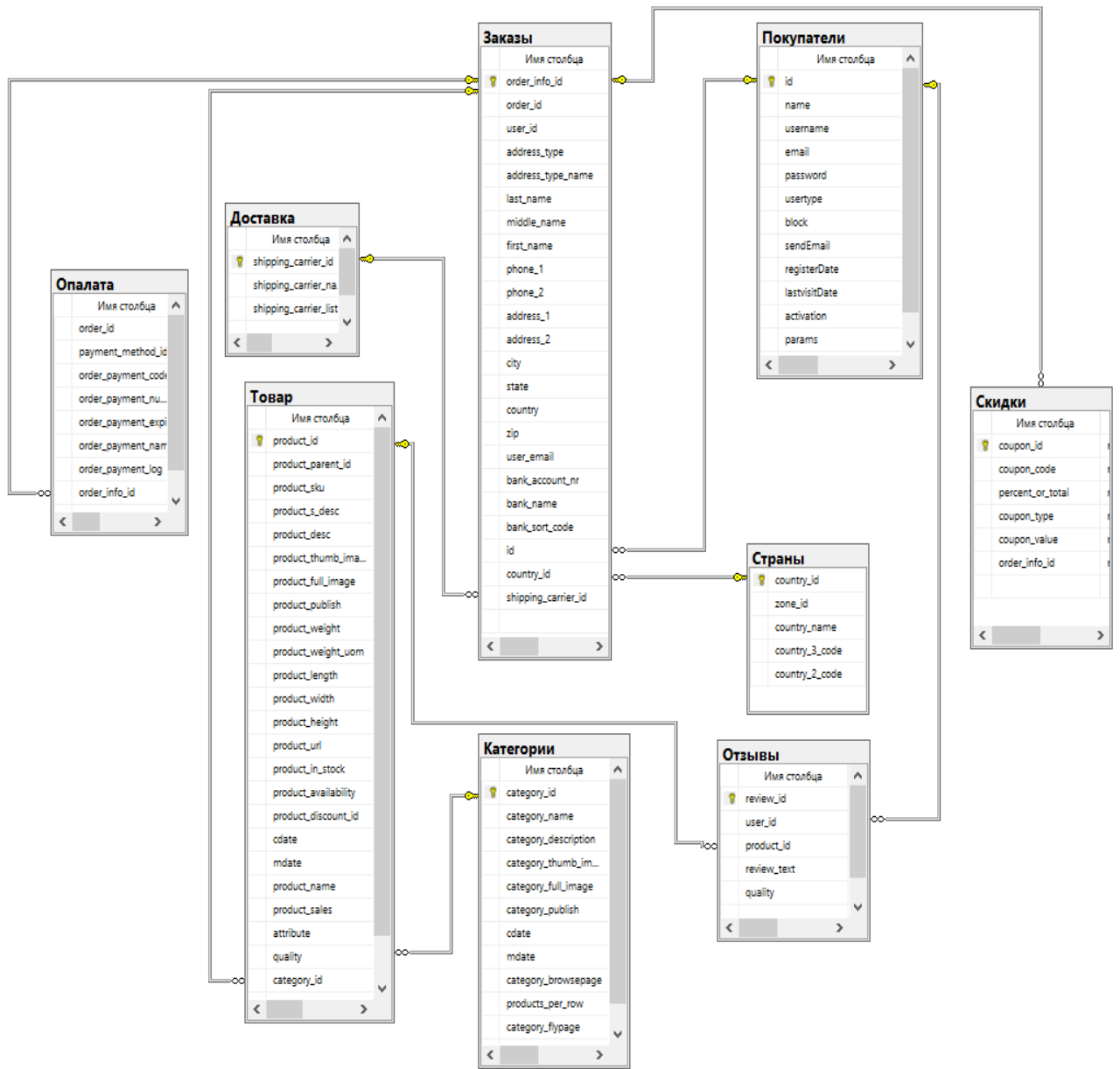


Рисунок 12 – Диаграмма базы данных

Наименование, перечень, тип поля, длина поля для каждой таблицы БД приведены в таблицах 1-9.

Таблица 1 – Покупатели

| № | Название | Идентификатор поля | Тип данных |
|-----|---------------------------|--------------------|--------------|
| 1. | Код | id | int(11) |
| 2. | ФИО | name | varchar(255) |
| 3. | Имя пользователя | username | varchar(150) |
| 4. | Электронная почта | email | varchar(100) |
| 5. | Пароль | password | varchar(100) |
| 6. | Тип покупателя | usertype | varchar(25) |
| 7. | Флаг блокировки | block | tinyint(4) |
| 8. | Подтверждение регистрации | sendEmail | tinyint(4) |
| 9. | Дата регистрации | registerDate | datetime |
| 10. | Дата последнего визита | lastvisitDate | datetime |
| 11. | Флаг активации | activation | varchar(100) |
| 12. | Параметры | params | text |

Таблица 2 – Категории

| № | Название | Идентификатор поля | Тип данных |
|----|---------------------------|----------------------|--------------|
| 1. | Код категории | category_id | int(11) |
| 2. | Наименование | category_name | varchar(128) |
| 3. | Описание | category_description | text |
| 4. | Малое изображение | category_thumb_image | varchar(255) |
| 5. | Полное изображение | category_full_image | varchar(255) |
| 6. | Порядок публикации | category_publish | char(1) |
| 7. | Дата добавления | cdate | int(11) |
| 8. | Дата последнего изменения | mdate | int(11) |

Продолжение таблицы 2

| | | | |
|-----|--------------------|---------------------|--------------|
| 9. | Обзор категории | category_browsepage | varchar(255) |
| 10. | Строки продукции | products_per_row | tinyint(2) |
| 11. | Страница категории | category_flypage | varchar(255) |

Таблица 3 – Страны

| № | Название | Идентификатор поля | Тип данных |
|----|--------------|--------------------|-------------|
| 1. | Код | country_id | int(11) |
| 2. | Код зоны | zone_id | int(11) |
| 3. | Наименование | country_name | varchar(64) |
| 4. | Код 2 | country_3_code | char(3) |
| 5. | Код 3 | country_2_code | char(2) |

Таблица 4 – Скидки (купоны)

| № | Название | Идентификатор поля | Тип данных |
|----|-----------------|--------------------|------------------------------|
| 1. | Код | coupon_id | int(16) |
| 2. | Код скидки | coupon_code | varchar(32) |
| 3. | Общее значение | percent_or_total | enum('percent', 'total') |
| 4. | Тип | coupon_type | enum('gift', 'permanent') |
| 5. | Значение купона | coupon_value | decimal(12,2) |

Таблица 5 – Оплата

| № | Название | Идентификатор поля | Тип данных |
|----|----------------------|----------------------|--------------|
| 1. | Код | order_id | int(11) |
| 2. | Метод оплаты | payment_method_id | int(11) |
| 3. | Код оплаты | order_payment_code | varchar(30) |
| 4. | Номер способа оплаты | order_payment_number | blob |
| 5. | Срок окончания | order_payment_expire | int(11) |
| 6. | Наименование | order_payment_name | varchar(255) |
| 7. | Лог | order_payment_log | text |

Таблица 6 – Заказы

| № | Название | Идентификатор поля | Тип данных |
|-----|---------------------|--------------------|-------------|
| 1. | Код | order_info_id | int(11) |
| 2. | Код заказа | order_id | int(11) |
| 3. | Код пользователя | user_id | int(11) |
| 4. | Тип адреса | address_type | char(2) |
| 5. | Наименование адреса | address_type_name | varchar(32) |
| 6. | Фамилия | last_name | varchar(32) |
| 7. | Отчество | middle_name | varchar(32) |
| 8. | Имя | first_name | varchar(32) |
| 9. | Телефон 1 | phone_1 | varchar(32) |
| 10. | Телефон 2 | phone_2 | varchar(32) |
| 11. | Адрес 1 | address_1 | varchar(32) |
| 12. | Адрес 2 | address_2 | varchar(32) |

Продолжение таблицы 6

| | | | |
|-----|----------------------|-----------------|-------------|
| 13. | Город | city | varchar(32) |
| 14. | Область | state | varchar(32) |
| 15. | Страна | country | varchar(32) |
| 16. | Индекс | zip | varchar(32) |
| 17. | Электронная почта | user_email | varchar(32) |
| 18. | Банковские реквизиты | bank_account_nr | varchar(32) |
| 19. | Наименование банка | bank_name | varchar(32) |
| 20. | Код банка | bank_sort_code | varchar(32) |

Таблица 7 – Товар

| № | Название | Идентификатор поля | Тип данных |
|----------|--------------------|---------------------------|-------------------|
| 1. | Код | product_id | int(11) |
| 2. | Код описания | product_parent_id | int(11) |
| 3. | Тип товара | product_sku | varchar(64) |
| 4. | Количество товара | product_s_desc | varchar(255) |
| 5. | Описание товара | product_desc | text |
| 6. | Полное изображение | product_thumb_image | varchar(255) |
| 7. | Малое изображения | product_full_image | varchar(255) |
| 8. | Порядок публикации | product_publish | char(1) |
| 9. | Вес | product_weight | decimal(10,4) |
| 10. | Нетто | product_weight_uom | varchar(32) |
| 11. | Длина | product_length | decimal(10,4) |
| 12. | Ширина | product_width | decimal(10,4) |
| 13. | Высота | product_height | decimal(10,4) |
| 14. | Ссылка | product_url | varchar(255) |

Продолжение таблицы 7

| | | | |
|-----|---------------------|----------------------|-------------|
| 15. | Наличие товара | product_in_stock | int(11) |
| 16. | Признак доступности | product_availability | varchar(56) |
| 17. | Скидка | product_discount_id | int(11) |
| 18. | Дата добавления | cdate | datetime |
| 19. | Дата обновления | mdate | datetime |
| 20. | Наименование | product_name | varchar(64) |
| 21. | Стоимость | product_sales | int(11) |
| 22. | Атрибуты | attribute | varchar(64) |
| 23. | Рейтинг | quality | varchar(14) |

Таблица 8 – Доставка

| № | Название | Идентификатор поля | Тип данных |
|----|--------------------|-----------------------------|------------|
| 1. | Код | shipping_carrier_id | int(11) |
| 2. | Наименование | shipping_carrier_name | char(80) |
| 3. | Порядок публикации | shipping_carrier_list_order | int(11) |

Таблица 9 – Отзывы

| № | Название | Идентификатор поля | Тип данных |
|----|------------------|--------------------|-------------|
| 1. | Код | review_id | int(11) |
| 2. | Код пользователя | user_id | int(11) |
| 3. | Код товар | product_id | int(11) |
| 4. | Текст | review_text | text |
| 5. | Рейтинг | quality | varchar(14) |

4 РЕАЛИЗАЦИЯ

4.1 Главная страница

На рисунках 13 – 18 представлена главная страница сайта. Код реализации представлен в листингах Б.1 – Б.5.

