**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc517172761)

[1 Анализ предметной области и постановка задачи 9](#_Toc517172762)

[1.1 Понятие и сущность интернет-магазина, преимущества его использования 9](#_Toc517172763)

[1.2 Анализ систем в области управления интернет-магазином 10](#_Toc517172764)

[1.3 Классификация электронных магазинов 11](#_Toc517172765)

[1.4 Постановка задачи 12](#_Toc517172766)

[1.5 Ключевые показатели эффективности продаж в интернет-магазине 19](#_Toc517172767)

[2 Разработка алгоритма решения задачи 20](#_Toc517172768)

[2.1 Структура сайта 20](#_Toc517172769)

[2.2 Выбор сервера 22](#_Toc517172770)

[2.3 Среда локальной разработки сайтов Open Server 24](#_Toc517172771)

[2.4 Проектирование и разработка объектной модели базы данных 26](#_Toc517172772)

[2.5 Средства создания web-сайта 28](#_Toc517172773)

[3 Разработка интерфейса и программная реализация 30](#_Toc517172774)

[3.1 Разработка архитектуры системы 30](#_Toc517172775)

[3.2 Проектирование интерфейса 30](#_Toc517172776)

[4 Экономический анализ 38](#_Toc517172777)

[4.1 Исходные данные для расчета затрат 38](#_Toc517172778)

[4.2 Расчет затрат на проектирование системы 41](#_Toc517172779)

[4.3 Расчет затрат на внедрение 45](#_Toc517172780)

[4.4 Расчет эксплуатационных расходов 46](#_Toc517172781)

[4.5 Расчет и построение диаграммы общей стоимости системы 47](#_Toc517172782)

[4.6 Расчет прибыли разработчика системы 48](#_Toc517172783)

[4.7 Расчет затрат, связанных с покупкой, внедрением и использованием системы и срока окупаемости 48](#_Toc517172784)

[4.8 Выводы 50](#_Toc517172785)

[5 Безопасность и экологичность проекта 51](#_Toc517172786)

[5.1 Обоснование важности охраны труда 51](#_Toc517172787)

[5.2 Характеристика условий труда программиста 52](#_Toc517172788)

[5.3 Анализ опасных и вредных производственных факторов при работе с компьютером 52](#_Toc517172789)

[5.4 Общие требования при работе за компьютером 54](#_Toc517172790)

[6 Тестирование 58](#_Toc517172791)

[6.1 Цели и основные методы тестирования web-сайта 58](#_Toc517172792)

[6.2 Функциональное тестирование сайта и тестирование пользовательского интерфейса (UI-интерфеса) 58](#_Toc517172793)

[6.3 Тестирование безопасности сайта 64](#_Toc517172794)

[ЗАКЛЮЧЕНИЕ 66](#_Toc517172795)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 67](#_Toc517172796)

[ПРИЛОЖЕНИЕ А](#_Toc517172797) [Исходный код приложения 69](#_Toc517172798)

## ВВЕДЕНИЕ

С каждым годом количество пользователей интернет неуклонно растет. И это неудивительно, ведь интернет дает нам обилие возможностей. Одним из главных его достоинств является доступность огромного количества информации. Не выходя из дома, имея только персональный компьютер и доступ в интернет, можно совершать путешествия по другим странам, изучать историю, читать интересующие книги, смотреть кино, рассматривать миллионы фотографий! Но это не все, что может дать нам всемирная сеть, она также очень упрощает нашу жизнь. Теперь не нужно стоять в очередях, чтобы купить билет или оплатить коммунальные платежи, все это можно быстро и безопасно сделать прямо со своего рабочего места.

Интернет, как средство массовой информации существует достаточно давно. Для большинства людей, использующих Интернет, это просто средство для получения какой-либо информации, либо средство коммуникации. Одним из лежащих на поверхности способов использования Интернета для бизнеса является Интернет-продажа товаров и услуг.

Интернет-магазин очень удобная система демонстрации и продажи товаров и услуг в Интернет. Интернет-магазин подходит для размещения большого количества информации, позволяет оперативно обновлять ассортимент, четко контролировать рабочие процессы. Например, автоматически обновлять прайс-листы. Причем Интернет-магазин может быть как продолжением традиционного бизнеса, так и совершенно независимой структурой. В любом случае, при правильной организации работы, он будет приносить прибыль.

С развитием интернета неизбежным стало появление и интернет-магазинов. Становятся доступны любые товары из любой точки планеты. Одежда, бытовая техника, книги, электроника, продукты, все, что угодно и все, что нужно теперь можно покупать прямо из дома. Основная сложность - выбрать конкретную модель среди множества аналогов. Но используя поисковики не так уж сложно найти лучшую. Всевозможные форумы помогут получить мнение пользователей о товаре, на которое обычно можно полагаться. Заодно можно узнать об интересных тонкостях и подробностях, или увидеть описание более интересного товара. Магазины экономят время и деньги.

Преимущества, которые даёт интернет-магазин:

* интернет-магазин работает 24 часа в сутки, 365 дней в году, без перерыва на обед, без выходных и праздничных дней;
* доступ к виртуальным витринам магазина может получить любой покупатель, находящийся в любой точке планеты на любом континенте;
* профессионально грамотно созданный интернет-магазин может работать полностью автономно. практически без обслуживания;
* интернет-магазин не имеет ограничений на виртуальную площадь, можно разместить сколь угодно много товаров или описать любое количество услуг;
* владелец интернет-магазина может сдавать в аренду свои виртуальные торговые площади так же, как и владелец обычного магазина;
* интернет-магазин позволяет общаться с потенциальными клиентами, находящимися сколь угодно далеко, в режиме реального времени;
* срок и стоимость создания интернет-магазина несоизмеримо ниже, чем обычного магазина;
* для создания интернет-магазина не требуется получения многочисленных разрешений и лицензий; его не проверяет пожарный инспектор, санитарно эпидемиологическая станция и другие хорошо знакомые вам службы;
* развитие мобильных устройств для доступа в интернет (мобильный телефон, карманный персональный компьютер и т.д.) позволяет получить доступ к интернет-магазину из любой точки мира, вне зависимости от местоположения потенциального клиента.

К сожалению, недостатков у интернет-магазинов тоже хватает. Самый, пожалуй, главный - невозможность увидеть товар вживую. Фотографии не всегда бывают информативны. Еще один недостаток, как это ни странно - избыток информации. Мнения об одном товаре у разных людей может кардинально различаться и сделать выбор становится гораздо сложнее. Цена и доставка тоже могут оставить неприятный осадок. Стоимость товаров обычно ниже, чем в обычных магазинах, но товар еще нужно доставить покупателю. Никто не захочет получить побитый, помятый, испачканный товар. А все это вполне может произойти в процессе доставки. Существуют специальные коммерческие службы, но стоимость их услуг немаленькая. В результате чего может получиться, что сумма заказа невелика, но стоимость доставки сведет на нет всю выгоду. С доставкой связана и еще одна сложность - время. Если в обычном магазине сразу после покупки товар оказывается у вас, то при покупке через интернет вам придется подождать, пока заказ до вас доберется.

Для разработчика создание такого магазина интересная и прибыльная, но и достаточно сложная задача.

# 1 Анализ предметной области и постановка задачи

## 1.1 Понятие и сущность интернет-магазина, преимущества его использования

Много лет прошло с тех пор, как появилась Всемирная сеть Интернет. С течением времени изменилось многое: технические возможности Сети расширились, аудитория выросла, цели и задачи, для решения которых была создана Сеть, перестали быть исключительно военными или научными. Современный Интернет - это огромное хранилище, в котором каждый по своему желанию может найти нужный текст, музыку, учебные материалы, а с некоторого времени и просто купить все, что необходимо - от продуктов, до программного обеспечения. В данное время многие дальновидные руководители компаний совершенно справедливо считают необходимым создание небольшого интернет-представительства компании. Другими словами все больше появляется коммерческих сайтов, основной целью которых является продвижение товаров и услуг с помощью интернета.

Вот уже несколько лет в мире успешно развивается электронная коммерция. Как и в обычной коммерции, в электронной коммерции можно выделить два направления: межфирменное деловое партнерство и розничная торговля. Первое направление развивается довольно успешно, так как многие фирмы и компании поняли, что заключать договора и вести переговоры с помощью технологий Интернет гораздо быстрее и дешевле, нежели обычными способами. Что касается розничной торговли, то в Луганской Народной Республике она находится на начальном этапе развития по сравнению с аналогичным направлением в мире. Однако уже сейчас можно сказать, что через некоторое время конкуренция в розничной торговле через Интернет станет столь же напряженной, как и в обычной розничной торговле.

Однако, что же представляет собой интернет-магазин, как он создается, в чем его преимущества и в чем недостатки? Прежде всего, Интернет-магазин - это сайт, содержащий подробный каталог товаров с описанием и изображением. Основное отличие от обычного интернет-каталога состоит в том, что товары, представленные в интернет-магазине можно не только увидеть, но и заказать, не вставая с места и не прерывая увлекательного путешествия по просторам Интернет.

## 1.2 Анализ систем в области управления интернет-магазином

Для создания интернет-магазина существуют десятки разнообразных решений по функциям. Для того чтобы охарактеризовать ту или иную систему управления контентом (content management system - далее CMS), которая может использоваться для создания интернет-магазина, необходимо определить критерии, которым она должна соответствовать. Основная задача интернет-магазина - это продажа товара, которую решает любая платформа для Интернет-торговли, однако существует ряд функций и характеристик, без которых интернет-магазин не сможет выдержать конкуренции в web-пространстве. К таким функциям можно отнести покупку «в один клик», которая избавляет пользователя от процедуры регистрации при оформлении того или иного товара.

Система управления контентом должна отвечать требованиям производительности. Большинство современных решений для создания интернет-магазинов разработаны популярным языком программирования PHP. Несмотря на одинаковую языковую основу, различным системам необходимо разное количество ресурсов для работы.

Немаловажной характеристикой является возможность расширения и дополнения существующего решения интернет-магазина, поскольку какой бы продуманной система не была, со временем ее функционала может не хватать. Поэтому важным является понятность и валидность при написании движка CMS.

## 1.3 Классификация электронных магазинов

Классифицировать электронные магазины можно по различным критериям. Наиболее интересной классификацией является классификация по модели бизнеса:

* чисто онлайновый магазин;
* совмещение офф-лайнового бизнеса с он-лайновым (когда интернет-магазин был создан на основе уже действующей реальной торговой структуры).

По отношению с поставщиками:

* имеют собственный склад (наличие реальных товарных запасов);
* работают по договорам с поставщиками (отсутствие значительных собственных запасов).