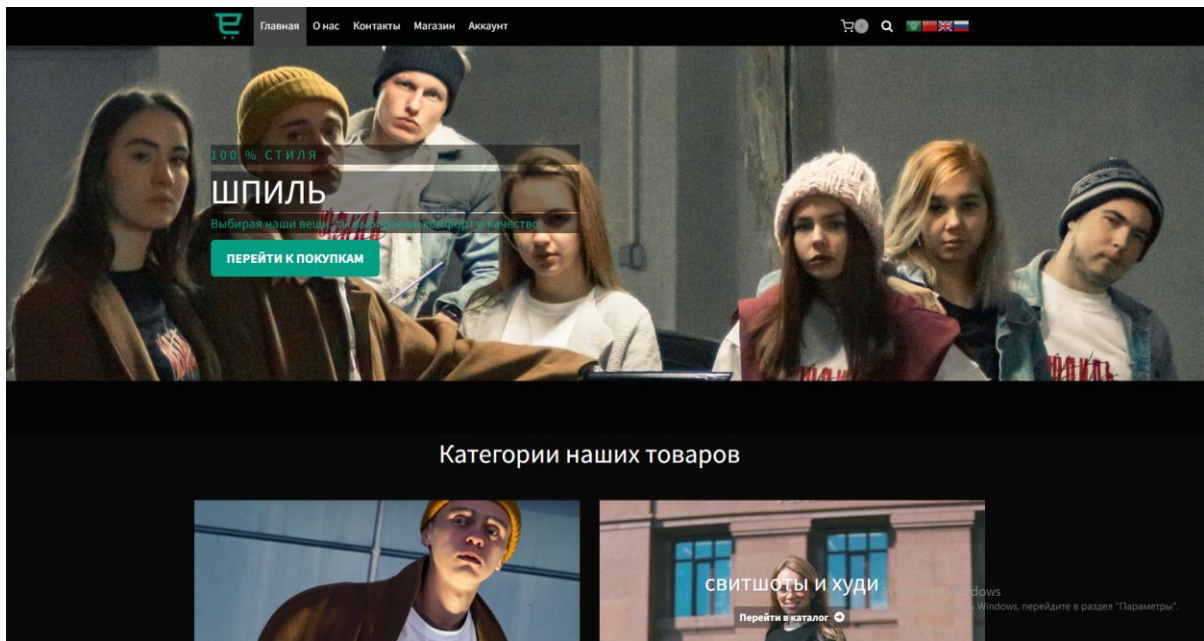


Рисунок 13 – Главная страница сайта часть 1

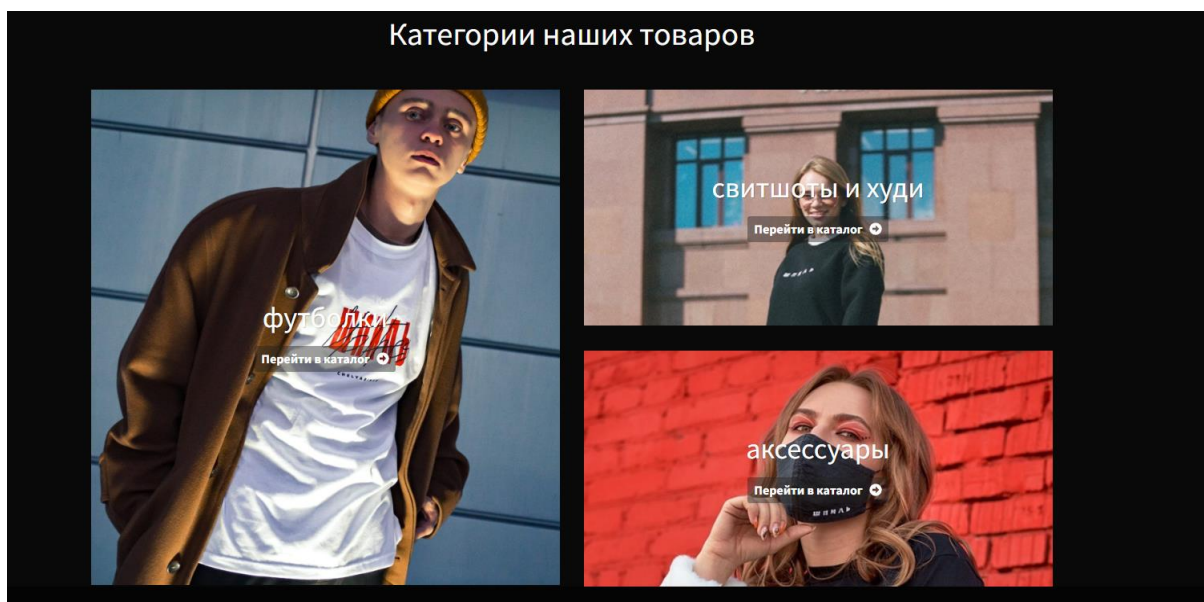


Рисунок 14 – Главная страница сайта часть 2

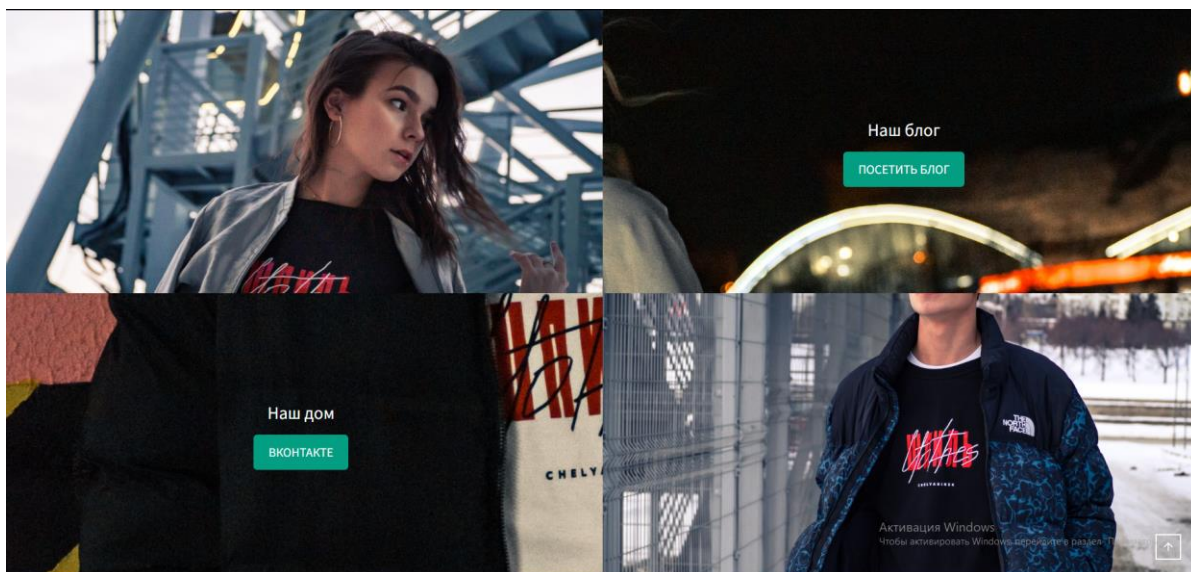


Рисунок 15 – Главная страница сайта часть 3

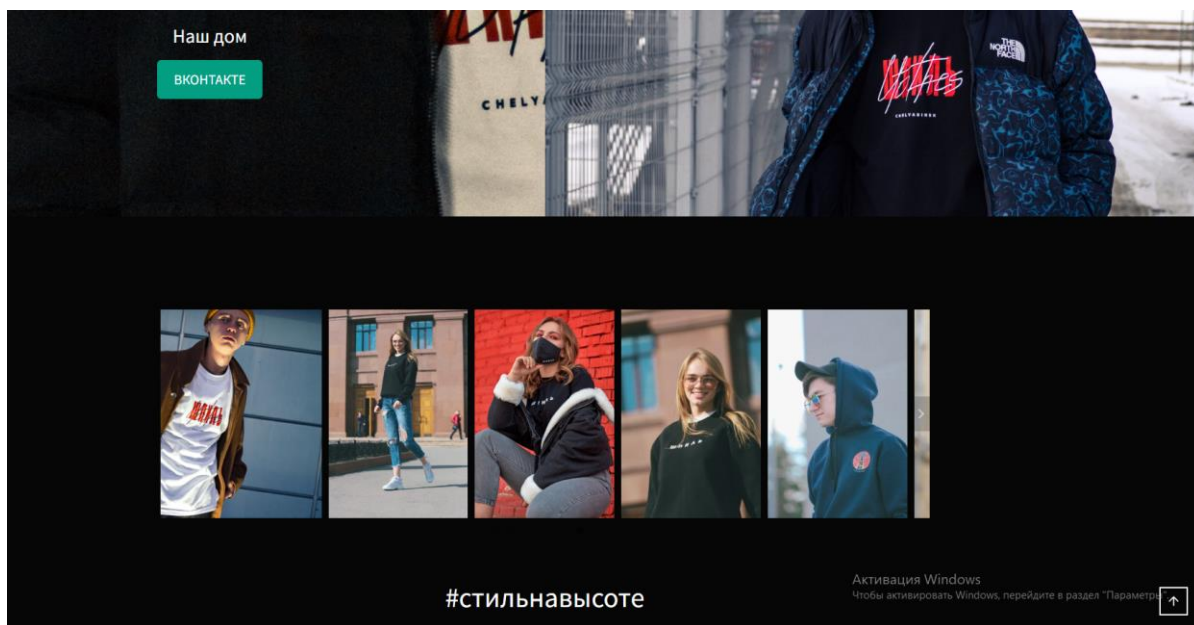


Рисунок 16 – Главная страница сайта часть 4

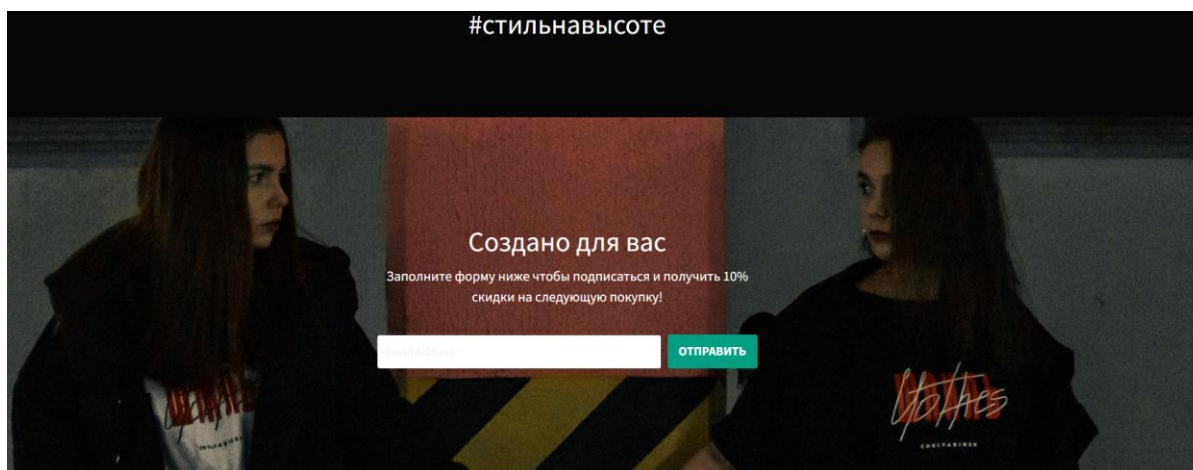


Рисунок 17 – Главная страница сайта часть 5

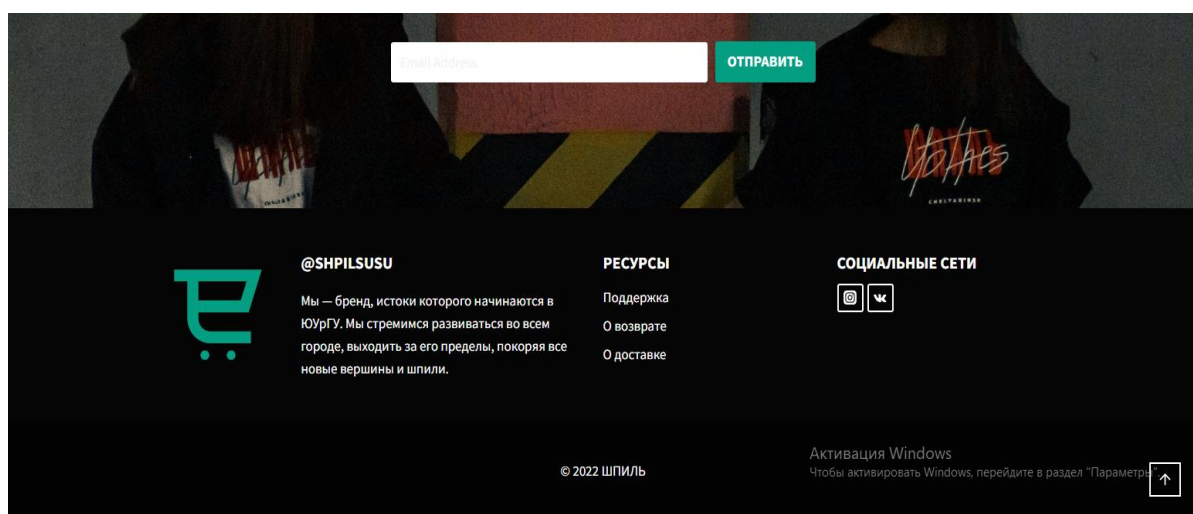


Рисунок 18 – Главная страница сайта часть 6

Далее будет подробно рассмотрен каждый элемент главной страницы.

Как и полагается первым, что видит пользователь, попав на главную страницу, шапку сайта (рисунок 19).



Рисунок 19 – Шапка сайта

Шапка сайта поделена на две части. В левой половине располагается меню навигации с пятью ссылками: «Главная», «О нас», «Контакты», «Магазин», «Аккаунт» (рисунок 20). Каждая при нажатии на нее, перенаправляет пользователя на новую страницу. Далее каждая страница из меню будет рассмотрена подробно. В правой половине находятся иконки-

ссылки на корзину, форму поиска, переключатели языка страницы, соответствующие флагам государств (арабский, китайский, английский, русский) (рисунок 21).



Рисунок 20 – Шапка сайта, меню навигации



Рисунок 21 – Шапка сайта, правая часть

Следом на странице располагается блочный элемент, содержащий в себе три фото-обложки с расположенными на них кнопками, перенаправляющими пользователя при нажатии на них, на страницы с соответствующими категориями товаров (рисунок 22).

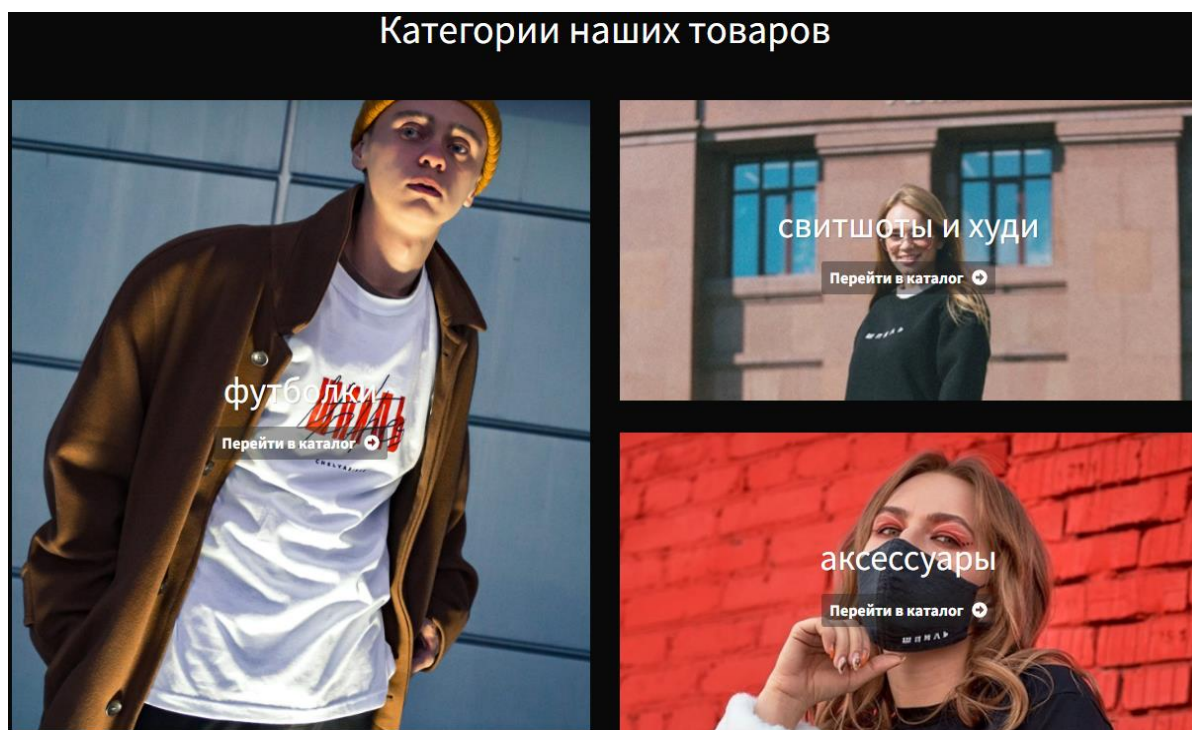


Рисунок 22 – Фото-обложки с расположенными на них кнопками

Далее при прокрутке вниз пользователь видит новый блочный элемент, а именно фотоколлаж, содержащий в себе кнопки, перенаправляющие

пользователя при нажатии на них соответственно на страницу бренда в группе во «Вконтакте» и на страницу блога (рисунок 23).

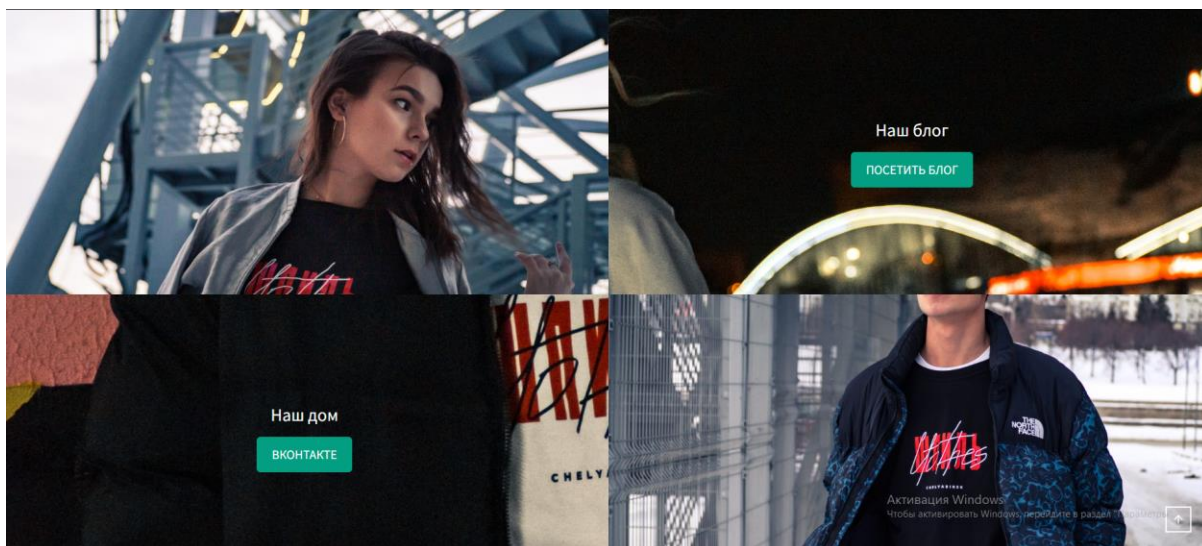


Рисунок 23 – Фотоколлаж с кнопками

Под фотоколлажем находится слайдер-галерея, с автоматической прокруткой фото, при нажатии на которое пользователь переходит на страницу «инстаграм» бренда (рисунок 24).

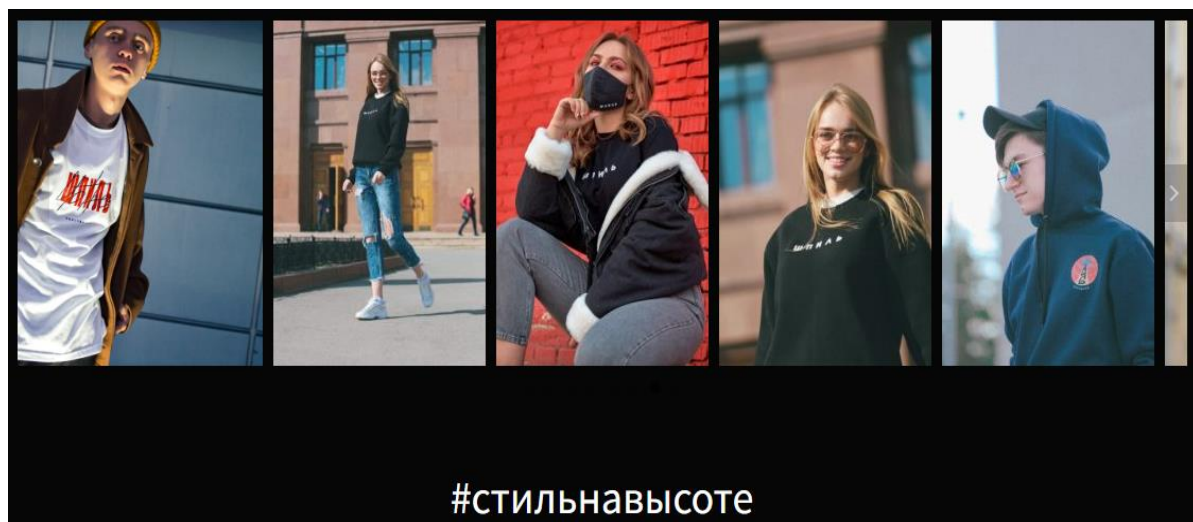


Рисунок 24 – Автоматический слайдер-галерея

После слайдер-галереи идёт фото с эффектом параллакса, на нём расположена форма, при заполнении которой, пользователь регистрируется и получает скидочный купон на первую покупку 10% (рисунок 25)

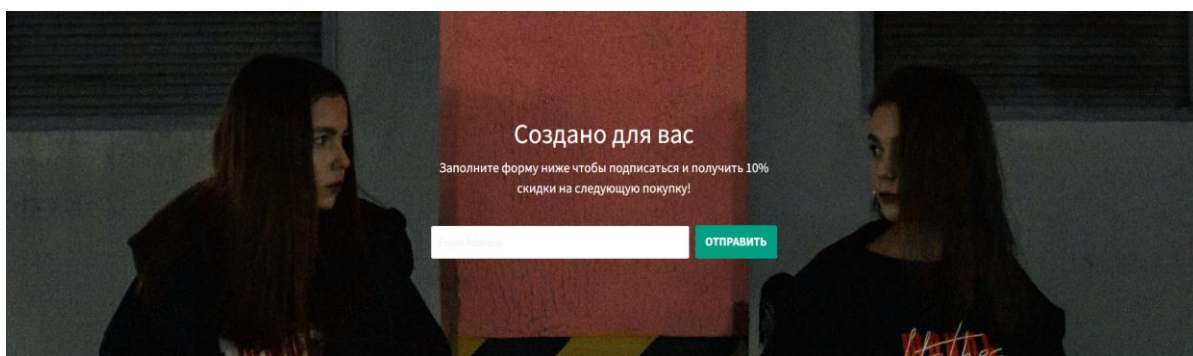


Рисунок 25 – Фото с эффектом параллакса с формой для получения скидочного купона

В подвале сайта располагается логотип, информация о бренде, вертикальное меню, содержащее в себе три ссылки на страницы: «Поддержка», «О возврате», «О доставке». В правой части иконки-ссылки на социальные сети бренда (рисунок 26).

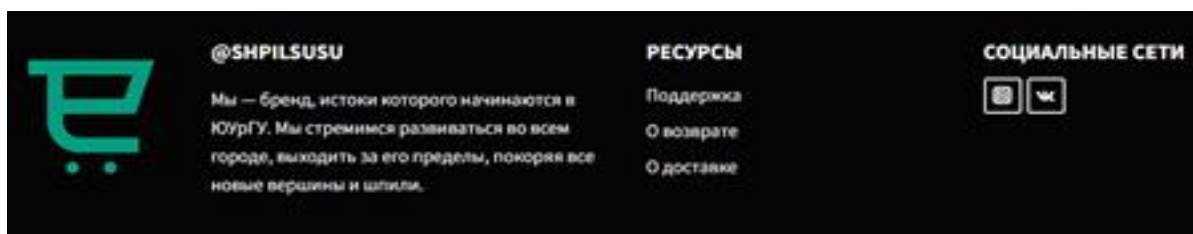


Рисунок 26 – Подвал сайта

Следом за главной страницей в меню навигации идет страница «О нас». На ней последовательно расположена интересная информация, повествующая от истории бренда (рисунок 27). Информация о команде, с ссылками на социальные сети (рисунок 28). в самом низу перед «футером» располагается фото-обложка с кнопкой, перенаправляющей пользователя на страницу с формой для обратной связи (рисунок 29).

4.2 Страница «О нас»

На рисунках 27 – 29 представлена страница «О нас».

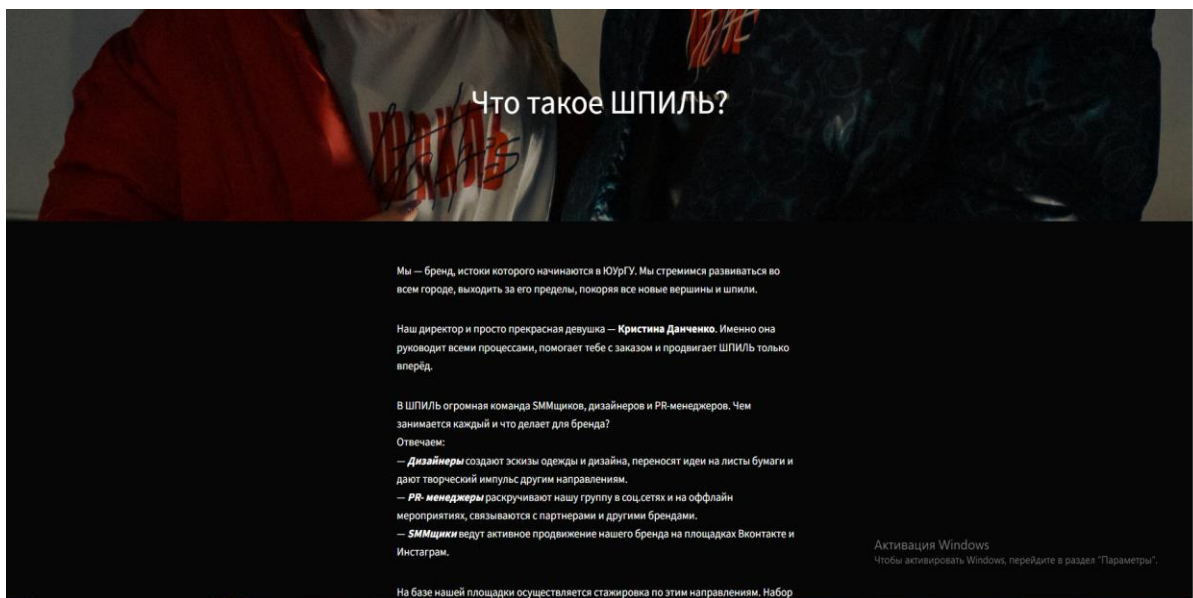


Рисунок 27 – «О нас» часть 1

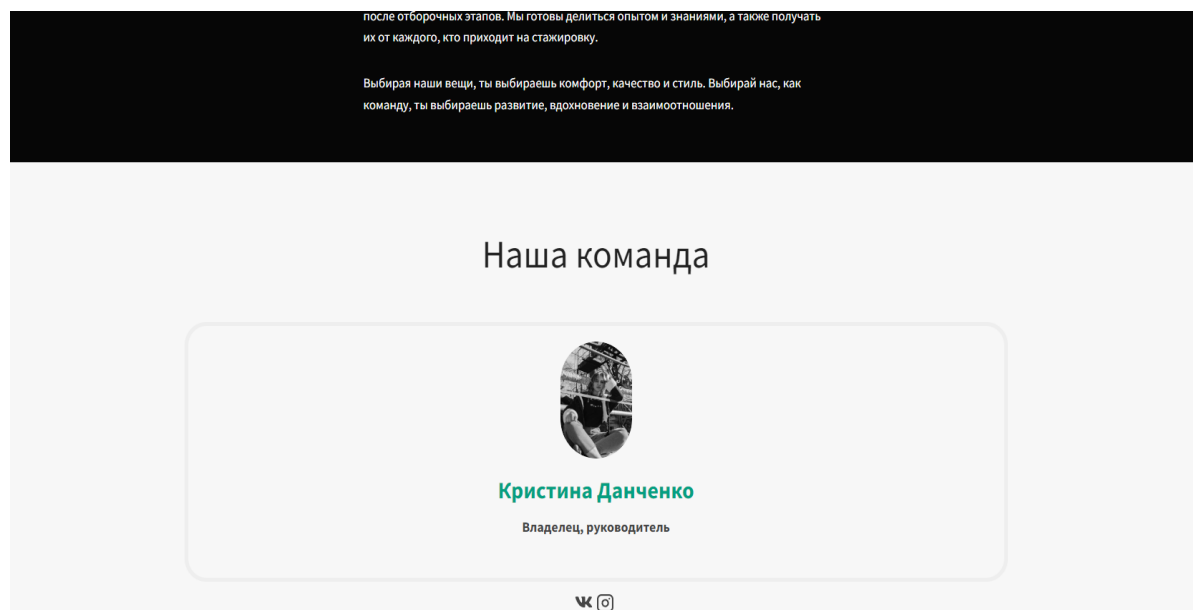


Рисунок 28 – «О нас» часть 2

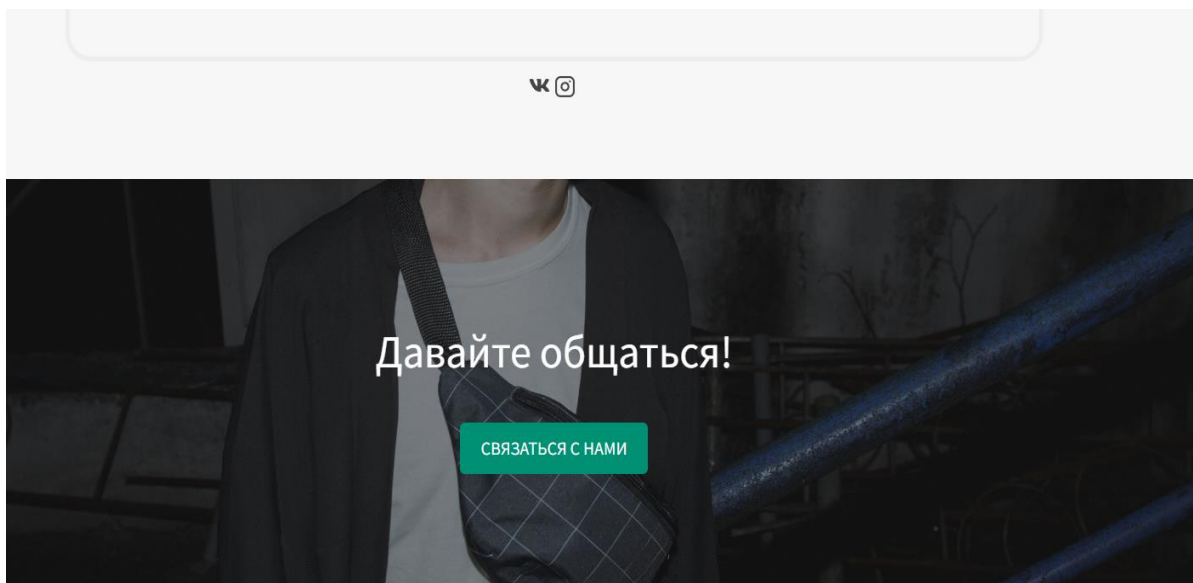


Рисунок 29 – «О нас» часть 3

4.3 Страница «Контакты»

Следом за страницей «О нас», идет страница «Контакты» (рисунок 30). В ней пользователя встречает форма для обратной связи (рисунок 31).