Классификация по товарному ассортименту – книги, аудио, видеокассеты, CD-диски, DVD-диски, компьютерная, бытовая техника, мобильные телефоны.

Среди методов розничной продажи товаров в Сети можно выделить:

* интернет - магазины (автоматические магазины);
* web-витрины;
* торговые автоматы.

Интернет-витрина – скорее это рекламный сервер. На витрине выкладывают информацию о товарах, которую постоянно обновляют. Затраты на ее создание и администрирование могут быть довольно низкими, а практическая польза такой витрины очевидна. Но это еще не торговля. Потенциальный покупатель, посетив витрину, должен позвонить на фирму, оплатить товар, договориться о доставке. Поэтому Интернет-витрина оправданна в тех случаях, когда покупателя надо познакомить со сложной продукцией, на изучение которой в торговом зале у него уйдет слишком много времени.

Интернет-витрина может быть размещена где угодно - на собственном сервере, на сервере провайдера, на сервере, предоставляющем бесплатные страницы. Для работы с витриной достаточно иметь подключение через телефонную линию и минимум навыков работы с HTML.

Торговый автомат может не только выполнять функции витрины, но и принимать заказы и передавать их менеджеру, то есть оформлять заказы и выписывать счета на оплату без присутствия покупателя. Торговый автомат реально торгует и по соотношению затрат к результату наиболее предпочтителен для пилотных и тестовых проектов с небольшим потоком покупателей. Торговый автомат, так же как и интернет-витрину, можно разместить и на своем сервере, и на сервере провайдера. Однако его создание и администрирование требует навыков и определенной квалификации.

Автоматический магазин – эффективное и комплексное решение в торговом бизнесе. Он не только выписывает счета, но и отслеживает заказы, принимает электронные платежи и формирует заявки на доставку товаров покупателям. Здесь задача менеджера – контролировать работу системы, сложную в обслуживании.

## 1.4 Постановка задачи

Прежде чем браться за выполнение той или иной работы необходимо четко понимать цели, которые необходимо достигнуть, выполняя ее, т.е. ответить на вопрос - зачем. В нашем случае вопрос будет звучать так: зачем нашей организации нужен сайт? Ответ на данный вопрос очень часто сводится к общим размышлением на тему, а зачем вообще нужны сайты? Согласитесь, что это два принципиально разных вопроса, поскольку в первом случае он имеет прямое отношение к области деятельности организации, а второй - простой обмен мнениями.

Для того чтобы ответить на данный вопрос необходимо рассмотреть все аспекты деятельности организации и окружающей ее среды, поскольку только в этом случае возможно эффективное планирование и создание сайта. Сначала следует произвести предварительный сбор, обработку и анализ информации, которая может пригодиться для дальнейшей работы. Особенно следует уделить внимание анализу уже существующих решений других организаций в данной области, стараясь найти связь и различия между реальным и виртуальным поведением рассматриваемой модели или поведения для определения удачных и неудачных решений. Полагаю, что вы не раз встречались с "заброшенными" сайтами, на которых информация не обновляется годами. И это, заметьте, произошло не только потому, что в данной организации нет специалиста необходимого уровня, а главным образом потому, что руководство самой организации не понимало цели его создания и затраты, необходимые для достижения поставленных целей.

Обратите внимание на то, что необходимо рассматривать не только деятельность и интересы самой организации, но и деятельность внешней среды - потребителей, поставщиков, конкурентов и т.п., работа которых в большей или меньшей степени влияет на деятельность самой организации. Простой пример. Организация заказала сайт с обычным статическим каталогом товаров без фотографий, описаний и механизмом поиска товаров. Как увеличится эффективность ее работы? Это будет не значительный прирост, поскольку сотрудники организации будут вынуждены постоянно отвечать на вопросы клиентов по поводу различных характеристик товаров по телефону или по электронной почте. Как увеличится эффективность работы клиентов – это будет не значительное увеличении, поскольку им необходимо будет производить дополнительные действия для получения информации. С другой стороны, грамотно разработанный интернет-магазин и автоматизированная система учета и обработки заказов может существенно увеличить производительность их работы. Вероятность потери потенциальных клиентов в Интернете намного выше, чем в случае розничной торговли, поскольку посетитель может уйти на другой сайт (к конкуренту) без покупки "замечательного со всех сторон" товара. Тогда стоит ли разрабатывать сайт только потому, что он есть у конкурентов или потому, что так принято?

В этом примере затрагивается очень важный аспект любой деятельности на рынке товаров и услуг - сервисное обслуживание и его уровень, о чем будет сказано несколько позже.

После предварительной подготовки и определения целей, можно перейти к постановке задачи, которая, в нашем случае, может звучать, что обходимо разработать сайт фирмы. Он должен предоставлять посетителям информацию о предлагаемых фирмой товарах и контактную информацию, проект должен быть разработан в кратчайшие сроки и с минимальными затратами, или так необходимо разработать интернет-представительство организации, которое будет нацелено на продвижение ее торговой марки, расширению целевого рынка и организации сети региональных представительств. Для этого необходимо:

* предоставить информацию о фирме, ее деятельности, режиме работы, новых поступлениях, новостях и др;
* разместить рубрики в каталог товаров с фотографиями и описаниями;
* обеспечить возможность поиска и заказа товаров;
* обеспечить информационную поддержку её партнерам, представителям и др.

Можно сказать, что качество постановки задачи влияет на дальнейшее планирование проекта и зависит как от четкости ее формулировки, так и от подхода к ее составлению.

Постановка задачи производится с постепенным уточнением задач от общего к частному, т.е. сверху вниз. Обратный подход позволит произвести более четкую группировку подзадач по каждому направлению. Совместное применение данных подходов позволит более четко увидеть общую картину общего плана задач.

В итоге может получиться достаточно большой иерархический список, который необходимо обязательно составить в письменной форме для дальнейшей работы - составления плана работ, бюджета, графика работ и т.д.

Очень часто на этом этап предварительного планирования будущего ресурса и заканчивается, что приводит к следующим последствиям:

1. Огромные затраты усилий и времени на обновление проекта и его расширение - постановка задачи без учёта перспектив развития организации и ее сайта в условиях постоянного изменения внешней среды как результат неопределенности в общей стратегии развития и недоработки тактического планирования.
2. Слабая отдача от работы ресурса - не была учтена специфика деятельности фирмы и область её работы, не была изучена целевая аудитория посетителей и потребителей, бессистемная информационная и рекламно-маркетинговая стратегия и др.
3. Медленная работа ресурса - непродуманная рекламная политика, в результате которой произошла информационная перегрузка страниц. Просчёты в выборе подходящей технологии - не был произведён анализ аналогичных проектов в сети, их работа, достоинства и недостатки. Не рассматривались вопросы повышения эффективности работы ресурса и ожидаемая нагрузка на Web-сервер.
4. Отсутствие заказов с сайта - интерфейсные ошибки и недочёты в работе механизма всего ресурса из-за поверхностного рассмотрения специфики и ограничений Web-среды, информационной и функциональной архитектуры и т.п.
5. Увеличение нагрузки на различные отделы организации, из-за непродуманных и неэффективных механизмов работы всей информационной системы организации и слабой связи между её отдельными составляющими.
6. Потеря марочной идентичности, капитала бренда, связи между реальным и виртуальным образом фирмы и т.д.

Поскольку, в общем случае, процесс планирования и разработки сайта происходит последовательно, то ошибки на одном этапе будут влиять на все последующие этапы, в худшем случае суммируясь с ними. Для устранения этого применяются петли обратной связи, которые охватывают несколько этапов проектирования и позволяют пройти их снова для устранения или уменьшения возникших в предыдущем варианте ошибок и недочётов. Понятно, что предварительный этап сбора и анализа информации, определения целей и постановки задачи является стратегическим, т.е. именно здесь закладывается базовая концепция всего проекта и его перспективы дальнейшего развития. Техническая сторона проекта может изменяться, поскольку технологии не стоят на месте и постоянно развиваются, однако именно стратегические цели и являются теми опорными точками, на которые ориентирована деятельность всей организации.

Для любого пользователя интернет-магазина главными преимуществами являются скорость загрузки и простота навигации. К сожалению, иногда бывает сложно разобраться в запутанной иерархии сайта и найти нужную вещь. Ещё одним неприятным явлением иногда бывает плохой дизайн - слишком мелкий текст или яркие, кричащие цвета, от которых устают глаза.

Итогом разработки должен стать сайт интернет-магазина. В качестве сферы деятельности были выбраны компьютерные комплектующие и сопутствующие им товары.

Очевидно, что была необходимость разработать несколько интерфейсов: для посетителя, для пользователя, для менеджеров и для администратора. Доступ к каждому интерфейсу открыт в соответствии с полномочиями.

Интерфейс посетителя самый упрощенный. Посетитель может просматривать информацию главной страницы, список товаров и сами товары. Купить товар посетитель сможет только зарегистрировавшись, тем самым перейдя в категорию пользователей.

Интерфейс пользователя должен давать возможность совершать покупки, а так же открывать доступ к личному кабинету, где будут отображаться корзина, подготовленные заказы, а так же история выполненных заказов.

Интерфейс менеджеров должен предусматривать всевозможные операции с товарами, категориями товаров, описаниями. Помимо этого, необходимо организовать работу с заказами клиентов, т.е. предоставить необходимую информацию для оформления и отгрузки заказа.

Интерфейс администратора должен позволять управлять правами пользователей (покупатель/менеджер), а так же учетными записями пользователей.

Помимо текстовой информации о товар необходимо предоставлять покупателю еще и графическую - фотографии товаров. Необходимо так же оформить сайт в спокойном неагрессивном стиле, не отвлекающем пользователя от основной задачи - совершения покупок.

Требуется предусмотреть проверку и разграничение прав доступа к различному контенту сайта в зависимости от прав пользователя.

Для любого интернет-магазина важно, чтобы пользователи регистрировались на сайте и оставляли о себе информацию. Это позволяет увеличить средний чек и сильно повышает шансы интернет-магазина превратить разового покупателя в постоянного.

Сама регистрация должна быть очень простой, чтобы не стать барьером для пользователя, а интернет-магазин должен стимулировать пользователей зарегистрироваться. Некоторые крупные интернет-магазины запрещают совершать покупки незарегистрированным пользователям – это довольно спорный способ заставить зарегистрироваться, но он показывает важность регистрации на сайте. Мотивация зарегистрироваться может быть самой разной: специальная скидка для зарегистрированных пользователей, доступ к любым акциям только для зарегистрированных пользователей, особые условия зарегистрированным пользователям и т.д.