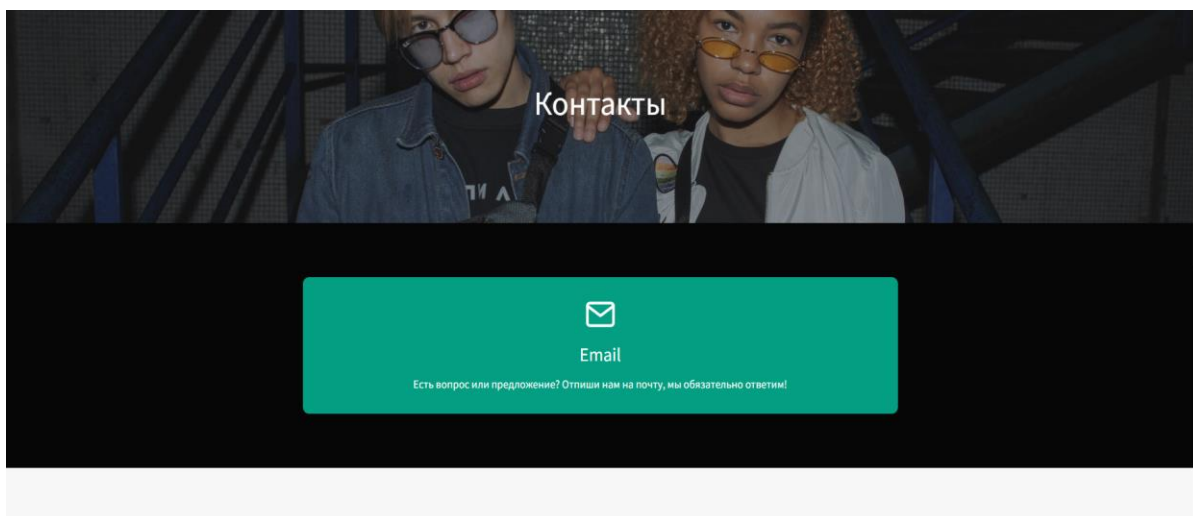


Рисунок 30 – «Контакты» часть 1

Напиши нам

Имя

Email адрес

Напишите здесь свой вопрос или предложение

ОТПРАВИТЬ

Рисунок 31 – «Контакты», форма для обратной связи

Страницу магазин представляет собой многостраничный каталог с фильтрами и боковой панелью категорий (рисунок 32).

4.4 Каталог товаров

На рисунке 32 представлена страница с каталогом товаров. Код функционала представлен в листингах Б.6 – Б.7.

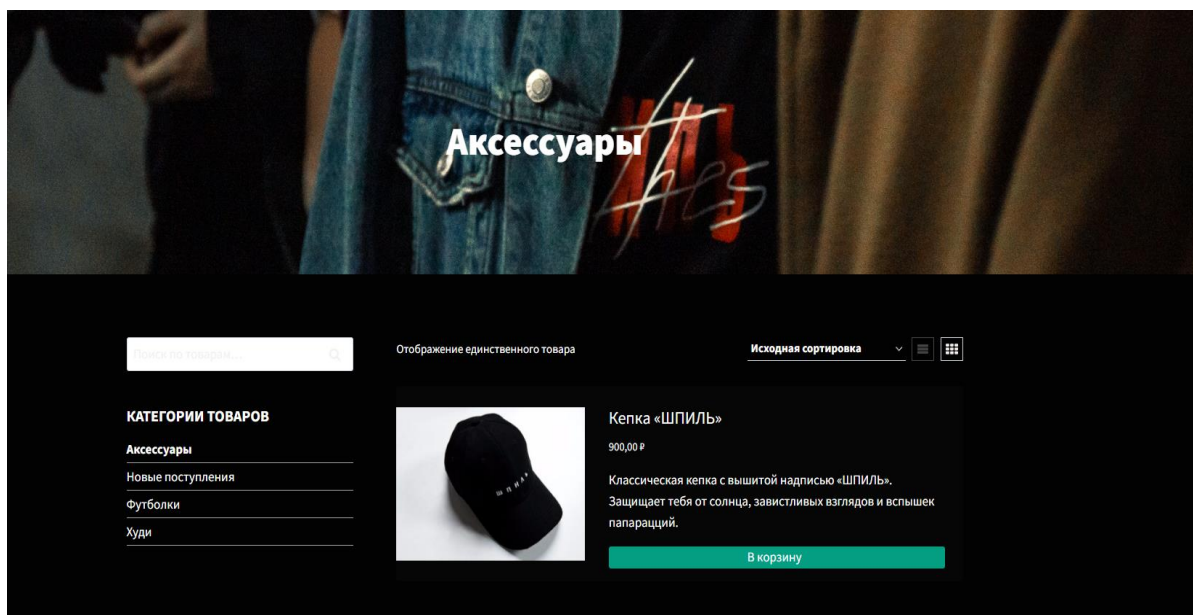


Рисунок 32 – «Магазин»

В правой части экрана расположена форма поиска и боковая панель с категориями товаров (рисунок 33).

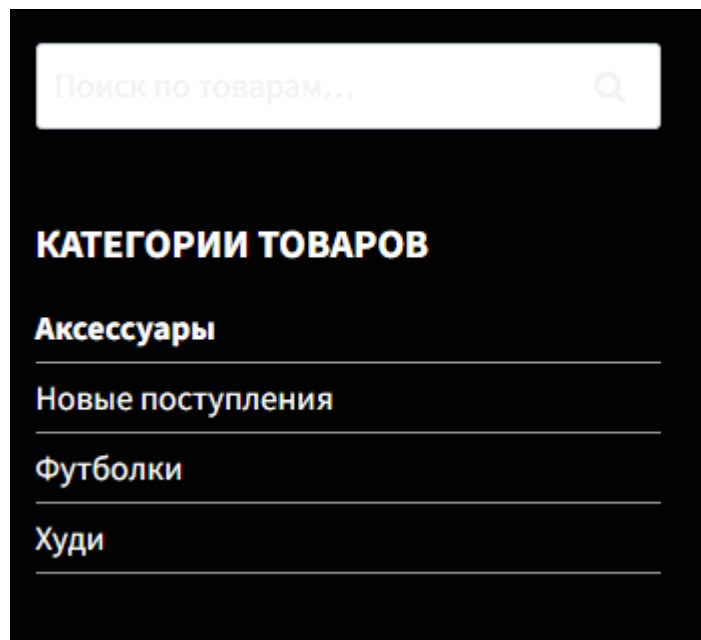


Рисунок 33 – «Магазин», форма поиска и боковая панель категорий

Справа сверху располагается выпадающее меню сортировки товара по выбранному свойству (рисунок 34).

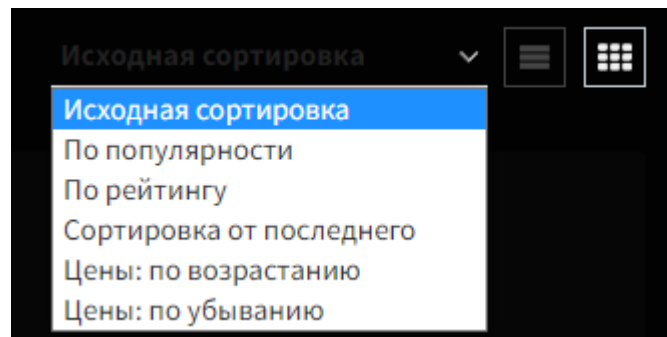


Рисунок 34 – «Магазин», выпадающее меню сортировки товара

Кликнув по карточке товара, пользователь переходит на его страницу (рисунок 35). Так как у кепки нет размерной и цветовой вариации, то здесь нет функции соответствующего выбора. Есть возможность выбрать количество товара, добавить товар в корзину, пролистать галерею фото, прочитать описание, детали, написать отзыв и поставить оценку, но для этого нужно быть авторизованным пользователем.

4.5 Карточка товара

На рисунке 35 изображена карточка товара. Код функционала представлен в листингах Б.8 – Б.9.

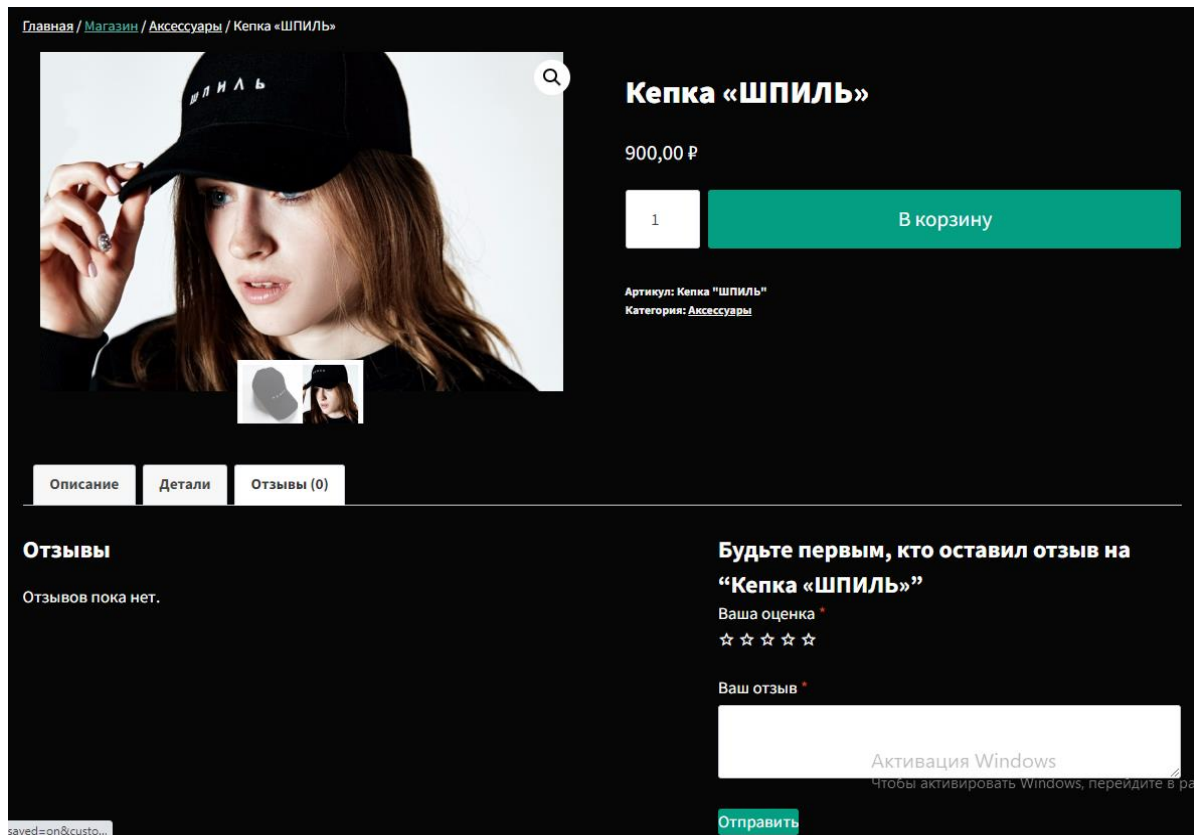


Рисунок 35 – «Магазин», страница товара

Крайней ссылкой в меню навигации является личная страница пользователя «Аккаунт» (рисунок 36). Справа находится виджет-меню позволяющий переключать страницы личного кабинета, на странице «Заказы» пользователь может увидеть актуальную информацию о своём заказе (рисунок 37), на странице «Адрес» может добавить платежный адрес и адрес доставки, которые будут использоваться по умолчанию в процессе оформления заказа (рисунок 38), на странице «Детали профиля» пользователь может изменить имя, фамилию и пароль (рисунок 39).

4.6 Личный кабинет пользователя

На рисунках 36 – 39 изображён личный кабинет пользователя. Код реализации функционала представлен в листинге Б.10.

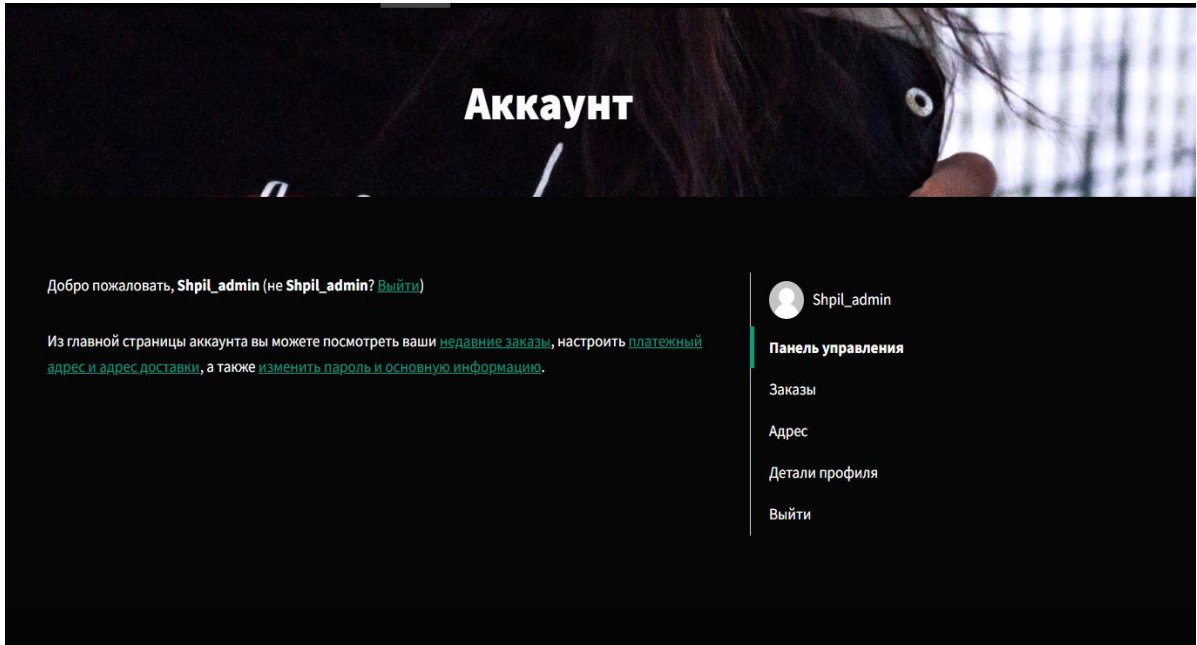


Рисунок 36 – «Аккаунт», панель управления

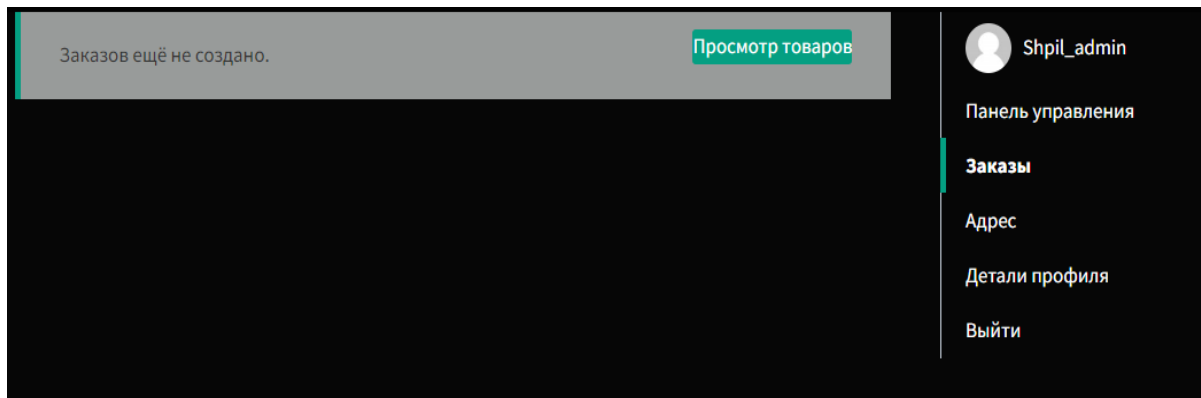


Рисунок 37 – «Аккаунт», заказы

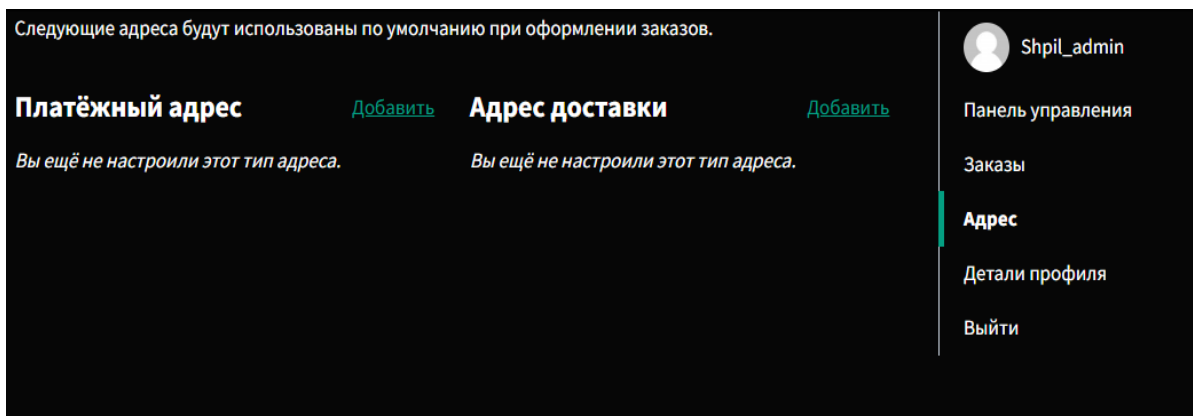


Рисунок 38 – «Аккаунт», адрес

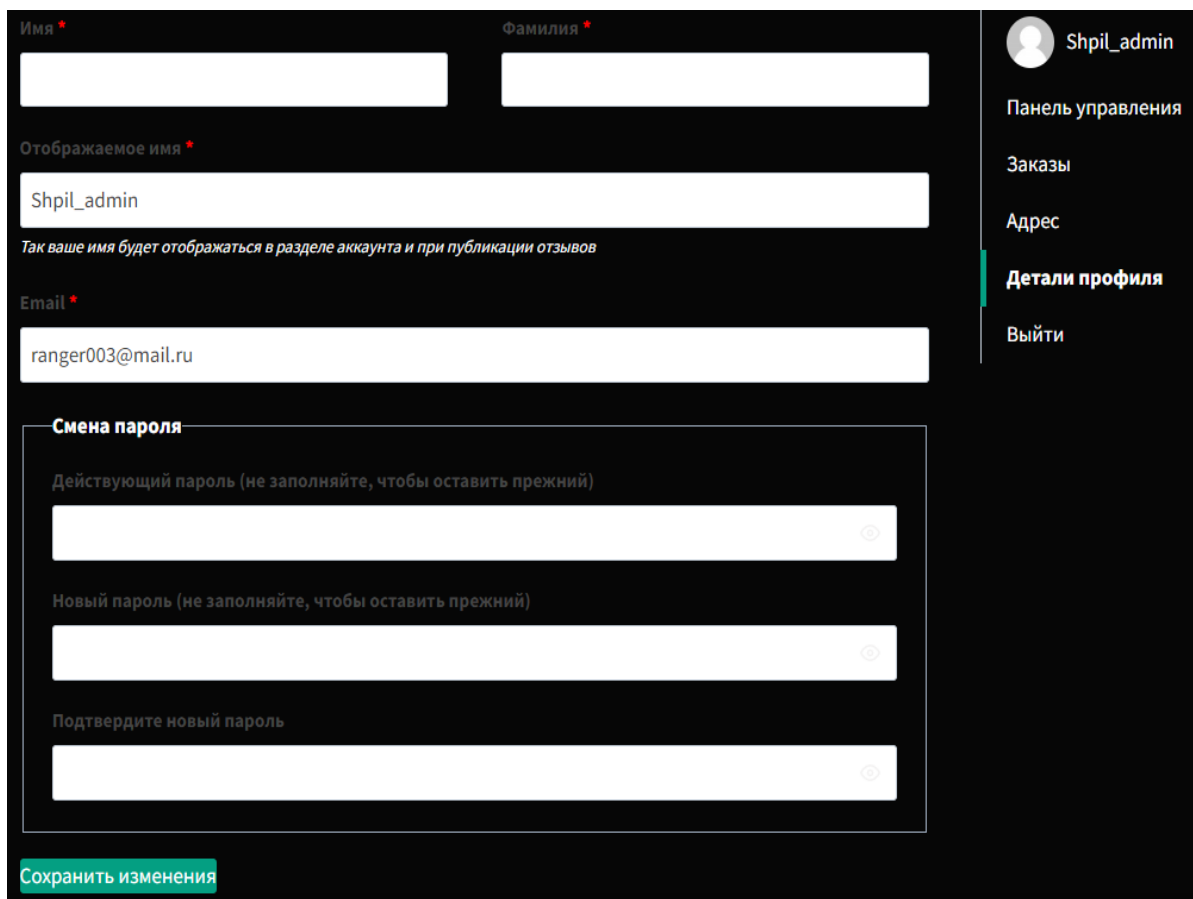


Рисунок 39 – «Аккаунт», детали профиля

5 ТЕСТИРОВАНИЕ

5.1 Тестирование форм ввода

При регистрации нужно ввести Email, попробуем ввести его без символа «@» (Рисунок 40). Как видно на скрине, появилось уведомление о том, что отсутствует специальный символ.