Еще одна важная часть любого интернет-магазина, без которой он не может работать – корзина. С корзиной покупатель сталкивается в середине своего путешествия: он уже выбрал товар, но еще не оформил покупку. Этот шаг на пути к покупке крайне важен, более 30% пользователей кладут товар в корзину, но так и не оформляют покупку по разным причинам. Тут наша задача помочь завершить покупку.

Обычно корзина имеет вид небольшого блока в правой части шапки, в котором указана маленькая пиктограмма, количество товаров в корзине и общая сумма. На блок можно нажать и при нажатии пользователь попадает внутрь корзины. Реже корзину могут делать в других местах, например, полоса внизу экрана, которая остается на месте при прокрутке.

В ходе проектирования сайта была построена Диаграмма прецедентов (см. рисунок – 1.1), для отображения взаимодействия пользователей с различными функциями сайта.

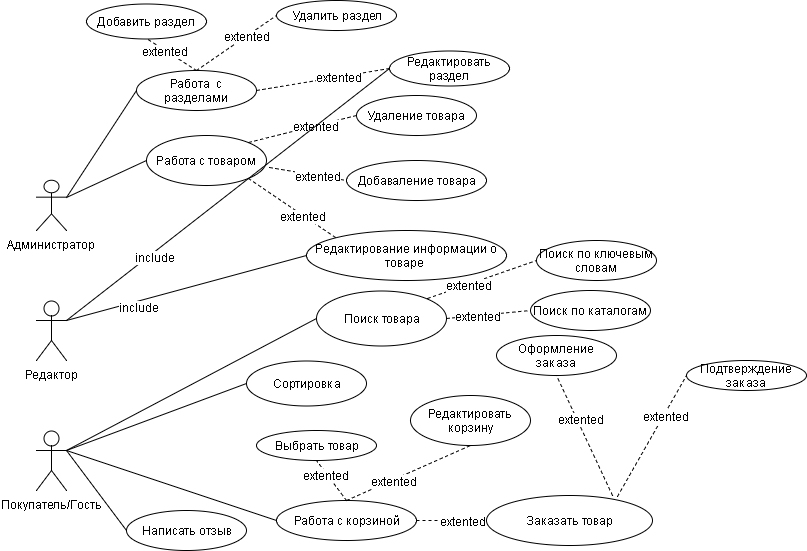


Рисунок 1.1 – Диаграмма прецедентов

**1.5 Ключевые показатели эффективности продаж в интернет-магазине**

Ключевые показатели эффективности показывают насколько быстро и эффективно растет (или падает) ваш доход. Одним из преимуществ ведения интернет-магазина является прозрачность отслеживания ключевых показателей эффективности и возможность оптимизации процессов для роста дохода.

Ключевые показатели эффективности:

* посещаемость сайта;
* просмотры товарных страниц;
* среднее время пребывания на сайте и среднее количество просмотренных страниц;
* страницы выхода(когда посетители покидают ваш сайт);
* каналы привлечения посетителей;
* показатель конверсии;
* показатель возврата посетителей.

# 2 Разработка алгоритма решения задачи

## 2.1 Структура сайта

Отличительной особенностью качественного web-сайта является его правильно организованная структура. Если вы спланируете ее до создания проекта, тогда вы сможете избежать многих проблем и ошибок в будущем.

Структура сайта – это логическая разметка и физическая связка страниц сайта, а так же расположение видимых элементов дизайна, обусловленная стандартами разработки сайтов. Разделяют внешнюю и внутреннюю структуру.

Самый «древний» вид структур сайта, ведущий свою историю с самого появления Интернета – линейный. Именно благодаря времени своего существования он является довольно простым. В такой структуре все странички равноправны в своем значении, они не делятся по разделам, а просто послушно следуют друг за дружкой в строго заданной последовательности

Более сложный вариант такого вида структур сайта – линейный с ответвлениями. Он подразумевает различные ветви для разного рода посетителей. То есть, для одной группы лиц представлена одна цепочка информации, для другой, соответственно, другая.

Более прогрессивный вид структур сайта – древовидный. Как Вы понимаете, здесь может существовать столько разделов и страниц, сколько понадобится web-мастеру. Такую структуру Вы можете встретить на большом количестве информационных ресурсов и не только. Тут важно проследить за навигацией, сделав ее более подробной, а также периодически ухаживать за «веточками» своей структуры, вовремя удаляя устаревшие материалы и добавляя новые в подходящие для них разделы.

Самый редкий в применении вид структур сайта – решетчатый. Он самый сложный в исполнении, да и затраченные на его реализацию труды в большинстве случае себя не оправдывают, так как такая структура удобна лишь для директорий. Именно на их примере Вы можете ознакомиться с этим видом структур сайта, и, как Вы заметите, он очень похож с предыдущим вариантом, но здесь присутствует навигация, отвечающая за мгновенный переход с одного конца ресурса на другой.

Для сайта «Сервисный центр и интернет-магазин компьютерной техники» была выбрана древовидная структура для удобного размещения информации на сайте, а также для последующего улучшения его внешнего вида. В разрабатываемом информационном интернет-сайте содержится 7 основных разделов.

Раздел «Главная» содержит яркую иллюстрацию для привлечения внимания посетителя и удержания его на сайте, так же там можно увидеть предлагаемые услуги, контактную информацию.

Раздел «О нас» содержит список предоставляемых услуги их описание.

Раздел «Магазин запчастей» содержит список товаров, комплектующих компьютера.

Раздел «Ремонт» содержит основные услуги по ремонту компьютерной техники.

Раздел «Оплата и доставка» предоставляет возможность заказать товар связанной тематики.

В разделе «Отзыв» пользователи могут оставить свой отзыв о проделанной работе.

В разделе «Контакты» содержатся основные контакты и форма для написания письма.

На рисунке 2.1 изображена логическая структура сайта.

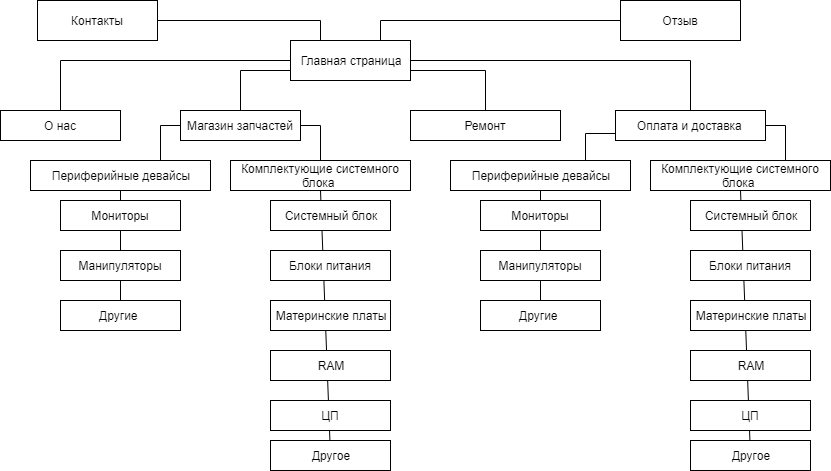


Рисунок 2.1 – Логическая структура сайта

## 2.2 Выбор сервера

В этом разделе проведем их сравнительный анализ, который позволит выбрать наиболее оптимальный интернет-сервер. За основу возьмем самые значимые характеристики серверов, а именно:

* эффективность;
* масштабируемость;
* безопасность;
* работоспособность;
* соответствие стандартам;
* гибкость;
* требования к платформе;
* управляемость.

Open Server – это портативный локальный WAMP/WNMP сервер, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов. Представленный пакет программ не является очередной любительской сборкой собранной «на коленке», это первый полноценный профессиональный инструмент, созданный специально для web-разработчиков с учётом их рекомендаций и пожеланий.

Для отладки скриптов в различном окружении Open Server предлагает на выбор сразу два вида HTTP серверов, различные версии PHP и СУБД модулей, а так же возможность быстрого переключения между ними.

Сервер Apache является приложением интернет-сервера с открытым исходным кодом, разработанным Apache Software Foundation. Программное обеспечение сервера свободно распространяется и данная лицензия с открытым исходным кодом означает, что пользователи могут отредактировать базовый код, чтобы настроить производительность и способствовать разработке программы.

Сервер Apache использует следующие модули:

* HTTP модули: Apache 2.2.21 и Nginx 1.0.11;
* СУБД модули: MySQL 5.1.61, MySQL 5.5.20 и PostgreSQL 9.1.1;
* PHP модули: PHP 5.2.17 (IMagick 2.2.1, Zend Optimizer 3.3.3, IonCube Loader 4.0.7, Memcache 2.2.4) и PHP 5.3.9 (IMagick 2.3.0, Xdebug 2.1.3, IonCube Loader 4.0.10, Memcache 2.2.6).

Отличный набор инструментов: HeidiSQL, Adminer, PHPMyAdmin, PHPPgAdmin, PgAdmin.

В состав пакета так же включены такие дополнения как: Perl, FTP сервер, Sendmail, Memcached сервер

Apache начатый в 1995 году Робертом Маккулом, выпускником Университета штата Иллинойс, постоянно развивается под управлением Apache Software Foundation с 1999 года.

Наиболее широко используется в Интернете с 1996 года (по умолчанию для многих хостов, например, пользователей cPanel).

Для настройки сервера используют расширяемую систему загрузки динамического модуля.

Сервер использует файл .htaccess для перезаписи URL и локальной настройки сервера под определенный сайт [5].

Сервер использует файл httpd.conf для конфигурации сервера с синтаксисом типа XML и для установки глобальных серверных настроек.

Основные достоинства Apache - надежность, безопасность и гибкость настройки. Apache позволяет подключать различные модули, добавляющие в него новые возможности - например, можно подключить модуль, обеспечивающий поддержку РНР или любого другого Web-ориентированного языка программирования.

Недостатки - отсутствие удобного графического интерфейса администратора. Настройка Apache осуществляется путем редактирования его конфигурационного файла. В интернете можно найти простые конфигураторы Apache, но их возможностей явно не хватает для настройки всех функций Web-сервера.