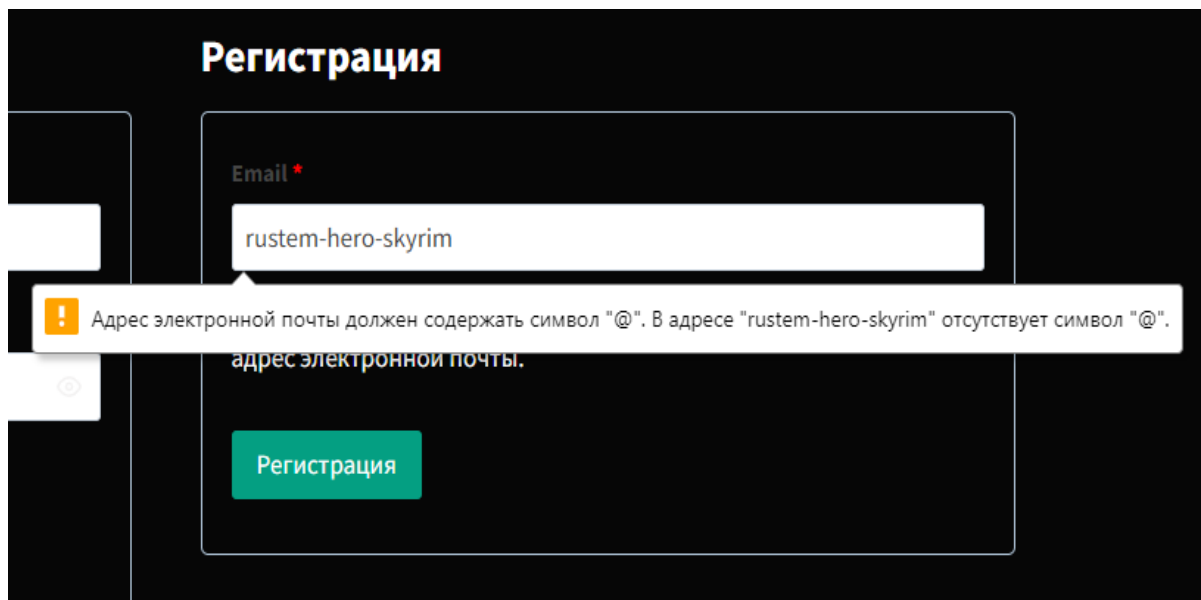


Рисунок 40 – Проверка форм ввода

5.2 Тестирование безопасности

Попытка авторизоваться в зарегистрированном аккаунте с неверным паролем (рисунок 41). Как видно на скрине, появилось уведомление о том, что введенный пароль неверен.

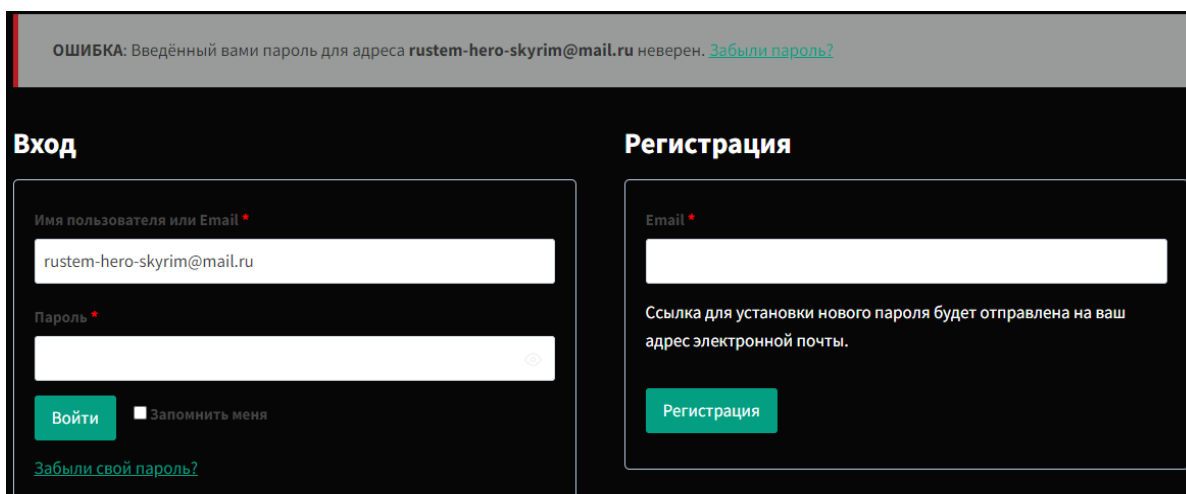


Рисунок 41 – Попытка входа с неверным паролем

5.3 Тестирование авторизации

В формы «Email» и «Пароль» были введены данные от зарегистрированного аккаунта (рисунок 42).

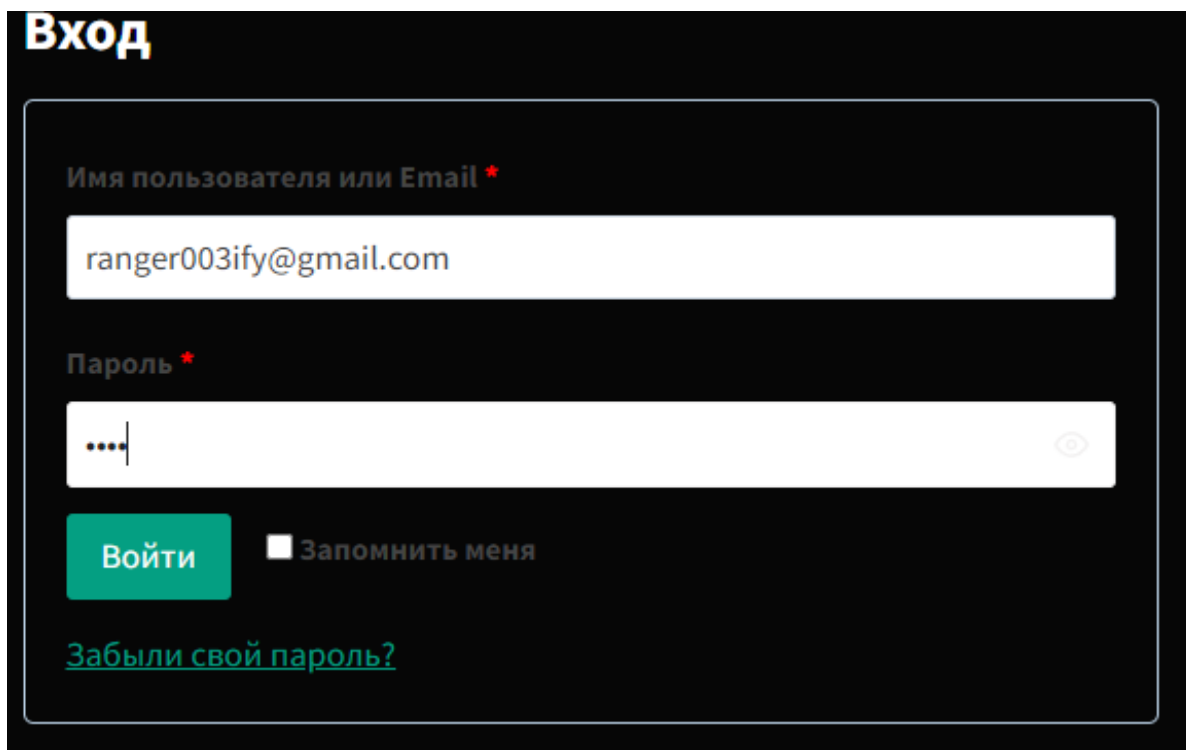


Рисунок 42 – Попытка авторизации

На данном скрине видно, что авторизация прошла успешно (рисунок 43).

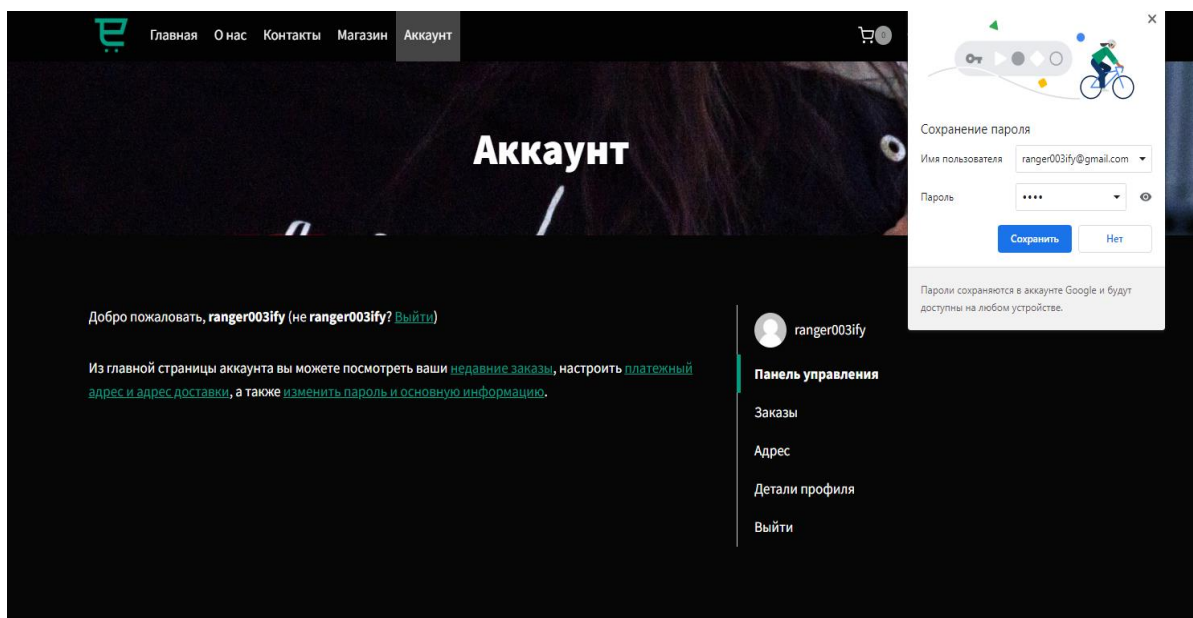


Рисунок 43 – Успешная авторизация

5.4 Тестирование мультязычности сайта

На рисунках 44-46 продемонстрирован перевод содержимого интернет-магазина на трёх доступных иностранных языках.

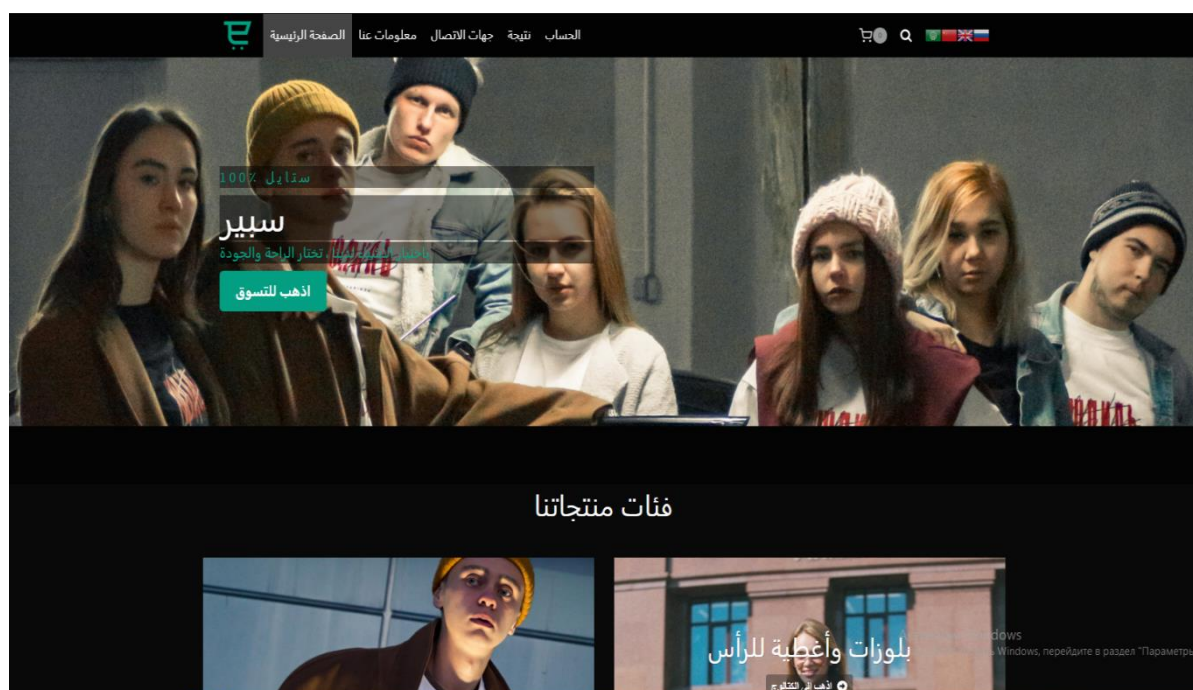


Рисунок 44 – Арабский язык

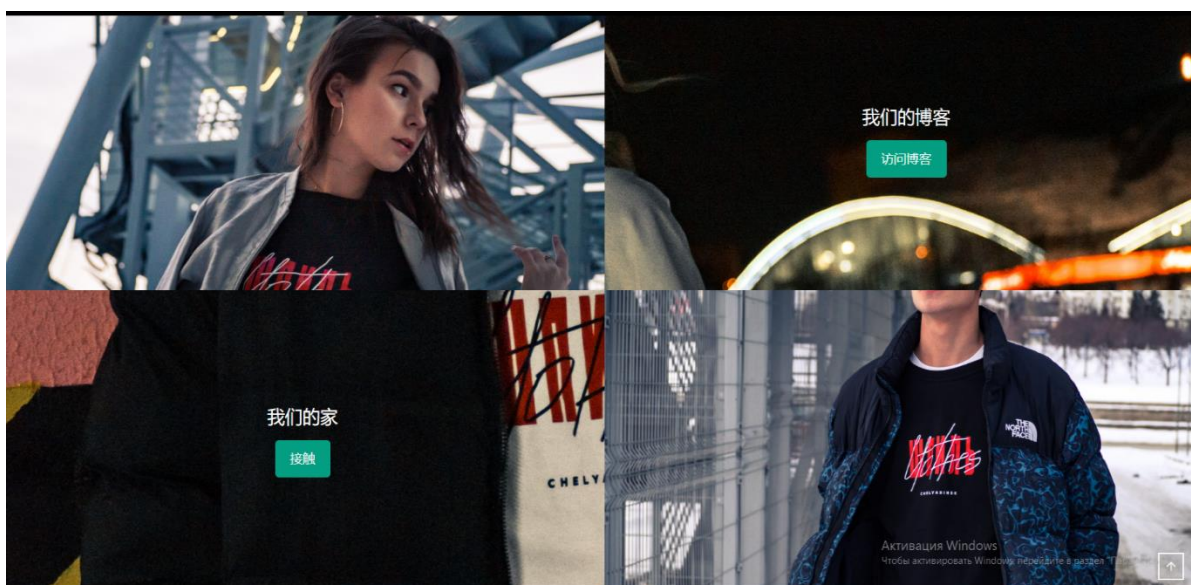


Рисунок 45 – Китайский язык

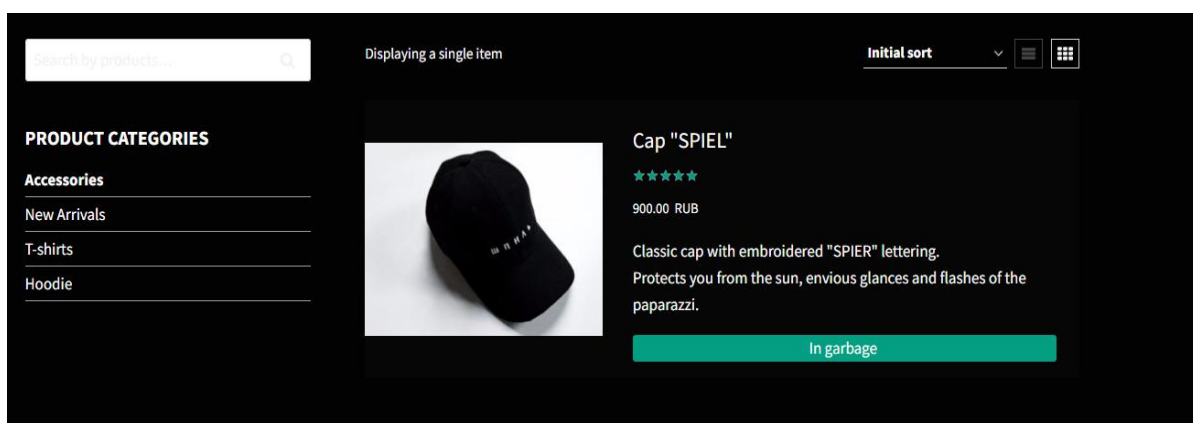


Рисунок 46 – Английский язык

5.5 Тестирование написания отзыва и выставления рейтинга товару

В форме предложенной для отзыва был написан комментарий и выставлено пять звёзд, после была нажата кнопка «отправить» (рисунок 47).