Основные преимущества Apache сервера перед другими, аналогичными, серверами:

* сервер на Apache – бесплатный, свободно распространяемый;
* Apache поддерживает разные ОС Linux, BSD, Mac OS, Microsoft Windows;
* обладает высокой надежностью;
* по производительности Apache не уступает другим серверам;
* легко устанавливается и прост в использовании;

## 2.3 Среда локальной разработки сайтов Open Server

Вышеприведенный анализ позволил сделать выбор в пользу сервера Apache и базы данных MySQL. Таким образом, для разработки интернет-портфолио можно использовать пакет программ Open Server, позволяющий создавать и отлаживать интернет-проект на локальной машине. Затем, когда работа будет завершена, продукт можно будет разместить на существующем интернет-хостинге. Базовый пакет продукта включает в себя:

* Open Server 4.9.0;
* Apache 2.4.7;
* Nginx 1.5.7;
* MySQL 5.5.35;
* MySQL 5.6.15;
* MariaDB 5.5.34;
* MongoDB 2.4.8;
* PostgreSQL 9.3.1;
* PHP 5.5.6 (Xdebug 2.2.3, OPcache, Win32ps 1.0.2-dev, Uploadprogress 1.0.3.1, SSH2 0.12, RAR 3.0.0, Memcache 2.2.7, Mongo 1.4.5, HTTP 1.7.6dev, Excel 0.9.8, FFmpeg 0.6.3, Imagick и др.);
* ImageMagick 6.8.0-7 Q16;
* FTP FileZilla 0.9.41;
* Fake Sendmail 32;
* NNCron Lite 1.17;
* Memcached 1.2.6;
* Memcached 1.4.5;
* Unbound 1.4.21;
* Adminer 3.7.1;
* HeidiSQL 8.1;
* Webgrind 1.0;
* RockMongo 1.1.5;
* PHPMyAdmin 4.0.10;
* PHPPgAdmin 5.1;
* PHPMemcachedAdmin 1.2.2.

После установки на локальном компьютере доступен сервер Apache, на котором может работать неограниченное количество сайтов, что очень эффективно для разработки и отладки PHP-кода без загрузки его файлов на удаленный сервер.

Чтобы создать новый сайт, достаточно создать папку в директории /domains и перезапустить сервер, чтобы новые хосты были восприняты системой. Чтобы увидеть новый код в действии, достаточно сохранить файл и открыть страницу в браузере. Имя папки может иметь произвольный вид в латинице, а сама папка – вложенную папку WWW, которая содержит все файлы проекта (.html, .php, .css), а также папки всех дополнительных материалов (картинки, звуки, видеофрагменты).

## 2.4 Проектирование и разработка объектной модели базы данных

MySQL – это одна из самых популярных и самых распространенных система управления базами данных (далее СУБД) в интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет-сайтов, как небольших, так и достаточно крупных.

MySQL отличатся хорошей скоростью работы, надежностью, гибкостью. Из списка задач мы можем вывести сущности, которые имеют важные роли в нашей системе. Товары, категории, клиенты и заказы – сущности, которые можно найти почти в каждой базе данных интернет-магазина. В данном примере я покажу вам модель, содержащую только следующие сущности: клиент, заказ и товар. Определившись с сущностями, мы можем подумать над связями между ними.

База данных - совокупность связанных данных, сохраняемая в двумерных таблицах информационной системы. Программное обеспечение информационной системы, обеспечивающей создание, ведение и совместное использование баз данных, называется системой управления базами данных (СУБД). База данных представляет собой удачно организованный набор поименованных таблиц. Каждая таблица - массив (возможно, очень большой) из однородных элементов, которые называются записями. Запись может содержать в себе одно или несколько именованных полей. Число и имена полей задаются при создании таблицы. Каждое поле имеет определенный тип.

Для работы с компьютерной системой была выбрана одну из самых популярных СУБД - MySQL, потому что она проста и универсальна для большинства приложений. Для работы с ней используется язык структурированных запросов SQL (Structure Query Language), который позволяет выполнять различные операции с базами данных: создавать таблицы, помещать, обновлять и удалять из них данные, производить запросы из таблиц и т.д. [10].

Для разработки сайта мне потребовалось создать семь таблиц: categorys, feedback, goods, main\_categorys, orders\_goods, orders\_repair, users.

Таблица «goods» хранит информацию о товаре, а таблицы «categorys» и «main\_categorys» хранятся доступные категории товара, в нее вносятся данные, относящиеся к какой-либо категории товара. Данные в таблицы «categorys», «main\_categorys» и «goods» заносятся только администратором.

Таблица category содержит в себе все возможные категории на которые ссылается таблица goods.

Таблица «orders\_repair» и «orders\_goods» по своей сути одинаковы и содержат перечень товаров, заказанных пользователем, или заявки на ремонт, а также информацию о пользователе (фамилия, имя, номер телефона, адрес). Данные в таблицу.

Таблица «users» содержит данные обо всех зарегистрированных пользователях системы.

Таблица «feedback» хранит сообщения

Прежде чем работать с базой данных, необходимо установить с ней сетевое соединение, а также провести авторизацию пользователя. Для этого служит функция mysql\_connect() [11].

Функция mysql\_connect() устанавливает сетевое соединение с базой данных.

До того как послать первый запрос серверу MySQL, необходимо указать, с какой базой данных мы собираемся работать. Для этого и предназначена описываемая функция mysql\_select\_db.

Она уведомляет PHP, что в дальнейших операциях будет использоваться база данных $category. В противном случае будет выведено сообщение об ошибке «Нет соединения с нашей базой данных».

Теперь можно формировать и посылать запросы к базе данных. Для этого существует функция - mysql\_query (string $query [, int $link\_identifier]). Она возвращает идентификатор результирующего набора данных.

Эта функция в своем роде универсальна: она посылает MySQL-серверу запрос $query и возвращает идентификатор ответа, или результата. Параметр $query представляет собой строку, составленную по правилам языка SQL.

## 2.5 Средства создания web-сайта

Языки web-программирования – это соответственно языки, которые в основном предназначены для работы с интернет-технологиями. Языки web-программирования делятся на две группы: клиентские и серверные. Сервер: это тот компьютер, где хранится ваш сайт, и та программа, которая обрабатывает запросы браузера к какой-либо странице. Клиентские языки - программы на клиентских языках обрабатываются на стороне пользователя, как правило, их выполняет браузер. Для разработки сайта были выбраны следующие языки, для обработки клиентской части:

1. HTML (от англ. HyperText Markup Language – «язык гипертекстовой разметки») . Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства;
2. CSS ( [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Cascading Style Sheets – каскадные таблицы стилей) – [формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) описания внешнего вида документа, написанного с использованием [языка разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) [7]. Преимущественно используется как средство описания, оформления внешнего вида [web-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0), написанных с помощью [языков разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) [HTML](https://ru.wikipedia.org/wiki/HTML) и [XHTML](https://ru.wikipedia.org/wiki/XHTML), но может также применяться к любым [XML-документам](https://ru.wikipedia.org/wiki/XML), например, к [SVG](https://ru.wikipedia.org/wiki/SVG) или [XUL](https://ru.wikipedia.org/wiki/XUL).
3. JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript . JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности web-страницам.

Когда пользователь дает запрос на какую-либо страницу, то вызванная страница сначала обрабатывается на сервере, т.е выполняются все программы, связанные со страницей, и только потом возвращается к посетителю по сети в виде файла. Этот файл может иметь расширения: HTML, PHP, ASP, ASPX, Perl, SSI, XML, DHTML, XHTML.

В ходе разработки сайта, для реализации серверной части был выбран серверный язык PHP. Это распространенный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения Web-разработок и его код может внедряться непосредственно в HTML.

PHP англ. PHP: Hypertext Preprocessor – «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools – «Инструменты для создания персональных web-страниц») – скриптовый язык общего назначения, интенсивно применяемый для разработки web-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических web-сайтов[6].

MySQL – это один из множества ПО для работы с SQL базами данных.

SQL – это структурированный язык запросов, созданный для управления реляционными БД. Он обладает широким перечнем возможностей, например, создать таблицу, редактировать и удалять данные, производить запросы из таблиц и многое другое.

# 3 Разработка интерфейса и программная реализация

## 3.1 Разработка архитектуры системы

Разработанная архитектура системы на рисунке 3.1 содержит в себе следующие части:

* серверная часть обработки и расчёта данных;
* клиентская часть обработки и запроса пользовательских данных;
* база данных MySQL;
* пользовательский интерфейс.

****

Рисунок 3.1 – Архитектура системы

Архитектура системы выполнена в виде модульных частей общей системы, а также написана по единому для всех модулей принципу. Выбор такого принципа позволяет в дальнейшем производить изменение, а так же дополнение системы без каких-либо трудностей.

## 3.2 Проектирование интерфейса

Визуально привлекательный и удобный пользовательский интерфейс - ключевой показатель качества сайта. В сочетании с грамотной структурой и логичной навигацией по разделам ресурса, он привлекает посетителей и улучшает функциональность сайта. Главная задача в такой работе как проектирование web- интерфейсов - максимально упростить жизнь пользователю, сделать так, чтобы он достигал желаемый результат, затрачивая минимум усилий. Как раз для этого служат кнопки, вкладки и элементы оформления[8].

Поскольку задача UI-интерфейса - организовать максимально простой процесс взаимодействия пользователя и ресурса, выделяют некоторые общепринятые приемы улучшения использования сайта и повышения его эффективности:

* элементы интерфейса должны быть взаимосвязаны и логически структурированы;
* интерфейс должен быть выполнен в едином стиле.

Выбирают самую лаконичную и понятную форму подачи информации, ведь ее избыток сбивает пользователя с толку, а слишком многоуровневая структура усложняет навигацию по сайту.

Общая структура главной страницы представлена на рисунке 3.2

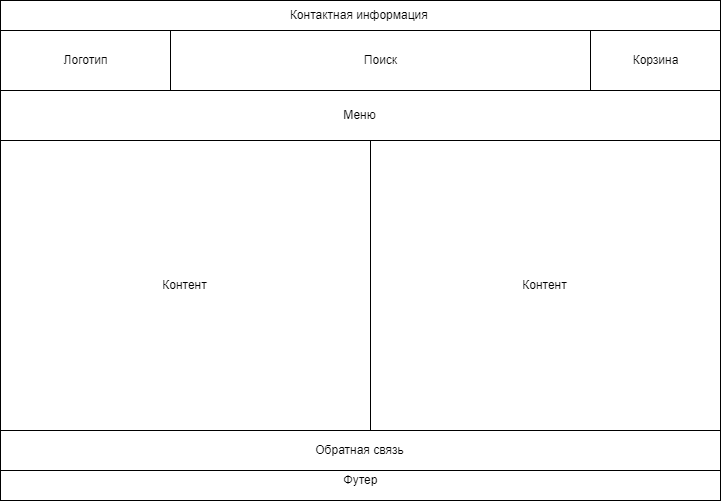


Рисунок 3.2 – Общая структура главной страницы

Сайт имеет две основных модели интерфейса:

* модель интерфейса администратора;
* модель интерфейса пользователя\гостя.