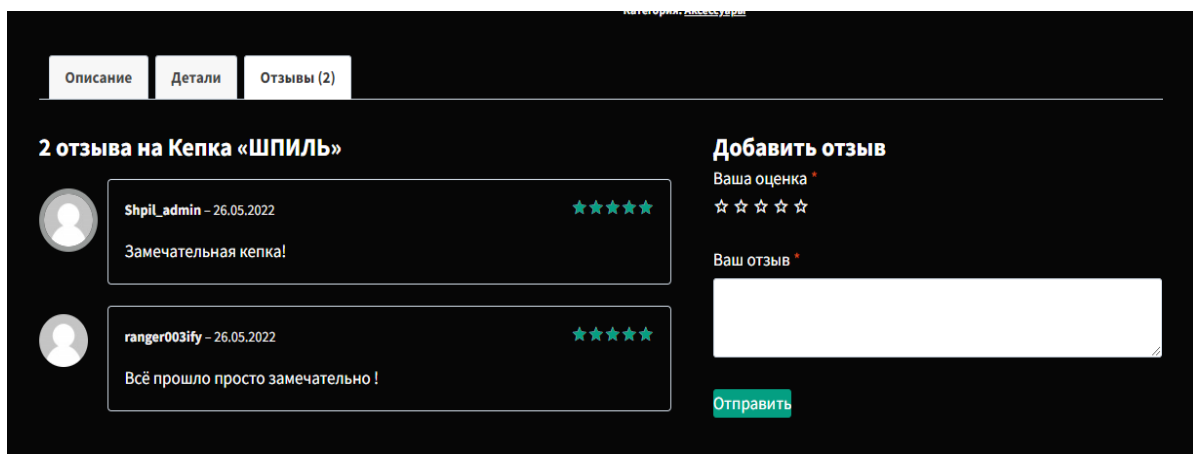


Рисунок 47 – Отзыв к карточке товара

5.6 Тестирование оформления заказа

На рисунках 48-52 продемонстрирован процесс оформления заказа.

Детали оплаты

Имя *

Рустем

Фамилия *

Аблялимов

Страна/регион (необязательно)

Россия

Адрес (необязательно)

Академика Королёва 47-24

Населённый пункт (необязательно)

Челябинск

Область / район (необязательно)

Челябинская область

Почтовый индекс (необязательно)

454004

Телефон *

89226354162

Email *

rustem-hero-skyrim@mail.ru

Рисунок 48 – Заполненные информационные поля на странице оформления заказа

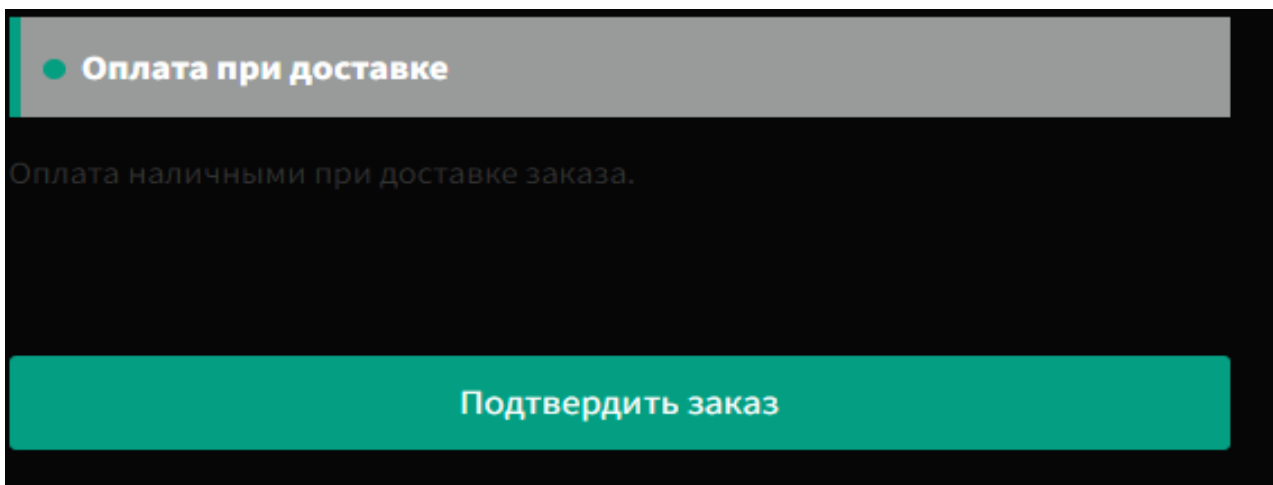


Рисунок 49 – Выбор способа оплаты, вариант доставки единственный – САМОВЫВОЗ

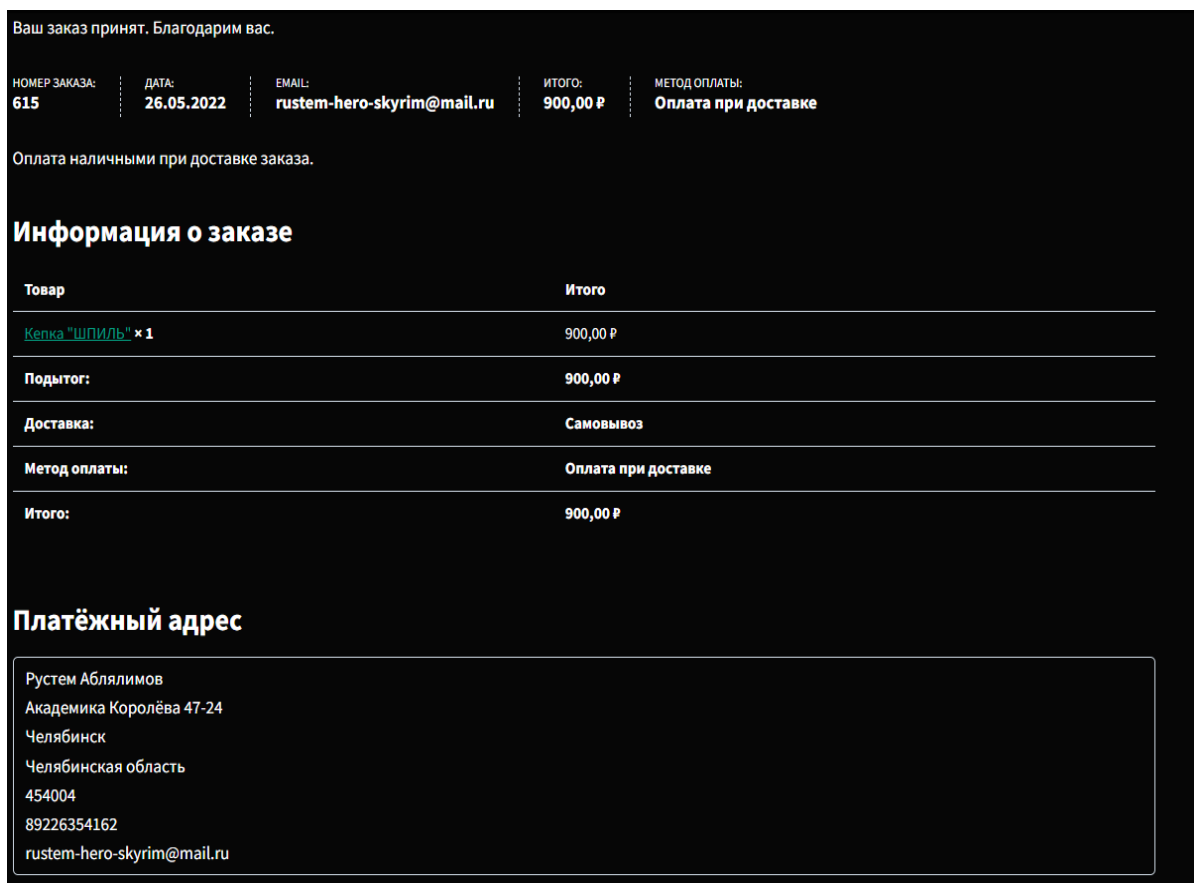



Рисунок 50 – Информация о заказе в личном кабинете пользователя

| Заказ | Дата | Статус | Итого | Действия |
|----------------------|------------|-----------|-----------------------|--------------------------|
| №615 | 26.05.2022 | Обработка | 900,00 Р за 1 единицу | Просмотр |

 **rustem-hero-skyrim**

[Панель управления](#)

Заказы

[Адрес](#)

[Детали профиля](#)

[Выйти](#)

Рисунок 51 – «Заказы» в личном кабинете пользователя

Заказ № 615
Обработка
×

Детали оплаты

Рустем Абляимов
 Академика Королёва 47-24
 Челябинск
 Челябинская область
 454004

Email
rustem-hero-skyrim@mail.ru

Телефон
[89226354162](tel:89226354162)

Платёж через
 Оплата при доставке

| Товар | Количество | Итого |
|---------------|------------|----------|
| Кепка "ШПИЛЬ" | 1 | 900,00 Р |
| Кепка "ШПИЛЬ" | | |

Завершен
Изменить

Рисунок 52 – Информация о заказе в административной панели WordPress

5.7 Тестирование адаптивности интернет-магазина

На рисунках 53 – 54 показана работа интернет-магазина на планшетном устройстве и телефоне.

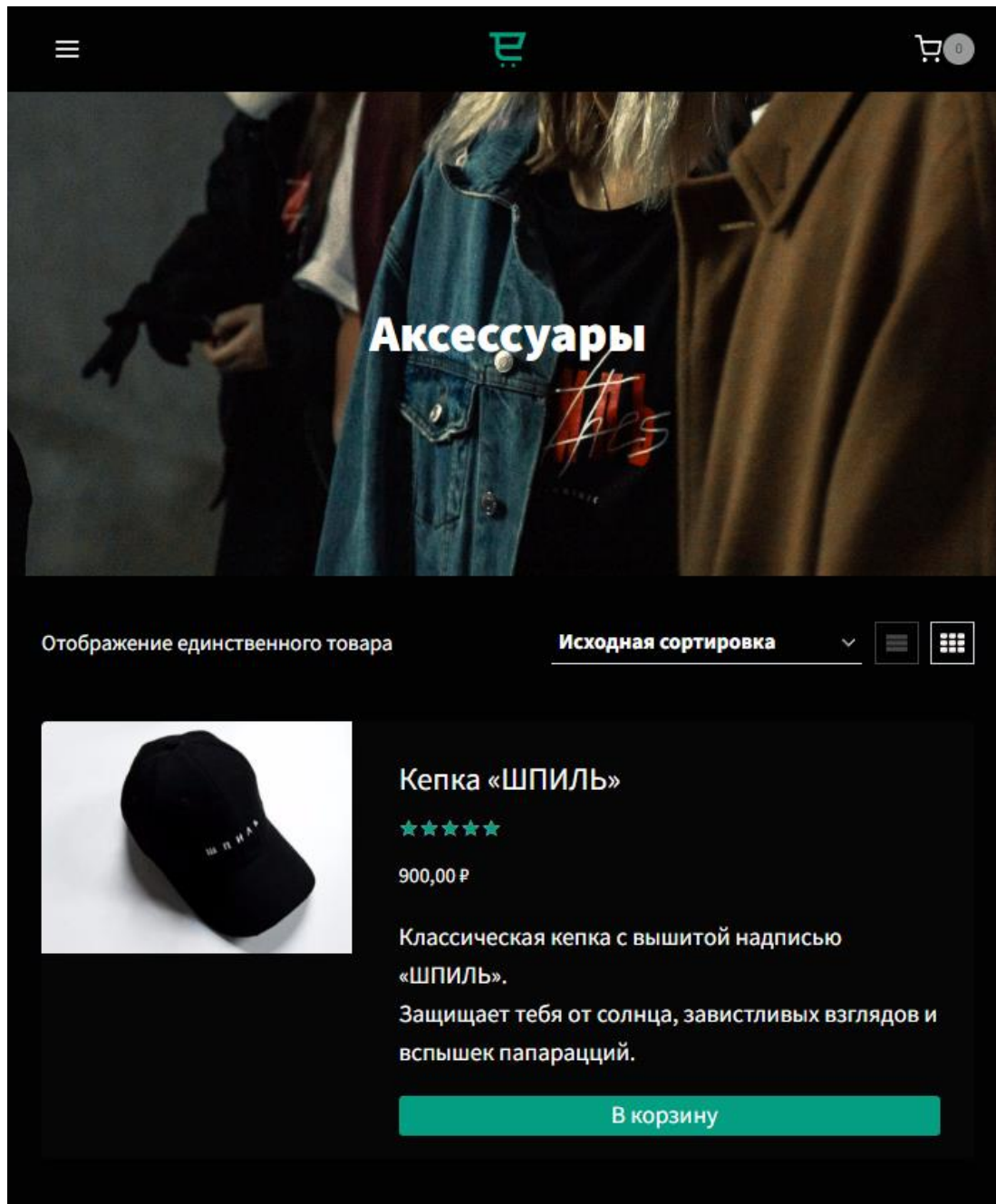


Рисунок 53 – Работа сайта на планшетном устройстве

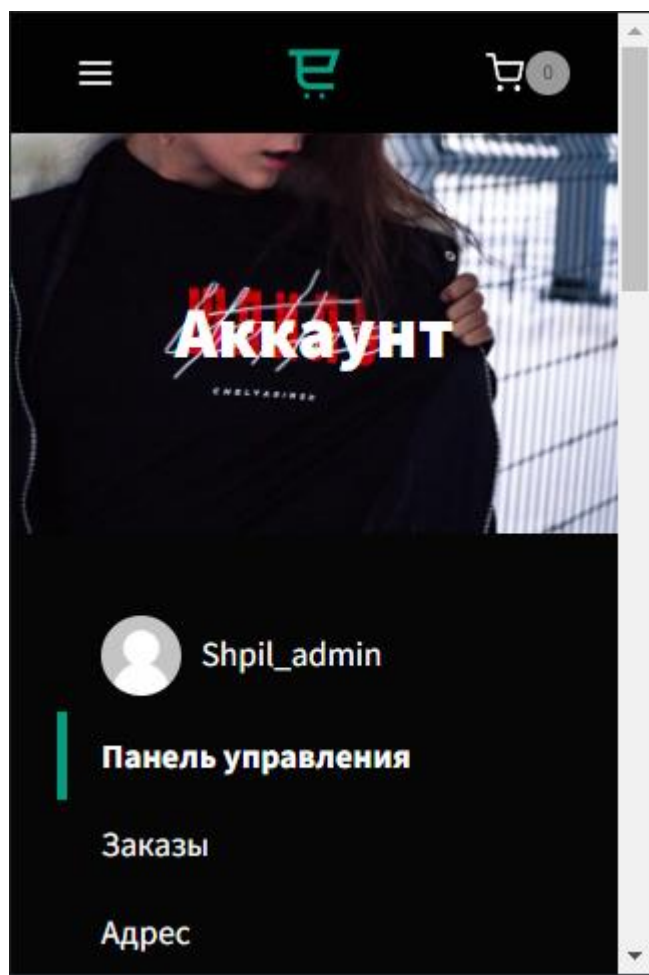


Рисунок 54 – Работа сайта на мобильном устройстве

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы был разработан и адаптирован для планшетов и мобильных устройств интернет-магазин для бренда одежды «ШПИЛЬ». При этом были решены следующие задачи:

- 1) проведен анализ аналогов;
- 2) проведено формирование требований к системе;
- 3) выполнено проектирование системы;
- 4) выполнена реализация системы;
- 5) проведено тестирование системы.

Конечный продукт удовлетворяет всем требованиям современных интернет-магазинов, что делает его удобным в использовании для пользователей. Размещение его на движке CMS WordPress делает редактирование и управление контентом понятным и простым для заказчика, а именно владельца бренда одежды «ШПИЛЬ».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Флэнаган, Д. JavaScript. Подробное руководство / Д. Флэнаган ; [пер. с английского А. Киселева]. – 6-е изд. – Санкт-Петербург : Символ-Плюс, 2012. – 1080 с. : ил. – ISBN 978-5-459-00439-7.
2. Ripndip : [официальный сайт] / Ripndip. – США, 2022. – URL: <https://www.ripndipclothing.com> (дата обращения: 15.05.2022). – Текст. Изображение : электронные.
3. Ricewear : [официальный сайт] / Ricewear. – РФ, 2022. – URL: <https://ricewear.com/shop> (дата обращения: 15.05.2022). – Текст. Изображение : электронные.
4. Novayawear : [официальный сайт] / Novayawear. – РФ, 2022. – URL: <https://novayawear.com> (дата обращения: 15.05.2022). – Текст. Изображение : электронные.
5. Volchok : [официальный сайт] / Volchok. – РФ, 2022. – URL: <https://volchok.ru/store> (дата обращения: 15.05.2022). – Текст. Изображение : электронные.
6. Krakatauwear : [официальный сайт] / Krakatauwear. – РФ, 2022. – URL: <https://en.krakatauwear.com> (дата обращения: 15.05.2022). – Текст. Изображение : электронные.
7. Зандстра, М. PHP. Объекты, шаблоны и методики программирования / М. Зандстра ; [пер. с английского С. Г. Тригуба]. – 3-е изд. – Москва : Диалектика-Вильямс, 2019. – 736 с. : ил. – ISBN 978-5-907144-54-5.
8. МакГрат, М. PHP7 для начинающих с пошаговыми инструкциями / М. МакГрат ; [пер. с английского М. А. Райтмана]. – Москва : Издательство «Э», 2018. – 256 с. : ил. – ISBN 978-5-699-98594-4.
9. Дакетт, Д. JavaScript и jQuery. Интерактивная веб – разработка /

Д. Дакетт ; [пер. с английского М. А. Райтмана]. – Москва : Издательство «Э», 2017. – 640 с. : ил. – ISBN 978-5-699-80285.

10. Справочник по HTML / В. Мержевич. – Текст : электронный // htmlbook.ru : [сайт]. – 2022. – URL: <http://htmlbook.ru/html> (дата обращения: 22.05.22).

11. Полянин, М. Красивый цветной текст в CSS: как это сделать / М. Полянин. – Текст : электронный // thecode.media.ru : [сайт]. – 30 августа 2021. – URL: <https://thecode.media/fill-color/> (дата обращения: 18.05.22).

12. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хэлм, Р. Джонсон, Д. Влиссидес ; [пер. с английского А. Силкина]. – Санкт-Петербург : Питер, 2001. – 368 с. : ил. – ISBN 5-272-00355-1.

13. Создание динамического веб сайта / Г. Дударь. – Изображение (подвижное ; двумерное) : электронное // Гоша Дударь : [канал YouTube]. – 21 августа 2015. – URL: https://www.youtube.com/c/gosha_dudar/search?query=php (дата обращения: 23.05.22).

14. Сафронов, М. Разработка веб-приложений в Yii2 / М. Сафронов. – Москва : ДМК Пресс, 2015. – 392 с. : ил. – ISBN 978-5-97060-252-2.

ПРИЛОЖЕНИЕ А

Интерфейсы сайтов, приведенных в обзоре



Рисунок А.1 – Шапка сайта ripndipclothing.com



Рисунок А.2 – Тело сайта ripndipclothing.com

RETURNS / EXCHANGES
SHIPPING / ORDERING FAQ
STOCKISTS
ABOUT
CONTACT
TERMS
PRIVACY POLICY
ACCESSIBILITY STATEMENT
CLICK FOR ACCESSIBILITY WIDGET BY USERWAY



RIPNDIP FLAGSHIP
441 N FAIRFAX AVE
LOS ANGELES, CA 90036

COPYRIGHT © 2022 RIPNDIP | SITE BY CAMP DOUG

INSTAGRAM
FACEBOOK
TWITTER
TUMBLR

QUESTIONS?
INFO [AT] RIPNDIPCLOTHING.COM

ACCOUNT LOGIN

SIGN UP FOR OUR NEWSLETTER

Stay up to date with the latest RIPNDIP news, releases, events and special offers. Subscribe to our newsletter and receive a personal discount code.