Модель интерфейса для администратора имеет права, которые дают возможность добавлять, редактировать и удалять товары. Форма добавления предоставлена на рисунке 3.3. Редактирование и удаление реализованы в виде кнопок у товара в общем каталоге и при подробном просмотре товара. На рисунках 3.4 и 3.5 изображены кнопки удаления и редактирования товара.

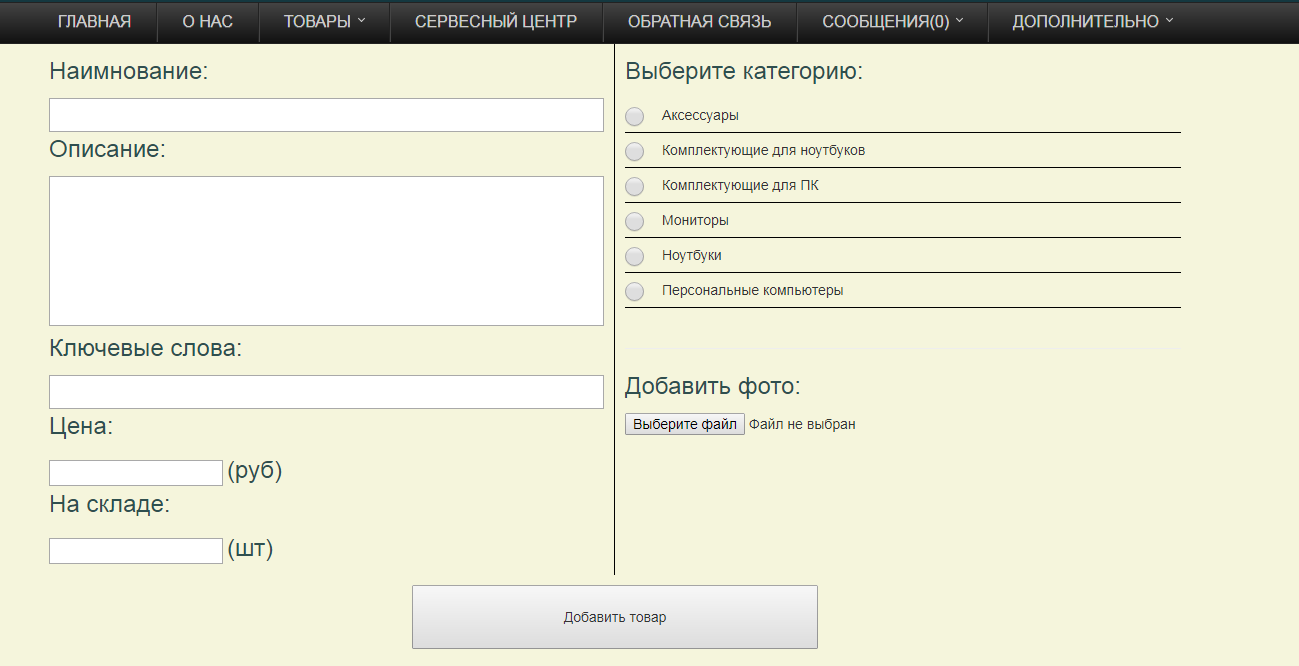
****

Рисунок 3.3 – Форма добавления товара

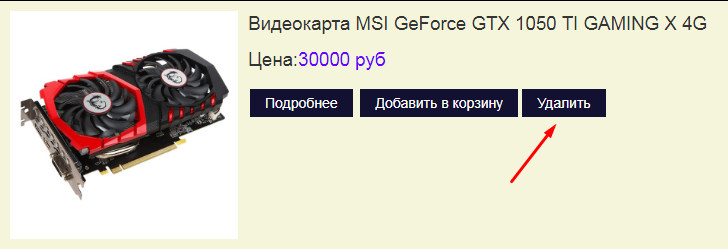
****

Рисунок 3.4 – Кнопка удаления товара

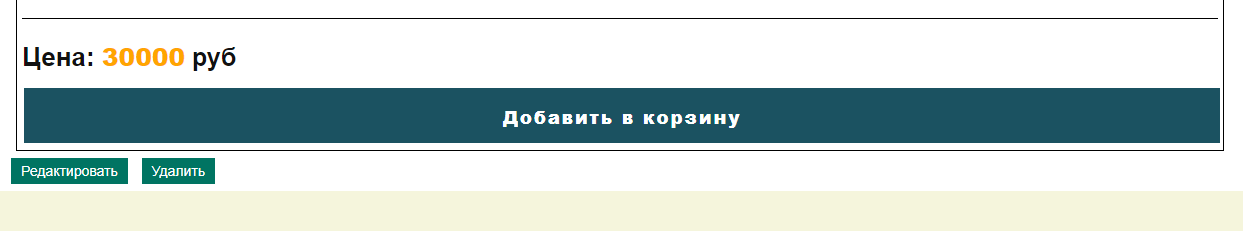
****

Рисунок 3.5 – Кнопки удаления и редактирования товара

А так же пользователю с правами администратора даются возможности работы с каталогом. Сам каталог является пунктом меню «Товары» с выпадающим списком товаров. Пример интерфейса каталога изображен на рисунке 3.6.

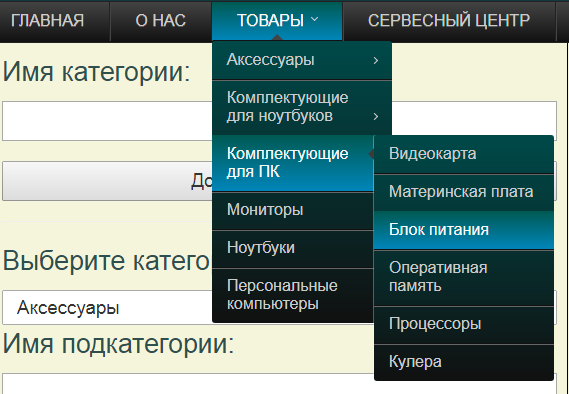
****

Рисунок 3.6 – Каталог товаров

Так же администратор может добавлять и удалять пункты меню. Пример формы добавления\удаления изображен на рисунке 3.7.

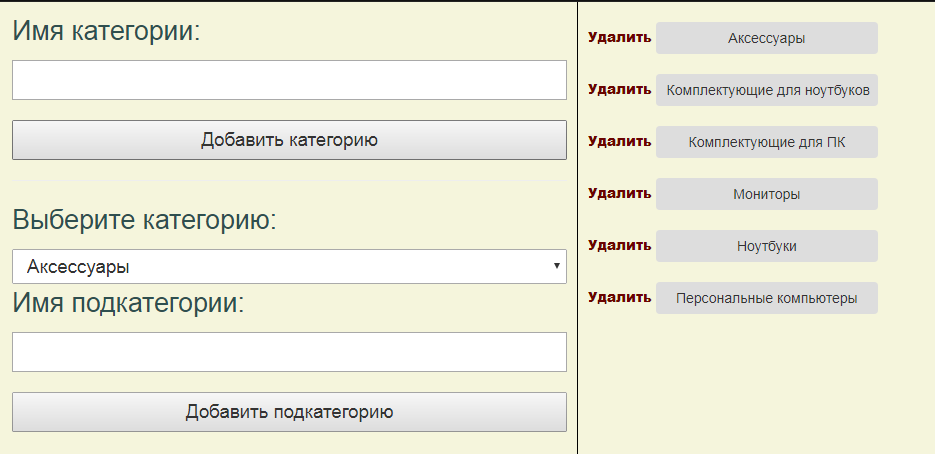


Рисунок 3.7 – Форма редактирования каталога

Модель интерфейса пользователя и гостя не имеют особых отличий. Сайт и тем и другим предоставляет возможность покупок, просмотра каталога, использование обратной связи. Основное отличие пользователя от гостя заключается в том, что пользователь имеет свою учетную запись и может вносить контактную информацию в свой профиль, которая хранится в базе данных, что в свою очередь упрощает оформление\заказ товара за счет того, что не нужно по несколько раз вводить информацию о себе, типа адреса электронной почты, номера телефона, адреса доставки и т.п. Пример учетной записи пользователя изображен на рисунке 3.8.

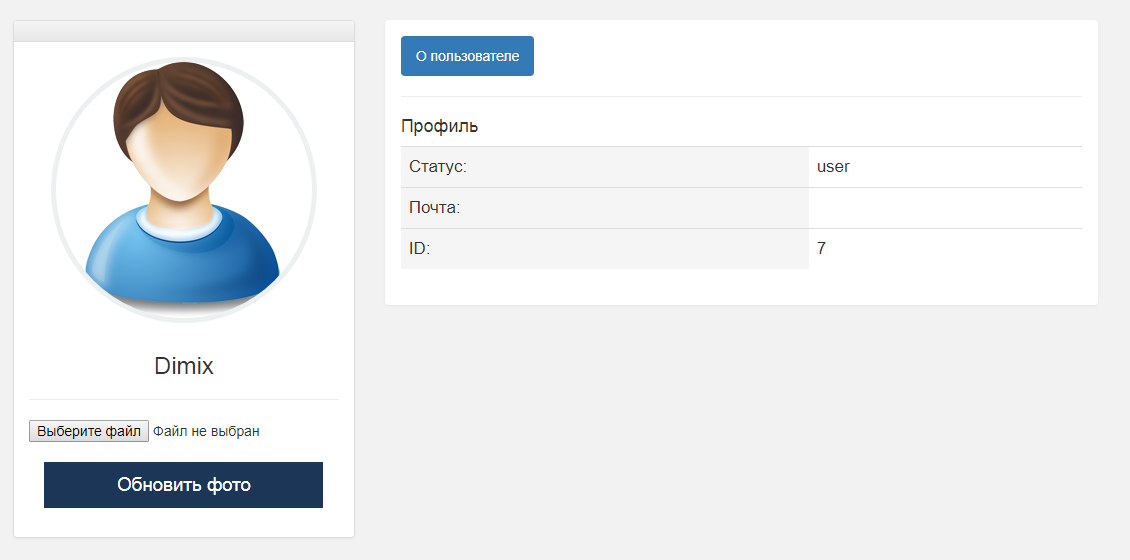
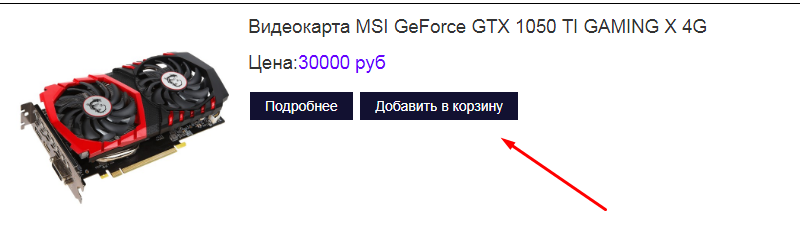
****

Рисунок 3.8 – Профиль пользователя

Как в любом нормальном интернет-магазине пользователям сайта предоставляется возможность выбирать и заказывать товар. На рисунках 3.9 и 3.10 предоставлен скриншот кнопок добавления товара в корзину. На рисунке 3.9 скриншот в общем списке товаров, а на рисунке 3.10 скриншот, при подробном просмотре товара.

Рисунок 3.9 – Кнопка добавления в корзину, в общем списке товаров

****

Рисунок 3.10 - Кнопка «добавления в корзину, при подробном просмотре

Скриншот корзины предоставлен на рисунке 3.11.

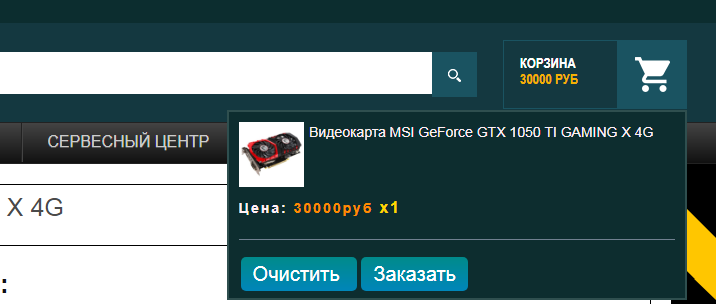
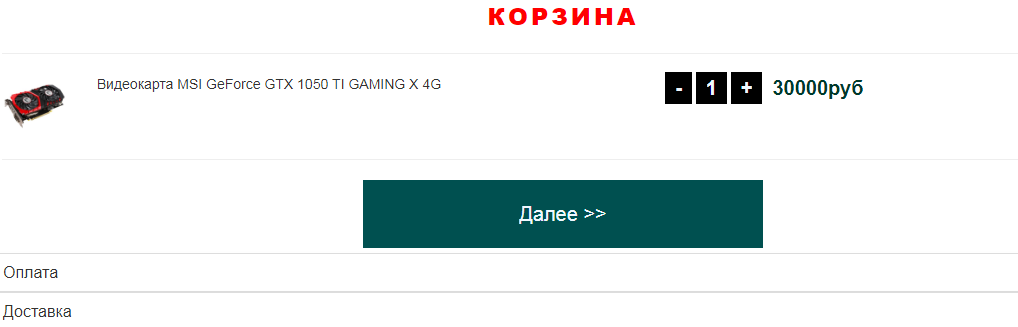


Рисунок 3.11 – Корзина

На рисунках 3.12 и 3.13 изображены формы оформления заказа.

Рисунок 3.11 – Редактирования товара в корзине

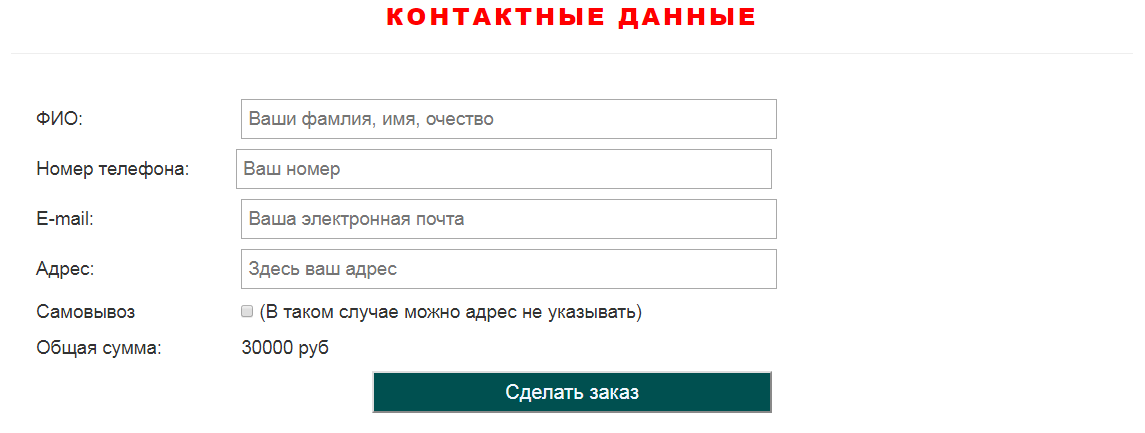


Рисунок 3.12 – Форма для указания контактных данных

Крометого что пользователи могут заказывать товары**,** они могут оставлять заявки на ремонт своей техники. Форма добавления заявки предоставлена на рисунке 3.14.

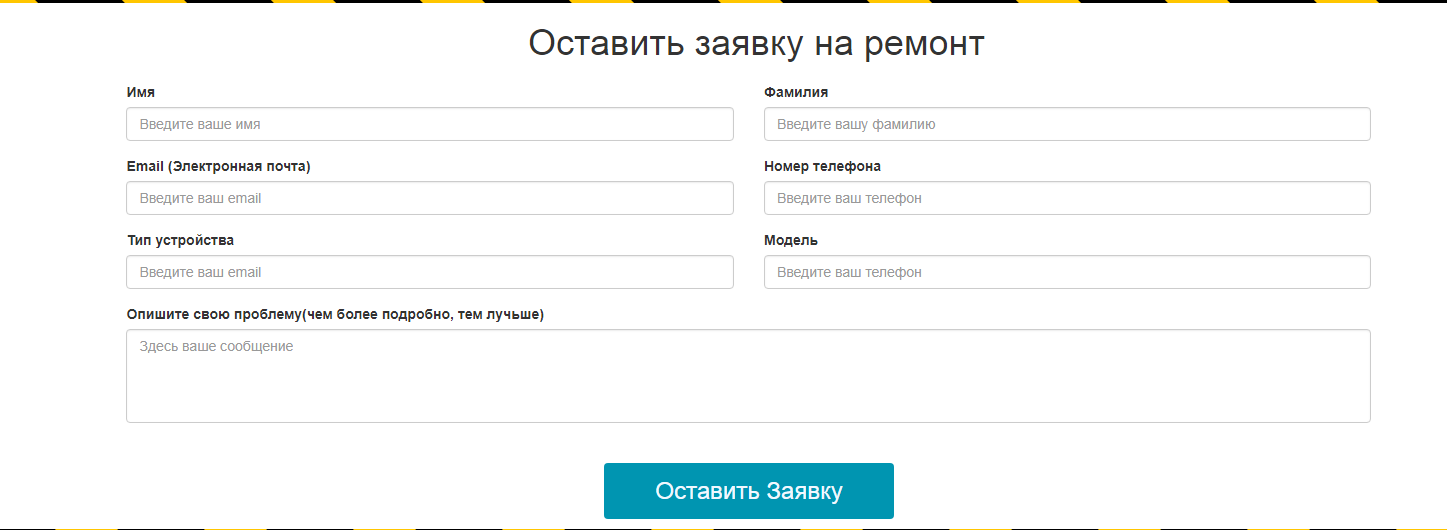


Рисунок 3.14 – Форма заявки на ремонт

Все заявки принимаются и обрабатываются администрацией сайта. На рисунке 3.15 изображена таблица заказов и заявок.

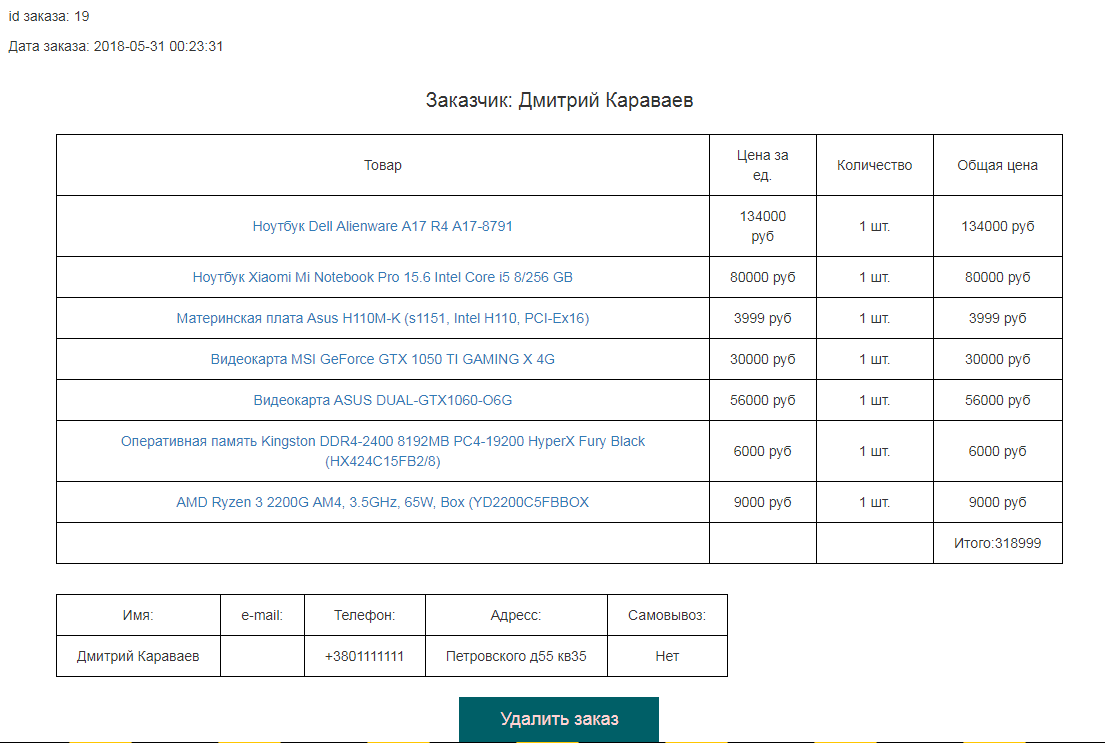


Рисунок 3.15 – таблица приема заказа товар

**4 Экономический анализ**

## 4.1 Исходные данные для расчета затрат

В дипломном  проекте разработана система «Компьютерный сервис и интернет магазин».

Общая стоимость владения (ОСВ) представляет собой совокупность затрат, осуществляемых на следующих стадиях жизненного цикла системы «Компьютерный сервис и интернет магазин»: разработка технологического проекта, его внедрение, эксплуатация и сопровождение.

Определим исходные данные для расчета затрат на осуществление  технического проекта системы. При проектировании используются трудовой потенциал сотрудников, а также персональный компьютер, оргтехника и специальное программное обеспечение для проектирования приложения.

Сведем  в  таблицу  1  трудоемкость  инженерно-технического персонала.