Enter your email

SUBSCRIBE

Рисунок А.3 – Подвал сайта ripndipclothing.com

Продолжение приложения А

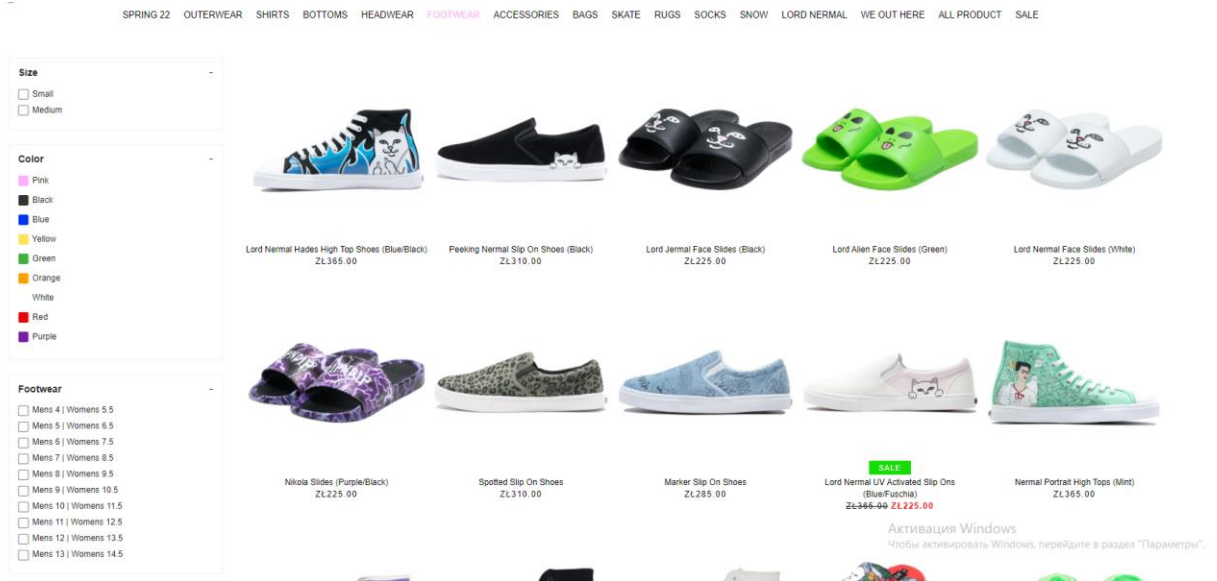


Рисунок А.4 – Страница каталога girndipclothing.com

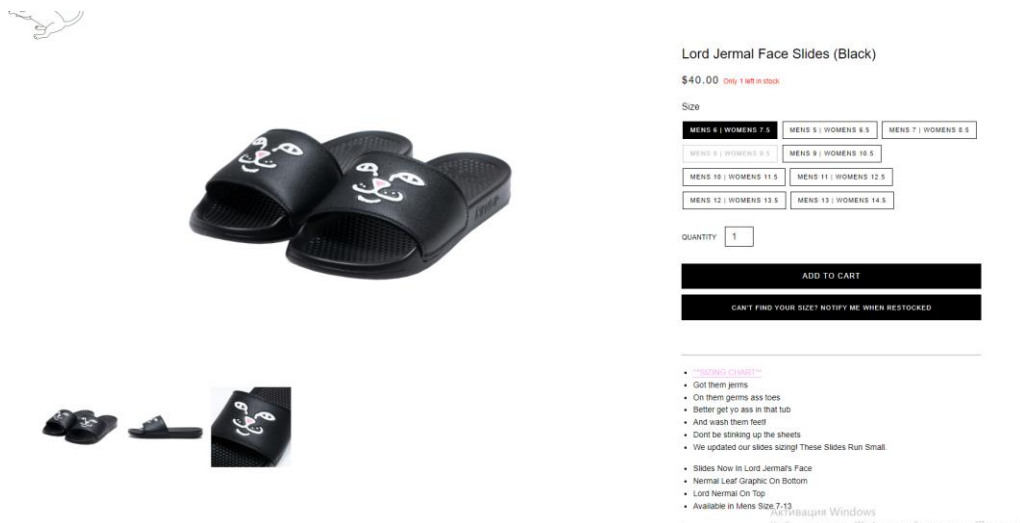


Рисунок А.5 – Карточка товара girndipclothing.com



RICE

+7 909 000 40 63
info@ricewear.com

[магазин](#)

[lookbook](#)

[доставка и оплата](#)

[о бренде](#)

[контакты](#)

Рисунок А.6 – Шапка сайта ricewear.com



Рисунок А.7 – Тело главной страницы ricewear.com



Rice.Korea

Необычные формы и силуэты раскрывают самобытную философию одной из стран Восточной Азии - Кореи



Anime

Коллекция, посвященная бескрайнему и чарующему миру Японской мультипликации

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры"

Рисунок А.8 – Тело главной страницы ricewear.com



Рисунок А.9 – Тело главной страницы ricewear.com



Рисунок А.10 – Тело главной страницы ricewear.com



Рисунок А.11 – Каталог ricewear.com

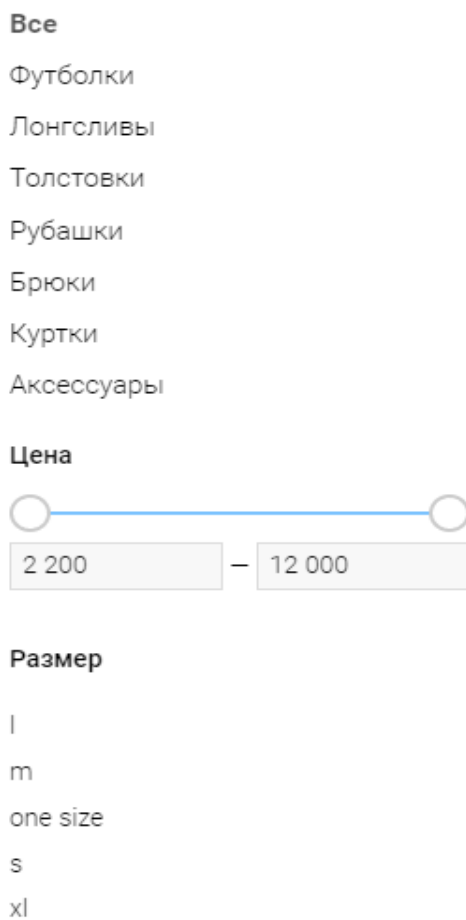


Рисунок А.12 – Фильтры в каталоге ricewear.com

Кимоно Rice коричневое

6 400 руб

Размер

КУПИТЬ



Кимоно RICE из хлопка. Кимоно свободного кроя. Классический воротник, завязка на широкую кулису спереди. Карманы в боковых швах. Кимоно сочетается с лонгсливами, шортами, брюками RICE. Альтернатива жакетам, джемперам в любое время года.

Состав: 100% хлопок

Цвет: коричневый

Размер: S, M, L

Унисекс

Рисунок А.13 – Карточка товара ricewear.com



RICE

магазин

lookbook

доставка и оплата

о бренде

контакты



+7 909 000 40 63
info@ricewear.com

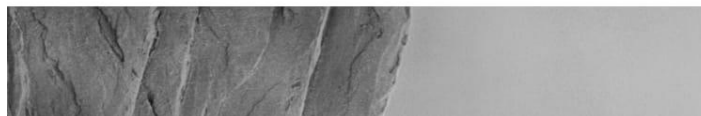


Рисунок А.14 – Мобильная версия ricewear.com

[Novaya](#) [Women](#) [Men](#) [Specials](#) [Sale](#) [Инфо](#) [Бренд](#) [Коллекции](#) [Контакты](#) [Магазины](#) [English](#)

Рисунок А.15 – Шапка сайта novayawear.com

ACCESSORIES



PANTS



Рисунок А.16 – Тело главной страницы novaawear.com

НОВОЕ ЖЕНСКОЕ



Рисунок А.17 – Тело главной страницы novaawear.com

НОВОЕ МУЖСКОЕ

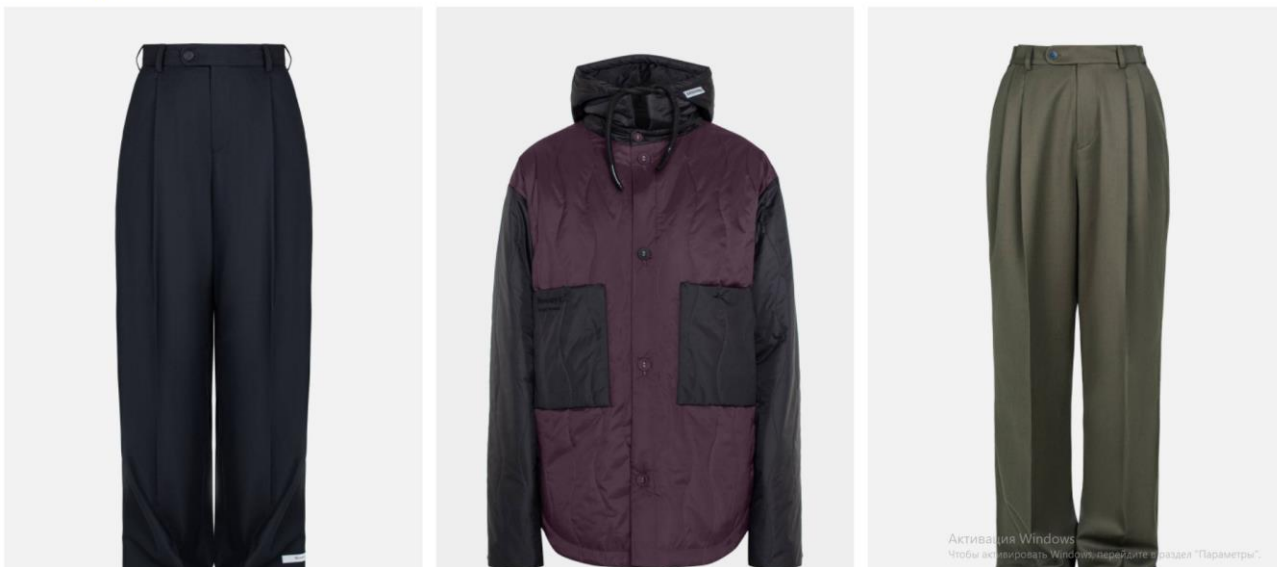


Рисунок А.18 – Тело главной страницы novayaawear.com

Бренд Контакты Магазины Доставка Оплата Возврат Privacy Newsletter Instagram Vkontakte Facebook Pinterest Back to top

Рисунок А.19 – Подвал сайта novayaawear.com

ФУТБОЛКИ



Рисунок А.20 – Каталог novayaawear.com

4 200 Руб



Добавить в корзину →

BASIC T-SHIRT BLACK

Футболка из плотного хлопка. Модель унисекс. Спущенная линия плеча, свободный прямой крой. Имеется накладной карман. На полочке внизу находится патч-составник.



[Состав и уход](#) [Таблица размеров](#) [Гид по посадке](#) [Доставка](#) [Оплата](#)



Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок А.21 – Карточка товара

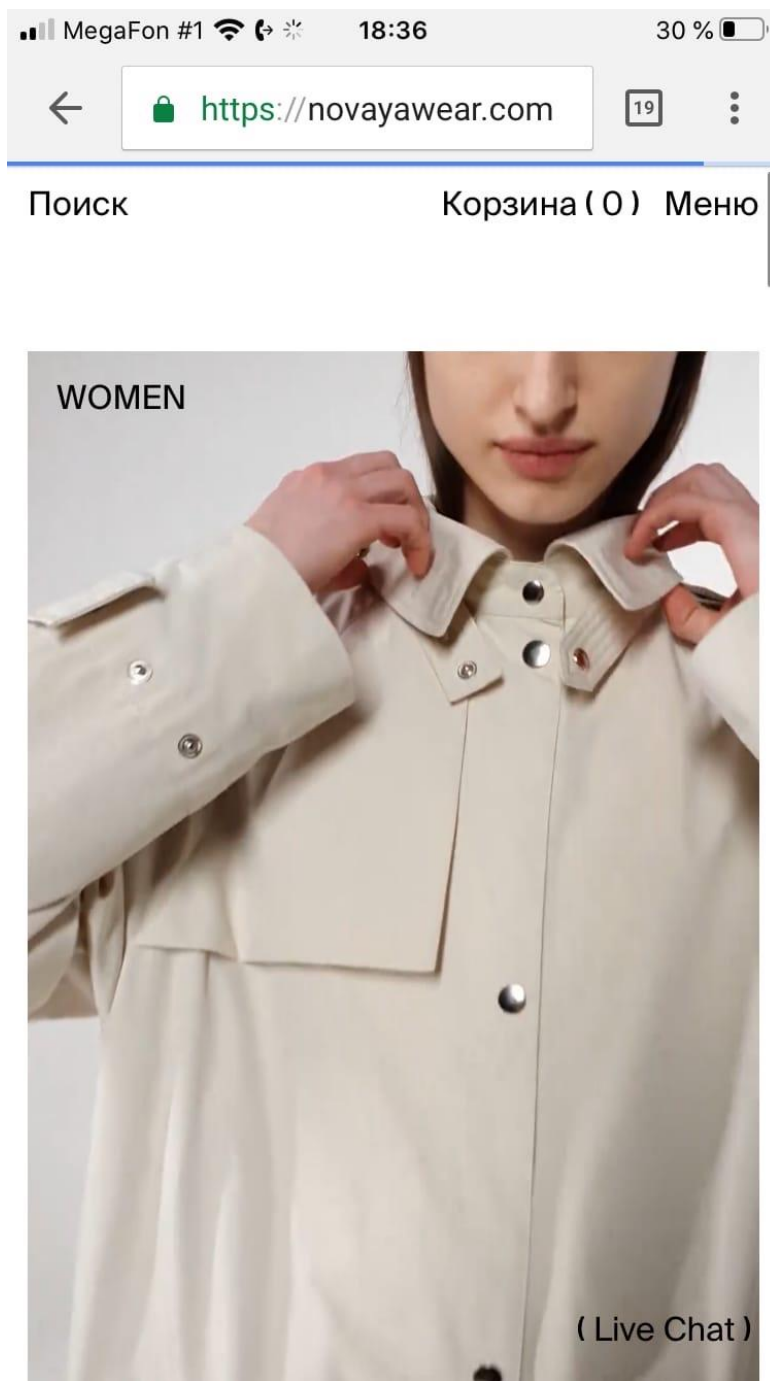


Рисунок А.22 – Мобильная версия novayawear.com

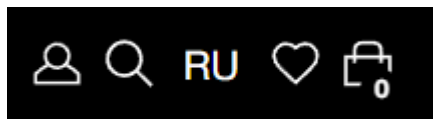


Рисунок А.23 – Шапка сайта volchok.ru



Рисунок А.24 – Шапка сайта volchok.ru

СКИДКИ

РАЗМЕР ▾ НОВИНКИ ▾



ХУДИ СВЯТОЙ СЕБАСТЬЯН
5 200 РУБ ~~7 400 РУБ~~



РЮКЗАК ЦВЕТЫ
4 500 РУБ ~~6 400 РУБ~~



КАПЮШОН СТЕГАНЫЙ ВОЗРОЖДЕНИЕ
3 200 РУБ ~~4 000 РУБ~~



БРЮКИ КАРГО КАМУФЛЯЖ
4 600 РУБ ~~5 800 РУБ~~
Активация Windows

Рисунок А.25 – Тело главной страницы volchok.ru

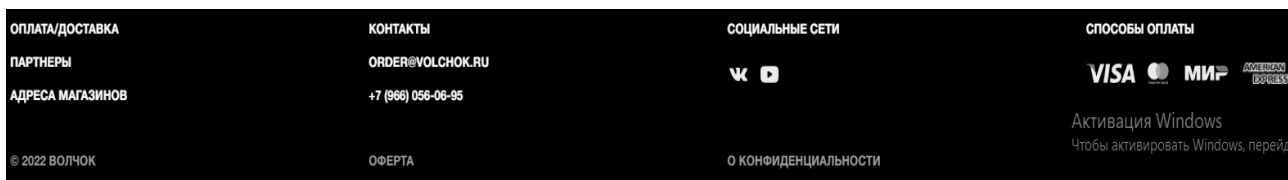


Рисунок А.26 – Подвал сайта volchok.ru

ФУТБОЛКА PEASE

2 200 РУБ



XS S M L XL XXL

РАЗМЕРНАЯ СЕТКА



В КОРЗИНУ

В НАЛИЧИИ

Москва: Да

Санкт-Петербург: Нет

Екатеринбург: Нет

Казань: Да

ОПИСАНИЕ

ДОСТАВКА

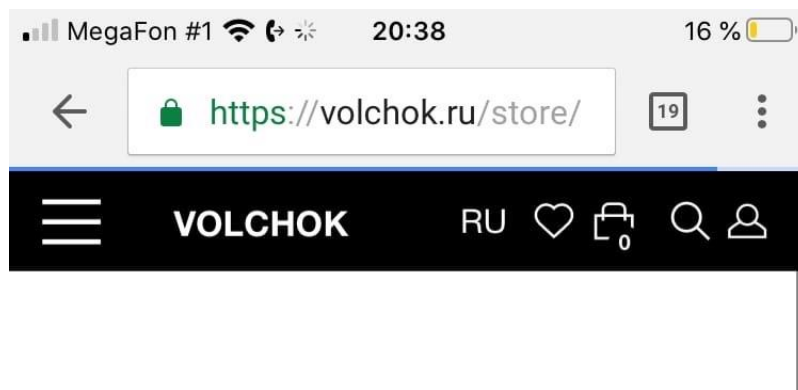
СОСТАВ

100% хлопок. Рекомендации по уходу: бережная машинная стирка при 30° без отбеливателей, глажка с изнаночной стороны.

Рисунок А.27 – Карточка товара volchok.ru

| | |
|---|---|
| <p>ЛИЧНЫЕ ДАННЫЕ</p> <p>АДРЕСНАЯ КНИГА</p> <p>ЗАКЛАДКИ</p> <p>ИСТОРИЯ ЗАКАЗОВ</p> <p>ВЫЙТИ</p> | <p>ЛИЧНЫЙ КАБИНЕТ</p> <p>ИМЯ* Рустем</p> <hr/> <p>ФАМИЛИЯ Аблялимов</p> <hr/> <p>E-MAIL ranger003@mail.ru</p> <hr/> <p>ТЕЛЕФОН* +7</p> <hr/> <p>ПАРОЛЬ</p> <hr/> <p>ЗАДАТЬ НОВЫЙ ПАРОЛЬ</p> <hr/> <p>СОХРАНИТЬ</p> |
|---|---|

Рисунок А.28 – Кабинет пользователя volchok.ru



SALE



**ХУДИ СВЯТОЙ
СЕБАСТЬЯН**
5 200 РУБ ~~7 400 РУБ~~



**ДЖИНСОВАЯ
КУРТКА ЦВЕТЫ**
4 300 РУБ ~~8 600 РУБ~~



Рисунок А.29 – Мобильная версия volchok.ru

Рисунок А.30 – Шапка сайта krakatauwear.com

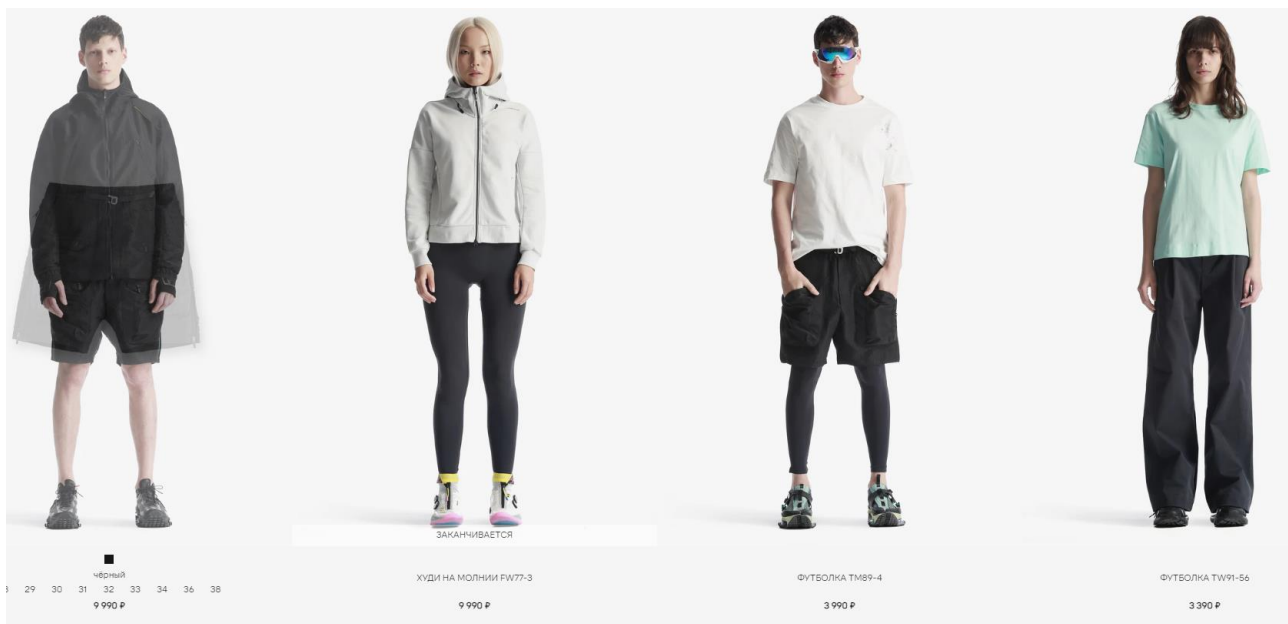


Рисунок А.31 – Тело главной страницы krakatauwear.com

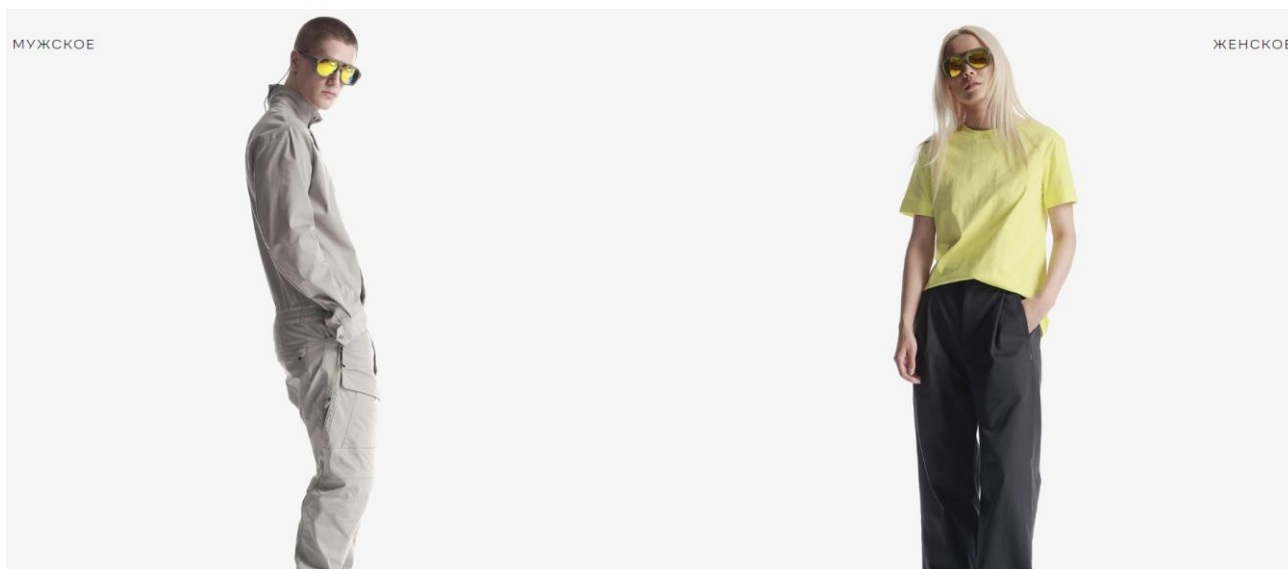


Рисунок А.32 – Тело главной страницы krakatauwear.com

ПОМОЩЬ ПОКУПАТЕЛЮ
8 800 444 2404

НУЖНА ПОМОЩЬ?