Таблица 1 – Трудоемкость работы исполнителей

|  |  |  |
| --- | --- | --- |
| Должность | Оклад  (руб./час). | Трудоемкость (Т),  дней (часов) |
| Программист/Тестировщик/Дизайнер | 100 | 20 |

В сферу деятельности руководителя проекта входит исследование предметной области, разработка технического задания, анализ экономической эффективности и исследование безопасных условий при работе проектируемого приложения. Программист занимается разработкой алгоритмов работы, разработкой пользовательского интерфейса, программированием и отладкой изготовленного приложения. Тестировщик выполняет полное тестирование приложения на наличие ошибок.

Для определения амортизационных отчислений сведем в таблицу 2 используемое оборудование и программное обеспечение (далее ПО).

Таблица 2 – Используемое оборудование и ПО для проектирования системы

|  |  |  |
| --- | --- | --- |
| Наименование оборудования | Кол-во единиц | Первоначальная стоимость,  руб. |
| Персональный компьютер | 1 | 10 000 |
| Ноутбук Dell vostro 3560 | 1 | 35 000 |

Ниже определим исходные данные для расчета затрат на внедрение системы. При внедрении систем учитываются трудозатраты сотрудников, осуществляющих работы (кодирование, монтаж систем, тестирование, отладку и т.д.), а также затраты на оборудование, комплектующие, материалы и специальное программное обеспечение, позволяющее физически реализовать и обеспечить качество системы. Они показаны в таблице 3 и таблице 4.

Таблица 3 – Затраты на оборудование и программное обеспечение, для внедрения системы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование затрат | Цена (руб./ед.) | Количество | Сумма (руб.) |
| Стоимость базового программного обеспечения | | | |
| ОС Windows | 0 | 1 | 0 |
| Breakets | Бесплатно | 1 | 0 |
| Браузер Chrome/Opera | Бесплатно | 1 | 0 |
| Стоимость базового оборудования | | | |
| Ноутбук Dell vostro 3560 | 35000 | 1 | 35000 |

Таблица 4 – Затраты на человеческие ресурсы, для написания и внедрение системы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работ | Время, часы | Оклад (руб./час) | Сумма (руб.) |
| Кодирование приложения | 70 | 50 | 3500 |
| Отладка приложения | 9 | 50 | 450 |
| Тестирование приложения | 20 | 50 | 1000 |
| Итого: |  |  | 4950 |

Определим нормы затрат (% берется от основной ставки: зар.платы, стоимости оборудования и т.д.):

- норма дополнительной заработной платы – 10% от основной зар.платы;

- норма премий – 20%;

- коэффициент отчислений в фонды социального страхования – 30% от ФЗП годового;

- норма транспортных расходов – 2-4% от первоначальной стоимости оборудования;

- норма амортизации – 15-20%;

- норма затрат на техническое обслуживание – 7-9% от первоначальной стоимости оборудования;

- норма затрат на расходные материалы – 2%;

- мощность, потребляемая вычислительной техникой:

а) компьютера – 0,42 кВт/ч. (или см. по техническому паспорту);

б) принтера и других устройств– 0,1 кВт/ч. (или см. по техническому паспорту);

- действительный фонд времени работы оборудования – 180 часов;

- норма накладных расходов – 60%;

- цена 1 кВт электроэнергии – 0,549 рублей;

- норма расходов на хранение и обновление программного обеспечения – 20%;

- норма расходов на содержание и ремонт производственных помещений – 1,5-2%.

## 4.2 Расчет затрат на проектирование системы

Затраты на проектирование и реализацию будут определяться выплатами заработной платы исполнителям работ с отчислениями на социальные нужды, а также стоимостью работы используемого оборудования (персонального компьютера, на котором производится проектирование системы и программирование ее управляющей программы).

Проектирование системы и реализация программы ведутся с использованием ПК, поэтому затраты можно определить по следующему выражению:

З проект= ЗПосн + ЗПдоп + Осн + tоб × Смч (1)

где ЗПосн – основная заработная плата исполнителей работ, рублей;

ЗПдоп – дополнительная заработная плата исполнителей работ, рублей;

Осн – отчисления на социальные нужды, рублей;

tоб – время работы используемого для проектирования компьютерного оборудования, часов;

Смч – стоимость машино-часа работы используемого оборудования, рублей/час.

Основная заработная плата (ЗПосн) исполнителей работ определяется выражением:

, (2)

где *Оi* – оклад i-го разработчика, рублей. (из таблицы1);

*Тднi* – трудоемкость дней i-го разработчика ( из таблицы 1);

*П* = 20% – норма премии.

Определим дополнительную заработную плату (ЗПдоп) исполнителей работ:

 **,** (3)

где Ндоп – норма дополнительной заработной платы (обычно 10-15%).

Тогда заработная плата исполнителей за весь срок составит:

ЗП=ЗПосн+ЗПдоп, (4)

ЗП=2400+240=2640

Определим отчисления в фонды (Осн) по формуле 5:

 , (5)

где Нсн = 30% – норма отчислений в фонды.

Стоимость машино-часа работы оборудования (Смч) определяется выражением:



где ЗП – затраты на заработную плату обслуживающего персонала, рублей;

Аm – годовые амортизационные отчисления, рублей;

Зто – затраты на техническое обслуживание за год, рублей;

Зм – затраты на расходные материалы, рублей;

Зэ – затраты на электроэнергию за год, рублей;

Знр – годовые накладные расходы, рублей;

Фд – действительный фонд времени работы оборудования, часы.

Определим годовые амортизационные отчисления для используемого оборудования:



где На = 15-20% - норма амортизации;

Сп – первоначальная стоимость используемого оборудования, рублей. (из таблицы 2).

Определим годовые затраты на техническое обслуживание используемого оборудования по формуле:



где Нто= 7-9% - норма затрат на техническое обслуживание.

Определим годовые затраты на расходные материалы:



где Нм = 2% - норма затрат на расходные материалы;

Определим годовые затраты на электроэнергию используемого оборудования:



где N – мощность оборудования;

Цэ – цена электроэнергии, рублей./кВт.;

Кс – коэффициент, учитывающий потери в сети Кс = 0,98;

Определим годовые накладные расходы используемого оборудования по формуле 11:

где Ннр – норма накладных расходов.

По найденным значениям вычислим стоимость машино-часа работы используемого оборудования:

Определим время работы используемого оборудования (tоб). Руководитель проекта выполняет проектирование с использованием ПК 1 в течение 20 часов, Тогда время работы оборудования составит: 20 часов.

Исходя из рассчитанных значений вычислим суммарные затраты на проектирование системы по формуле 1.

Зпроект= 2400 + 240 +792 + 20 × 100,18=5435,6 руб.

## 4.3 Расчет затрат на внедрение

В расчет затрат на внедрение должные входить затраты на изготовление системы с учетом стоимости ее аппаратного обеспечения, кодирование, отладку и тестирование системы.

Затраты на внедрение будут определяться выражением:

 Звнедр=Зизг+Зтр+Змонт, (12)

где Зизг – затраты на изготовление системы , рублей;

Зтр – затраты на транспортировку, рублей;

Зм – затраты на монтаж и установку (включая демонтаж), или инсталляцию системы, в рублях.

Затраты на изготовление (Зизг) определяются как сумма из таблицы .

Затраты на транспортировку (Зтр) определяются выражением:



где Нтр – норма транспортных расходов, рублей.

Подставляя известные значения в выражение, найдем затраты на внедрение системы по формуле 12:

Звнедр=4950+99=5049

## 4.4 Расчет эксплуатационных расходов

Годовые текущие затраты при эксплуатации системы будут состоять только из затрат на хранение и обновление программного обеспечения, так как другие эксплуатационные расходы остаются прежними: потребление электроэнергии эквивалентной по мощности, расходы на обслуживание, расходы на содержание и ремонт производственных помещений. Поэтому эксплуатационные расходы будут определяться выражением 15:



где Нрох = 20% – годовая норма расходов на хранение и обновление программного обеспечения (перепрограммирование).

## 4.5 Расчет и построение диаграммы общей стоимости системы

Суммарные затраты будут составлять общую стоимость спроектированной системы без учета рентабельности, потому как система проектируется для собственных нужд при использовании взамен базовых устройств. Полную стоимость можно определить выражением:

С=Зпроет+Звнедр+Зэкс , (16)

С=5435 + 5049 + 1009,8 =11 513,8

Покажем на диаграмме статьи затрат, составляющие общую стоимость системы (рисунок 4.1).



Рисунок 4.1 – Диаграмма общей стоимости системы

## 4.6 Расчет прибыли разработчика системы

Размер прибыли определяется по формуле 17.

 PR=C×R, (17)

где R – уровень рентабельности проекта (выбираем сами, желательно нормальный 15-30%).

PR=11513,8 15% = 1727,07



Цена реализации проекта определяется по формуле 18.



где НДС – налог на добавленную стоимость (18%).

Цр=PR+НДС=2038,57

## 4.7 Расчет затрат, связанных с покупкой, внедрением и использованием системы и срока окупаемости

Предприятие, покупающее новую систему, несет затраты не только по покупке системы, но и по организации новых рабочих мест. Оно должно иметь экономический эффект от ее внедрения в виде экономии денег за вычетом всех расходов, повышения качества своей работы и за счет этого расширения рынка сбыта своей продукции или увеличения объемов продаж и т.д. Рентабельность проекта определяет, через какой срок предприятие окупит затраты и начнет получать выгоду от приобретения системы [3].

Рассчитаем эти показатели.

Капитальные затраты покупателя на приобретение и внедрение системы определяются по формуле 19.

К=Цр+Ккрм+Ктех+Кпр, (19)

где Ккрм – капитальные вложения на создание новых рабочих мест персонала, обслуживающих новую систему.

Ктех – капитальные вложения на техническое оснащение рабочего места пользователя (посчитать сумму затрат на рабочее место нового специалиста.

Кпр – прочие капитальные вложения, связанные с внедрением (5% от Цр).

Таким образом затраты покупателя на покупку, установку и внедрение составят:

Ккрм=0×12=0 руб.

Кпр=Цр×5%, (20)

Кпр=2038,57×5%=101,92 руб.

Капитальные затраты составят 2140,49 рублей.

Срок окупаемости затрат на приобретение и внедрение проекта рассчитывается по формуле 21.



где В – годовая выгода от внедрения приложения, рублей.

## 4.8 Выводы

Выше были проведены экономические расчеты, связанные с проектированием, построением и продажей со стороны организации-разработчика, и приобретением и эксплуатацией со стороны организации-покупателя сайта «Компьютерный сервис и интернет магазин». Разработчиком создано приложение, соответствующая требованиям заказчика [2].