КРАКАТАУ

FAQ

О БРЕНДЕ

ДОСТАВКА

МАГАЗИНЫ

ВОЗВРАТ

ОПЛАТА

ПОЛИТИКА КОНФИДЕНЦИАЛЬНОСТИ

СВЯЗАТЬСЯ С НАМИ

ПОЛЬЗОВАТЕЛЬСКОЕ СОГЛАШЕНИЕ



Рисунок А.33 – Подвал сайта krakatauwear.com

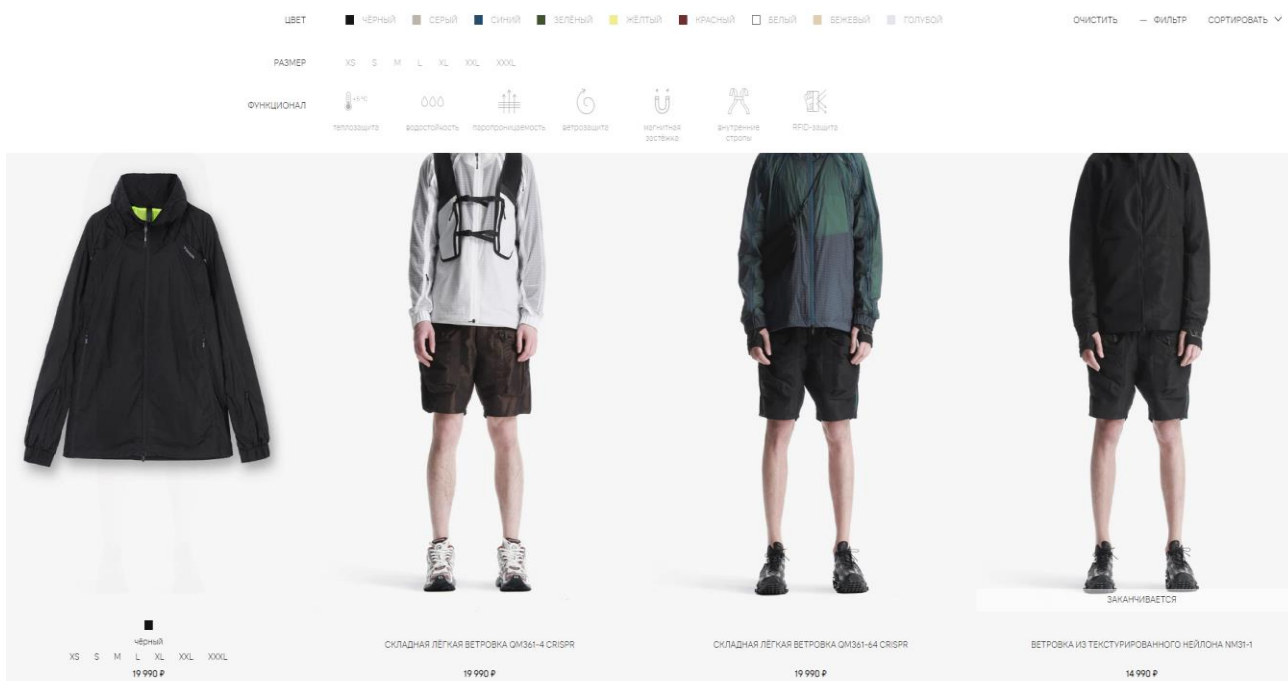


Рисунок А.34 – Каталог krakatauwear.com

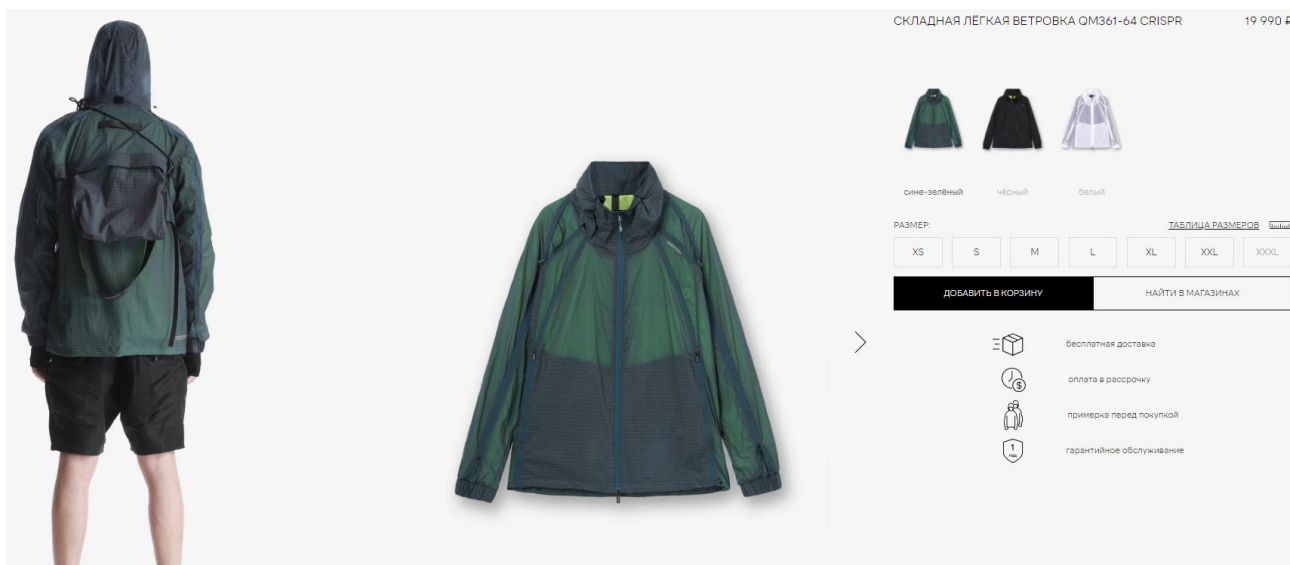


Рисунок А.35 – Карточка товара krakatauwear.com

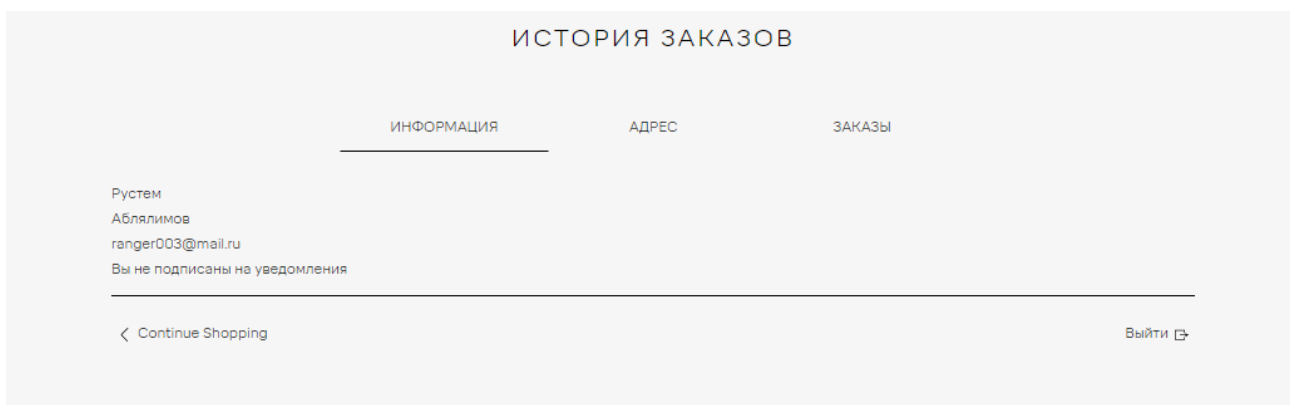


Рисунок А.36 – Личный кабинет пользователя krakatauwear.com

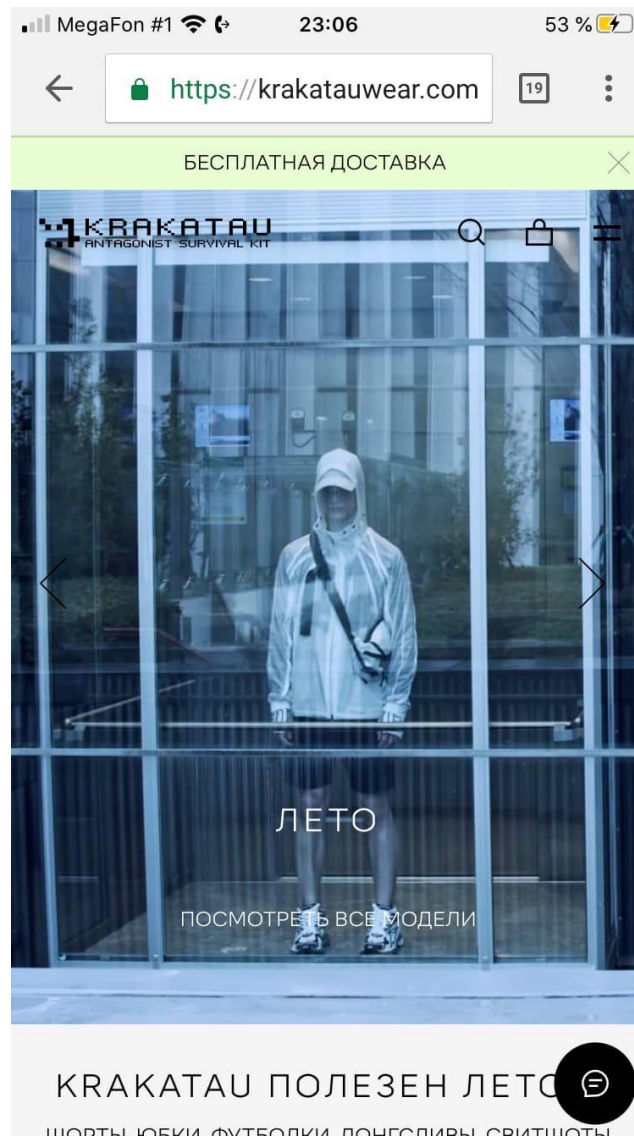


Рисунок А.37 – Мобильная версия krakatauwear.com

ПРИЛОЖЕНИЕ Б

Код программы

Листинг Б.1 – Абстрактный класс, который содержит общую логику

```
<?php

/**
 * абстрактный класс, который содержит общую логику
 * для контроллеров панели администрирования
 */
abstract class adminBase {

    /**
     * Проверяет, является ли пользователь администратором
     *
     * @return bool
     */
    public static function checkAdmin (){

        //Если авторизован - получаем id пользователя
        $userId = User::checkLog();

        //Вытаскиваем информацию о пользователе
        $user = User::getUserById($userId);

        //Проверяем роль пользователя
        if($user['role'] == 'admin'){
            return true;
        }else{
            die ("неверная роль");
        }
    }
}
```

Листинг Б.2 – Автозагрузка классов

```
<?php

/**
 * Автозагрузка классов
 */
function __autoload ($className) {

    //директории, в которых будет идти поиск классов
    $arrayPath = array(
        '/models/',
        '/components/'
    );

    foreach ($arrayPath as $path) {
        $path = ROOT . $path . $className . '.php';

        if (is_file($path))
            include_once $path;
    }
}
```



```

    */
private $total;
/**
 *
 * @var Записей на страницу
 *
 */
private $limit;
/**
 * Запуск необходимых данных для навигации
 * @param type $total <p>Общее количество записей</p>
 * @param type $currentPage <p>Номер текущей страницы</p>
 * @param type $limit <p>Количество записей на страницу</p>
 * @param type $index <p>Ключ для url</p>
 */
public function __construct($total, $currentPage, $limit, $index)
{
    # Устанавливаем общее количество записей $this->total = $total;
    # Устанавливаем количество записей на страницу $this->limit = $limit;
    # Устанавливаем ключ в url $this->index = $index;
    # Устанавливаем количество страниц $this->amount = $this->amount();
    # Устанавливаем номер текущей страницы $this->setCurrentPage($currentPage);
}

/**
 * Для вывода ссылок
 * @return HTML-код со ссылками навигации
 */
public function get() {
    # Для записи ссылок $links = null;
    # Получаем ограничения для цикла $limits = $this->limits();
    $html = '<ul class="pagination">';
    # Генерируем ссылки for ($page = $limits[0]; $page <= $limits[1]; $page++)

    {
        # Если текущая это текущая страница, ссылки нет и добавляется класс active if ($page ==
        $this->current_page)

        {
            $links .= '<li class="active"><a href="#">' . $page . '</a></li>';
        }

    else {
        # Иначе генерируем ссылку $links .= $this->generateHtml($page);
    }

}

# Если ссылки создались
if (!is_null($links)) {
    # Если текущая страница не первая if ($this->current_page > 1) # Создаём ссылку "На первую"
    $links = $this->generateHtml(1, '&lt;&lt;') . $links;
}

```

```

    # Если текущая страница не первая if ($this->current_page < $this->amount) # Создаём ссылку
    "На последнюю" $links .= $this->generateHtml($this->amount, '&gt;&gt;');
}

$html .= $links . '</ul>';

# Возвращаем html
return $html;
}

/**
 * Для генерации HTML-кода ссылки
 * @param integer $page - номер страницы
 *
 * @return
 */
private function generateHtml($page, $text = null) {
    # Если текст ссылки не указан if (!$text) # Указываем, что текст - цифра страницы $text =
    $page;
    $currentURI = rtrim($_SERVER['REQUEST_URI'], '/') . '/';
    $currentURI = preg_replace('~\/page-[0-9]+\~', '', $currentURI);
    # Формируем HTML код ссылки и возвращаем return '<li><a href="' . $currentURI . $this->index
    . $page . '">' . $text . '</a></li>';
}

/**
 * Для получения, откуда стартовать
 *
 * @return массив с началом и концом отсчёта
 */
private function limits() {
    # Вычисляем ссылки слева (чтобы активная ссылка была посередине) $left = $this->current_page
    - round($this->max / 2);
    # Вычисляем начало отсчёта $start = $left > 0 ? $left : 1;
    # Если впереди есть как минимум $this->max страниц if ($start + $this->max <= $this->amount)

    {
        # Назначаем конец цикла вперёд на $this->max страниц или просто на минимум $end = $start > 1
        ? $start + $this->max : $this->max;
    }

    else {
        # Конец - общее количество страниц $end = $this->amount;
        # Начало - минус $this->max от конца $start = $this->amount - $this->max > 0 ? $this->amount
        - $this->max : 1;
    }

    # Возвращаем
    return
    array($start, $end);
}

/**

```

```

    * Для установки текущей страницы
    *
    *
    */
private function setCurrentPage($currentPage) {
    # Получаем номер страницы $this->current_page = $currentPage;
    # Если текущая страница больше нуля if ($this->current_page > 0)

    {
        # Если текущая страница меньше общего количества страниц if ($this->current_page > $this->amount) # Устанавливаем страницу на последнюю $this->current_page = $this->amount;
    }

    else
    # Устанавливаем страницу на первую
    $this->current_page = 1;
    }

    /**
     * Для получения общего числа страниц
     *
     * @return число страниц
     */
private function amount() {
    # Делим и возвращаем return ceil($this->total / $this->limit);
}
}

```

Листинг Б.5 – Настройка маршрутизатора

```

<?php
/**
 * Class Router маршрутизатор
 */
class Router {
    private $routes;
    /**
     * Router constructor подключает маршруты
     */
    public function __construct ()

    {
        $routesPath = ROOT . "/config/routes.php";
        $this->routes = include($routesPath);
    }

    /**
     * @return string текущий адрес запроса
     */
private function getURI() {
    if (isset($_SERVER['REQUEST_URI']) and !empty($_SERVER['REQUEST_URI'])) return
    trim($_SERVER['REQUEST_URI'], '/');
}
}

```

```

}

public function start() {
    //получаем строку запроса $uri = $this->getURI();
    // Проверить наличие такого запроса в routes.php foreach ($this->routes as $uriPattern =>
    $path)

    {
        //Сравниваем uriPattern и $uri if(preg_match("~$uriPattern~", $uri))

    {
        // Получаем внутренний путь из внешнего согласно правилу $internalRoute =
preg_replace("~$uriPattern~", $path, $uri);
        // Определить контроллер, action, параметры $segments = explode('/', $internalRoute);
        $controllerName = array_shift($segments) . 'Controller';
        $controllerName = ucfirst($controllerName);
        $actionName = 'action' . ucfirst(array_shift($segments));
        $parameters = $segments;
        //Подключаем файл контроллера $controllerFile = ROOT . "/controllers/" . $controllerName .
        ".php";
        if(file_exists($controllerFile)) include_once($controllerFile);
        //Создаем объект контроллера и дергаем нужный action $controllerObject = new $controllerName;
        $result = call_user_func_array(array($controllerObject, $actionName), $parameters);
        if($result !== null) break;
    }
}
}
}
}
}

```

Листинг Б.6 – Модель для работы с корзиной покупателя

```

?php
/**
    * Модель для работы с корзиной покупателя
    */
class Cart {
    /**
        * Добавление товара в корзину
        *
        * @param $id - id товара
        * @return int - количество товаров в корзине
        */
    public static function addProduct ($id)

    {
        $id = intval($id);
        //Пустой массив для товаров в корзине (ключ - id товара, значение - кол-во) $productsInCart =
        array();
        //Если в корзине уже есть товары, заполняем массив if (isset($_SESSION['products']))
        Продолжение приложения Б

    {
        $productsInCart = $_SESSION['products'];
    }
}

```

```

//то увеличиваем количество
if (array_key_exists($id, $productsInCart)) {
    $productsInCart[$id]++;
}

else {
    //Добавляем новый товар в корзину $productsInCart[$id] = 1;
}

$_SESSION['products'] = $productsInCart;

return self::itemsCount();
}

/**
 * Считаем кол-во товаров в корзине(сессия)
 *
 * @return int
 */
public static function itemsCount () {
    if (isset($_SESSION['products']))
    {
        $count = 0;
        foreach ($_SESSION['products'] as $id => $quantity)
        {
            $count = $count + $quantity;
        }

        return $count;
    }
    else
        return 0;
}

/**
 * Вытягиваем массив товаров из сессии
 *
 * @return bool
 */
public static function getProducts () {
    if (isset($_SESSION['products']))
    {

```

```

        return $_SESSION['products'];
    }

    return false;
}

/**
 * Считаю общую сумму товаров в корзине
 *
 * @param $products - информация о товарах из корзины
 * @return int
 */
public static function getTotalPrice ($products) {
    $productsInCart = self::getProducts();
    $total = 0;
    if ($products)
    {
        foreach ($products as $item)
        {
            $total += $item['price'] * $productsInCart[$item['id']];
        }
    }

    return $total;
}

/**
 * Удаление товара из корзины
 *
 * @param $id
 */
public static function deleteProduct ($id) {
    //Получаем массив с id и количеством товара в корзине $productsInCart = self::getProducts();
    //удаляем нужный товар по 1 единице if ($productsInCart[$id] == 1)

    {
        unset($productsInCart[$id]);
    }

    else {
        $productsInCart[$id]--;
    }

    //записываем новый массив в сессию
    $_SESSION['products'] = $productsInCart;
}

```



```

/**
     * Очистка корзины
     */
public static function clear () {
    if (isset($_SESSION['products']))

    {
        unset($_SESSION['products']);
    }

}
}
}

```

Листинг Б.7 – Модель для работы с категориями

```

<?php
/**
 * Модель для работы с категориями
 */
class Category {
    /**
     * @return array список категорий
     */
    public static function getCategory ()

    {
        $db = Db::getConnection();
        $sql = "SELECT id, name FROM category WHERE status = 1 ORDER BY sort_order ASC"; $res
= $db->query($sql);
        $catList = $res->fetchAll(PDO::FETCH_ASSOC);
        return $catList;
    }

    /**
     * Список категорий для админпанели
     * Возвращает массив всех категорий, включая те, у которых статус отображения = 0
     *
     * @return array
     */
    public static function getCategoryListAdmin () {
        $db = Db::getConnection();
        $sql = "SELECT id, name, sort_order, status FROM category ORDER BY sort_order ASC";
$res = $db->query($sql);
        $catList = $res->fetchAll(PDO::FETCH_ASSOC);
        return $catList;
    }

    /**
     * Вместо числового статуса категории, отображаем определенную строку
     *
     * @param $status
     * @return string
     */
    public static function getStatusText ($status) {

```

```

        switch ($status)
        {
            case '1': return 'Отображается';
            break;
            case '0': return 'Скрыта';
            break;
        }
    }
}

/**
 * Удаление категории(админка)
 *
 * @param $id
 * @return bool
 */
public static function deleteCategoryById ($id) {
    $db = Db::getConnection();
    $sql = " DELETE FROM category WHERE id = :id "; $res = $db->prepare($sql);
    $res->bindParam(':id', $id, PDO::PARAM_INT);
    return $res->execute();
}

/**
 * Добавление категории(админка)
 *
 * @param $options массив параметров
 * @return bool
 */
public static function addCategory ($options) {
    $db = Db::getConnection();
    $sql = " INSERT INTO category(name, sort_order, status) VALUES (:name, :sort_order,
:status) "; $res = $db->prepare($sql);
    $res->bindParam(':name', $options['name'], PDO::PARAM_STR);
    $res->bindParam(':sort_order', $options['sort_order'], PDO::PARAM_INT);
    $res->bindParam(':status', $options['status'], PDO::PARAM_INT);
    return $res->execute();
}

/**
 * Возвращаем инфу о категории по ее id
 *
 * @param $id
 * @return mixed
 */
public static function getCategoryById ($id) {
    $db = Db::getConnection();
    $sql = "SELECT name, sort_order, status FROM category WHERE id = :id"; $res = $db-
>prepare($sql);
    $res->bindParam(':id', $id);
    $res->execute();
    $catList = $res->fetch(PDO::FETCH_ASSOC);
    return $catList;
}

```

```

/**
 * Изменение категории(админка)
 *
 * @param $id
 * @param $options - новые параметры
 * @return bool
 */
public static function editCategory ($id, $options) {
    $db = Db::getConnection();
    $sql = " UPDATE category SET name = :name, sort_order = :sort_order, status = :status
WHERE id = :id "; $res = $db->prepare($sql);
    $res->bindParam(':name', $options['name'], PDO::PARAM_STR);
    $res->bindParam(':sort_order', $options['sort_order'], PDO::PARAM_INT);
    $res->bindParam(':status', $options['status'], PDO::PARAM_INT);
    $res->bindParam(':id', $id, PDO::PARAM_INT);
    return $res->execute();
}
}

```

Листинг Б.8 – Модель для работы с заказами

```

<?php
/**
 * Модель для работы с заказами
 */
class Order {
    /**
     * Сохранение заказа пользователя в БД
     *
     * @param $userName
     * @param $userPhone
     * @param $userText
     * @param $userId
     * @param $productsInCart
     * @return bool
     */
    public static function save ($userName, $userPhone, $userText, $userId,
    $productsInCart)
    {
        $db = Db::getConnection();
        //Преобразовываем массив товаров в строку JSON $productsInCart =
    json_encode($productsInCart);
        $sql = " INSERT INTO product_order(user_name, user_phone, user_comment, user_id,
    products) VALUES (:userName, :userPhone, :userText, :userId, :products) "; $res = $db-
    >prepare($sql);
        $res->bindParam(':userName', $userName, PDO::PARAM_STR);
        $res->bindParam(':userPhone', $userPhone, PDO::PARAM_INT);
        $res->bindParam(':userText', $userText, PDO::PARAM_STR);
        $res->bindParam(':userId', $userId, PDO::PARAM_INT);
        $res->bindParam(':products', $productsInCart, PDO::PARAM_STR);
        return $res->execute();
    }
}
/**

```

```

*
* @return array
*/
public static function getOrdersList () {
    $db = Db::getConnection();
    $sql = " SELECT id, user_name, user_phone, DATE_FORMAT(`date`, '%d.%m.%Y %H:%i:%s')
AS formated_date, status FROM product_order ORDER BY id DESC "; $ordersList = array();
    $res = $db->query($sql);
    $ordersList = $res->fetchAll(PDO::FETCH_ASSOC);
    return $ordersList;
}

/**
 * Строковое представление статуса заказа
 *
 * @param $status
 * @return string
 */
public static function getStatusText ($status) {
    switch ($status)
    {
        case '1' : return 'Новый';
        break;
        case '2' : return 'В обработке';
        break;
        case '3' : return 'Доставляется';
        break;
        case '4' : return 'Закрыт';
        break;
    }
}

/**
 * Выбираем заказ по его id
 *
 * @param $id
 * @return mixed
 */
public static function getOrderById ($id) {
    // Соединение с БД $db = Db::getConnection();
    $sql = "SELECT id, user_name, user_phone, user_comment, user_id, DATE_FORMAT(`date`,
'%d.%m.%Y %H:%i:%s') AS formated_date, products, status FROM product_order WHERE id = :id ";
    $res = $db->prepare($sql);
    $res->bindParam(':id', $id, PDO::PARAM_INT);
    // Выполняем запрос $res->execute();
    // Возвращаем данные return $res->fetch(PDO::FETCH_ASSOC);
}

/**
 * Изменение заказа(админка)
 *
 * @param $id
 * @param $userName
 * @param $userPhone

```

```

    * @param $userComment
    * @param $status
    * @return bool
    */
    public static function updateOrder ($id, $userName, $userPhone, $userComment, $status) {
        $db = Db::getConnection();
        $sql = " UPDATE product_order SET user_name = :name, user_phone = :phone,
user_comment = :comment, status = :status WHERE id = :id "; $res = $db->prepare($sql);
        $res->bindParam(':id', $id, PDO::PARAM_INT);
        $res->bindParam(':name', $userName, PDO::PARAM_STR);
        $res->bindParam(':phone', $userPhone, PDO::PARAM_INT);
        $res->bindParam(':comment', $userComment, PDO::PARAM_STR);
        $res->bindParam(':status', $status, PDO::PARAM_STR);
        return $res->execute();
    }

    /**
     * Удадение заказа
     *
     * @param $id
     * @return bool
     */
    public static function deleteOrderById ($id) {
        $db = Db::getConnection();
        $sql = " DELETE FROM product_order WHERE id = :id "; $res = $db->prepare($sql);
        $res->bindParam(':id', $id, PDO::PARAM_INT);
        return $res->execute();
    }

    /**
     * Просмотр истории заказов для пользователя(личный кабинет)
     *
     * @param $id
     * @return array
     */
    public static function getOrdersListByUserId ($id) {
        // Соединение с БД $db = Db::getConnection();
        $sql = "SELECT id, DATE_FORMAT(`date`, '%d.%m.%Y %H:%i:%s') AS formatted_date, status,
products FROM product_order WHERE user_id = $id ORDER BY id DESC "; // Выполняем запрос
        $res= $db->query($sql);
        return $res->fetchAll(PDO::FETCH_ASSOC);
    }
}
}

```

Листинг Б.9 – Модель для работы с товарами

```

<?php
/**
 * Модель для работы с товарами
 */
class Product {
    //Количество отображаемых товаров по умолчанию const SHOW_BY_DEFAULT = 6;
    /**
     * Получаем последние товары
     */
}

```

```

* @param int $page
* @return array
*/
public static function getLatestProducts ($page = 1)
{
    $limit = self::SHOW_BY_DEFAULT;
    //Задаем смещение $offset = ($page - 1) * self::SHOW_BY_DEFAULT;
    $db = DB::getConnection();
    $sql = " SELECT id, name, price, is_new FROM product WHERE status = 1 ORDER BY id
DESC LIMIT :limit OFFSET :offset "; //Подготавливаем данные $res = $db->prepare($sql);
    $res->bindParam(':limit', $limit, PDO::PARAM_INT);
    $res->bindParam(':offset', $offset, PDO::PARAM_INT);
    //Выполняем запрос $res->execute();
    //Получаем и возвращаем результат $productsList = $res->fetchAll(PDO::FETCH_ASSOC);
    return $productsList;
}

/**
 * Выводим товары по выбранной категории
 *
 * @param $catId id категории
 * @param $page - текущая страница
 * @return array
 */
public static function getProductListByCatId ($catId, $page = 1) {
    $limit = self::SHOW_BY_DEFAULT;
    //Задаем смещение $offset = ($page - 1) * self::SHOW_BY_DEFAULT;
    $db = DB::getConnection();
    $sql = " SELECT id, name, price, is_new FROM product WHERE status = 1 AND category_id
= :category_id ORDER BY id DESC LIMIT :limit OFFSET :offset "; $res = $db->prepare($sql);
    $res->bindParam(':category_id', $catId, PDO::PARAM_INT);
    $res->bindParam(':limit', $limit, PDO::PARAM_INT);
    $res->bindParam(':offset', $offset, PDO::PARAM_INT);
    $res->execute();
    //Получение и возврат результатов $products = $res->fetchAll(PDO::FETCH_ASSOC);
    return $products;
}

/**
 * Выбираем товар по идентификатору
 *
 * @param $productId
 * @return mixed
 */
public static function getProductById ($productId) {
    $db = Db::getConnection();
    $sql = " SELECT id, name, code, price, availability, brand, description, is_new,
category_id, status FROM product WHERE id = :id "; $res = $db->prepare($sql);
    $res->bindParam(':id', $productId, PDO::PARAM_INT);
    $res->execute();
    $products = $res->fetch(PDO::FETCH_ASSOC);
    return $products;
}

/**

```

```

* Выборка товаров по массиву id
*
* @param $arrayIds
* @return array
*/
public static function getProductsByIds ($arrayIds) {
    $db = Db::getConnection();
    //Разбиваем пришедший массив в строку $stringIds = implode(',', $arrayIds);
    $sql = " SELECT id, name, code, price FROM product WHERE status = 1 AND id IN
($stringIds) "; $res = $db->query($sql);
    $products = $res->fetchAll(PDO::FETCH_ASSOC);
    return $products;
}

/**
* Выводит список всех товаров
*
* @return array
*/
public static function getProductsList () {
    $db = Db::getConnection();
    $sql = " SELECT id, name, code, price FROM product ORDER BY id ASC "; $res = $db-
>query($sql);
    $products = $res->fetchAll(PDO::FETCH_ASSOC);
    return $products;
}

/**
* Удаление товара(админка)
*
* @param $id
* @return bool
*/
public static function deleteProductById ($id) {
    $db = Db::getConnection();
    $sql = " DELETE FROM product WHERE id = :id "; $res = $db->prepare($sql);
    $res->bindParam(':id', $id, PDO::PARAM_INT);
    return $res->execute();
}

/**
* Добавление продукта
*
* @param $options - характеристики товара
* @return int|string
*/
public static function addProduct ($options) {
    $db = Db::getConnection();
    $sql = " INSERT INTO product(name, category_id, code, price, availability, brand,
description, is_new, status) VALUES (:name, :category_id, :code, :price, :availability,
:brand, :description, :is_new, :status) "; $res = $db-
>prepare($sql);
    $res->bindParam(':name', $options['name'], PDO::PARAM_STR);
    $res->bindParam(':category_id', $options['category'], PDO::PARAM_INT);
    $res->bindParam(':code', $options['code'], PDO::PARAM_INT);

```

```

$res->bindParam(':price', $options['price'], PDO::PARAM_INT);
$res->bindParam(':availability', $options['availability'], PDO::PARAM_INT);
$res->bindParam(':brand', $options['brand'], PDO::PARAM_STR);
$res->bindParam(':description', $options['description'], PDO::PARAM_STR);
$res->bindParam(':is_new', $options['is_new'], PDO::PARAM_INT);
$res->bindParam(':status', $options['status'], PDO::PARAM_INT);
//Если запрос выполнен успешно if ($res->execute())

{
    //Возвращаем id последней записи, позже, в контроллере переходим на страницу этого
    товара, если все успешно return $db->lastInsertId();
}

else {
    return 0;
}

}

/**
 * Изменение товара
 *
 * @param $id
 * @param $options
 * @return bool
 */
public static function editProduct ($id, $options) {
    $db = Db::getConnection();
    $sql = " UPDATE product SET name = :name, category_id = :category, code = :code,
price = :price, availability = :availability, brand = :brand, description = :description,
is_new = :is_new, status = :status WHERE id = :id "; $res = $db->prepare($sql);
$res->bindParam(':name', $options['name'], PDO::PARAM_STR);
$res->bindParam(':category', $options['category'], PDO::PARAM_INT);
$res->bindParam(':code', $options['code'], PDO::PARAM_INT);
$res->bindParam(':price', $options['price'], PDO::PARAM_INT);
$res->bindParam(':availability', $options['availability'], PDO::PARAM_INT);
$res->bindParam(':brand', $options['brand'], PDO::PARAM_STR);
$res->bindParam(':description', $options['description'], PDO::PARAM_STR);
$res->bindParam(':is_new', $options['is_new'], PDO::PARAM_INT);
$res->bindParam(':status', $options['status'], PDO::PARAM_INT);
$res->bindParam(':id', $id, PDO::PARAM_INT);
return $res->execute();
}

/**
 * Возвращает путь к изображению
 * @param integer $id
 * @return string <p>Путь к изображению</p>
 */
public static function getImage ($id) {
    // Название изображения-пустышки $noImage = 'no-image.jpg';
    // Путь к папке с товарами $path = '/upload/images/products/';
    // Путь к изображению товара $pathToProductImage = $path . $id . '.jpg';
    if (file_exists($_SERVER['DOCUMENT_ROOT'] . $pathToProductImage))
{

```



```

        // Если изображение для товара существует // Возвращаем путь изображения товара
return $pathToProductImage;
}

// Возвращаем путь изображения-пустышки
return $path . $noImage;
}

/**
 * Возвращаем количество товаров в указанной категории
 * @param integer $catId
 * @return integer
 */
public static function getTotalProductsInCategory ($catId) {
    // Соединение с БД $db = Db::getConnection();
    // Текст запроса к БД $sql = 'SELECT count(id) AS count FROM product WHERE status="1"
AND category_id = :category_id';
    // Используется подготовленный запрос $res = $db->prepare($sql);
    $res->bindParam(':category_id', $catId, PDO::PARAM_INT);
    // Выполнение коменды $res->execute();
    // Возвращаем значение count - количество $row = $res->fetch();
    return $row['count'];
}

/**
 * Общее кол-во товаров в магазине
 *
 * @return mixed
 */
public static function getTotalProducts () {
    // Соединение с БД $db = Db::getConnection();
    // Текст запроса к БД $sql = "SELECT count(id) AS count FROM product WHERE status=1
";
    // Выполнение коменды $res = $db->query($sql);
    // Возвращаем значение count - количество $row = $res->fetch();
    return $row['count'];
}
}
}

```

Листинг Б.10 – Модель для работы с пользователями

```

<?php
/**
 * Модель для работы с пользователями
 */
class User {
    /**
     * Если в контроллере все ОК, принимаем данные и записываем в БД
     *
     * @param $name имя
     * @param $email email
     */
}
}

```

```

* @param $password пароль
* @return bool возвращает true/false
*/
public static function register ($name, $email, $password)

{
    $db = Db::getConnection();
    $sql = " INSERT INTO user(name, email, password) VALUES(:name, :email, :password) ";
$res = $db->prepare($sql);
    $res->bindParam(':name', $name, PDO::PARAM_STR);
    $res->bindParam(':email', $email, PDO::PARAM_STR);
    $res->bindParam(':password', $password, PDO::PARAM_STR);
    return $res->execute();
}

/**
 * Проверяем поле Имя на корректность
 *
 * @param $name
 * @return bool
 */
public static function checkName ($name) {
    if (strlen($name) > 1)

{
    return true;
}

return false;
}

/**
 * Проверяем поле Телефон на корректность
 *
 * @param $phone
 * @return bool
 */
public static function checkPhone ($phone) {
    if (strlen($phone) > 9)

{
    return true;
}

return false;
}

/**
 * Проверяем поле Пароль на корректность
 *
 * @param $password
 * @return bool
 */
public static function checkPassword ($password) {
    if (strlen($password) >= 6)

```

```

{
    return true;
}

return false;
}

/**
 * Проверяем поле Email на корректность
 *
 * @param $email
 * @return bool
 */
public static function checkEmail ($email) {
    if (filter_var($email, FILTER_VALIDATE_EMAIL))

{
    return true;
}

return false;
}

/**
 * Проверем email на доступность
 *
 * @param $email
 * @return bool
 */
public static function checkEmailExists ($email) {
    $db = Db::getConnection();
    $sql = " SELECT count(*) FROM user WHERE email = :email "; $res = $db->prepare($sql);
    $res->bindParam(':email', $email, PDO::PARAM_STR);
    $res->execute();
    if ($res->fetchColumn()) return true;
    return false;
}

/**
 * Проверка на существование введенных данных при аауризации
 *
 * @param $email
 * @param $password
 * @return bool
 */
public static function checkUserData ($email, $password) {
    $db = Db::getConnection();
    $sql = " SELECT id, name, email, password, role FROM user WHERE email = :email ";
$res = $db->prepare($sql);
    $res->bindParam(':email', $email, PDO::PARAM_INT);
    $res->execute();
    $user = $res->fetch();
    if (password_verify($password, $user['password']))

{
    return $user['id'];
}

```

```

}

return false;
}

/**
 *Запись пользователя в сессию
 *
 * @param $userId
 */
public static function auth ($userId) {
    $_SESSION['user'] = $userId;
}

/**
 * Проверяем, авторизован ли пользователь при переходе в личный кабинет
 *
 * @return mixed
 */
public static function checkLog () {
    //Если сессия есть, то возвращаем id пользователя if (isset($_SESSION['user']))

    {
        return $_SESSION['user'];
    }

header('Location: user/login');
}

/**
 * Проверяем наличие открытой сессии у пользователя для
 * отображения на сайте необходимой информации
 *
 * @return bool
 */
public static function isGuest () {
    if (isset($_SESSION['user']))

    {
        return false;
    }

return true;
}

/**
 * Вытягиваем информацию о пользователе по id
 *
 * @param $userId
 * @return mixed
 */
public static function getUserById ($userId) {
    $db = Db::getConnection();
    $sql = " SELECT id, name, email, password, role FROM user WHERE id = :id "; $res =
$db->prepare($sql);
    $res->bindParam(':id', $userId);

```

```
        $res->execute();
        return $res->fetch(PDO::FETCH_ASSOC);
    }

    /**
     * редактируем информацию из личного кабинета
     *
     * @param $userId
     * @param $name
     * @param $password
     * @return bool
     */
    public static function edit ($userId, $name, $password) {
        $db = Db::getConnection();
        $sql = " UPDATE user SET name = :name, password = :password WHERE id = :id "; $res =
        $db->prepare($sql);
        $res->bindParam(':name', $name, PDO::PARAM_STR);
        $res->bindParam(':password', $password, PDO::PARAM_STR);
        $res->bindParam(':id', $userId, PDO::PARAM_INT);
        return $res->execute();
    }
}
```