В результате расчетов были получены следующие экономические показатели:

* затраты на создание системы равны 5435 рублей;
* цена реализации проекта составляет 5049 рублей;
* разработчиком получена и распределена прибыль от реализации проекта, составляющая рублей;
* заказчик понесет капитальные затраты на покупку и внедрение системы, равные 11 513,8 рублей;
* заказчиком достигается выгода от приобретения системы, равная 0,36 рублей;
* срок окупаемости проекта 1 год.

Внедрение системы «Компьютерный сервис и интернет магазин» принесет экономический эффект заказчику в виде экономического дохода.

Таким образом, разработанная система полностью удовлетворяет требованиям потребителя.

# 5 Безопасность и экологичность проекта

## 5.1 Обоснование важности охраны труда

Охрана труда - это важнейшая социально-экономическая проблема, требующая к себе постоянного внимания со стороны государства, нанимателей и объединений работников.

Охрана труда – система сохранения жизни и здоровья работников в процессе трудовой деятельности, включающая в себя правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия (Трудовой кодекс Луганской Народной Республики, статья 39) [14]. Общество несёт огромные экономические потери, обусловленные недооценкой потенциальной возможности охраны труда в повышении эффективности производства.

Охрана труда - это система законодательных актов, социально-экономических, организационных, технических, гигиенических и лечебно-профилактических мероприятий, и средств, обеспечивающих безопасность, сохранение здоровья и работоспособности человека в процессе труда. Полностью безопасных и безвредных производственных процессов не существует.

Задачи охраны труда - свести к минимуму вероятность поражения или заболевания работающего с одновременным обеспечением комфорта при максимальной производительности труда.

Охрана труда играет важную роль в трудовой жизни человека. Правильная организация труда значительно повышает его производительность и резко снижает возможность производственных травм, увечий. Это, в свою очередь, оказывает и непосредственное положительное влияние на экономическую сторону труда: происходит снижение на оплату больничных листов и лечения сотрудников, уменьшается количество и размер компенсаций за работу во вредных условиях и пр. По статистическим подсчетам, затраты на необходимые мероприятия и средства для охраны труда и безопасности жизнедеятельности обходятся в десять раз меньше, чем расходы из-за несчастных случаев, связанных с их своевременным не проведением.

## 5.2 Характеристика условий труда программиста

В настоящее время компьютерная техника широко применяется во всех областях деятельности человека. При работе с компьютером человек подвергается воздействию ряда опасных и вредных производственных факторов: электромагнитных полей (диапазон радиочастот: ВЧ, УВЧ и СВЧ), инфракрасного и ионизирующего излучений, шума и вибрации, статического электричества и др.

В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

## 5.3 Анализ опасных и вредных производственных факторов при работе с компьютером

При работе с компьютером человек может попасть под влияние таких вредных факторов:

* повышенная или пониженная влажность;
* повышенное значение напряжения в электрической цепи;
* повышенный уровень электромагнитных излучений;
* отсутствие или недостаток естественного света;
* недостаточная освещенность рабочей зоны;
* повышенная яркость света;
* пониженная контрастность;
* повышенная пульсация светового потока;
* расположение рабочего места на значительной высоте;
* повышенная или пониженная подвижность воздуха;
* повышенный уровень ионизирующих излучений в рабочей зоне;
* повышенная запыленность и загазованность воздуха рабочей зоны;
* повышенное или пониженное барометрическое давление в рабочей зоне;
* повышенный уровень статического электричества;
* повышенная напряженность электрического поля;
* повышенная напряженность магнитного поля;
* прямая и отраженная блесткость;
* повышенный уровень ультрафиолетовой радиации;
* повышенный уровень инфракрасной радиации.

Химические опасные и вредные производственные факторы подразделяются:

1. По характеру воздействия на организм человека:

* токсические;
* раздражающие;
* сенсибилизирующие;
* канцерогенные;
* мутагенные;
* влияющие на репродуктивную функцию.

2. По пути проникновения в организм человека через:

* органы дыхания;
* желудочно-кишечный тракт;
* кожные покровы и слизистой оболочки.

Биологические опасные и вредные производственные факторы включают следующие биологические объекты:

* патогенные микроорганизмы (бактерии, вирусы, грибы, простейшие) и продукты их жизнедеятельности;
* опасные свойства микро и макро организмов.

Психофизические опасные и вредные производственные факторы по характеру действия подразделяются:

* физические перегрузки (статические и динамические);
* нервно-психические перегрузки (умственное напряжение и перенапряжение, монотонность труда, эмоциональные перегрузки, утомление, эмоциональный стресс, эмоциональная перегрузка).

В данном дипломном проекте среди приведенных выше четырех групп опасных и вредных производственных факторов можно пренебречь биологическими и химическими факторами, так как на данном рабочем месте они оказывают незначительное влияние на деятельность техника программиста. Рассмотрим только физические и психофизические опасные и вредные производственные факторы и мероприятия по их устранению или снижению.

## 5.4 Общие требования при работе за компьютером

Инструкция по охране труда при работе с персональными компьютерами (далее - Инструкция) устанавливает общие требования безопасности для работников, использующих в работе персональные компьютеры (далее - ПК).

К работе с ПК допускаются работники, не имеющие медицинских противопоказаний, прошедшие инструктаж по вопросам охраны труда, с группой по электробезопасности не ниже 1.

Женщины со времени установления беременности и в период кормления грудью к выполнению всех видов работ, связанных с использованием ПК, не допускаются.

При работе с ПК на работников могут оказывать неблагоприятное воздействие следующие опасные и вредные производственные факторы:

* повышенный уровень электромагнитных излучений;
* повышенный уровень ионизирующих излучений;
* повышенный уровень статического электричества;
* повышенная напряженность электростатического поля;
* повышенная или пониженная ионизация воздуха;
* повышенная яркость света;
* прямая и отраженная блесткость;
* повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
* статические перегрузки костно-мышечного аппарата и динамические локальные перегрузки мышц кистей рук;
* перенапряжение зрительного анализатора;
* умственное перенапряжение;
* эмоциональные перегрузки;
* монотонность труда.

В зависимости от условий труда, в которых применяются ПК, и характера работы на работников могут воздействовать также другие опасные и вредные производственные факторы.

Организация рабочего места с ПК должна учитывать требования безопасности, удобство положения, движений и действий работника.

Рабочий стол с учетом характера выполняемой работы должен иметь достаточный размер для рационального размещения монитора (дисплея), клавиатуры, другого используемого оборудования и документов, поверхность, обладающую низкой отражающей способностью.

Клавиатура располагается на поверхности стола таким образом, чтобы пространство перед клавиатурой было достаточным для опоры рук работника (на расстоянии не менее чем 300 мм от края, обращенного к работнику).

Чтобы обеспечивалось удобство зрительного наблюдения, быстрое и точное считывание информации, плоскость экрана монитора располагается ниже уровня глаз работника предпочтительно перпендикулярно к нормальной линии взгляда работника (нормальная линия взгляда минус 15° вниз от горизонтали).

Для исключения воздействия повышенных уровней электромагнитных излучений расстояние между экраном монитора и работником должно составлять не менее 500 мм (оптимальное 600-700 мм).

Применяемые подвижные подставки для документов (пюпитры) размещаются в одной плоскости и на одной высоте с экраном.

Рабочий стул (кресло) должен быть устойчивым, место сидения должно регулироваться по высоте, а спинка сиденья - по высоте, углам наклона, а также расстоянию спинки от переднего края сиденья. Регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Для тех, кому это удобно, предусматривается подставка для ног.

Рабочее место размещается таким образом, чтобы естественный свет падал сбоку (желательно слева).

Для снижения яркости в поле зрения при естественном освещении применяются регулируемые жалюзи, плотные шторы.

Светильники общего и местного освещения должны создавать нормальные условия освещенности и соответствующий контраст между экраном и окружающей обстановкой с учетом вида работы и требований видимости со стороны работника. Освещенность на поверхности стола в зоне размещения рабочего документа должна составлять 300-500 люкс.

Возможные мешающие отражения и отблески на экране монитора и другом оборудовании устраняются путем соответствующего размещения экрана, оборудования, расположения светильников местного освещения.

При рядном размещении рабочих столов расположение экранов видеомониторов навстречу друг другу из-за их взаимного отражения не допускается.

Для обеспечения безопасности работников на соседних рабочих местах расстояние между рабочими столами с мониторами (в направление тыла поверхности одного монитора и экрана другого монитора) должно быть не менее 2,0 м, а расстояние между боковыми поверхностями мониторов - не менее 1,2 м.

Для снижения уровня напряженности электростатического поля при необходимости применяются экранные защитные фильтры.

Для обеспечения оптимальных параметров микроклимата проводятся регулярное в течение дня проветривание и ежедневная влажная уборка помещений, используются увлажнители воздуха.

При работе с ПК обеспечивается доступ работников к первичным средствам пожаротушения, аптечкам первой медицинской помощи.

При работе с ПК работники обязаны:

* соблюдать режим труда и отдыха, установленный законодательством, правилами внутреннего трудового распорядка организации, трудовую дисциплину, выполнять требования охраны труда, правил личной гигиены;
* выполнять требования пожарной безопасности, знать порядок действий при пожаре, уметь применять первичные средства пожаротушения;
* курить только в специально предназначенных для курения местах;
* знать приемы оказания первой помощи при несчастных случаях на производстве;
* о неисправности оборудования и других замечаниях по работе с ПК сообщать непосредственному руководителю или лицам, осуществляющим техническое обслуживание оборудования.

Не допускается:

* выполнять работу, находясь в состоянии алкогольного опьянения либо в состоянии, вызванном употреблением наркотических средств, психотропных или токсических веществ, а также распивать спиртные напитки, употреблять наркотические средства, психотропные или токсические вещества на рабочем месте или в рабочее время;
* устанавливать системный блок в закрытых объемах мебели, непосредственно на полу;
* использовать для подключения ПК розетки, удлинители, не оснащенные заземляющим контактом (шиной).

**6 Тестирование**

## 6.1 Цели и основные методы тестирования web-сайта

Тестирование является одним из важнейших этапов разработки программного обеспечения. На данном этапе производится отладка программного продукта и поиск ошибок, для дальнейшего их исправления. Тестирование сайта – это последний и обязательный этап технической разработки сайта. Он играет важнейшую роль в процессе создания ресурса, так как именно от качества тестирования зависит дальнейшая жизнь ресурса [13].

Тестирование сайта на ошибки выполняется разными методами в разных браузерах. Чтобы добиться безупречного функционирования сайта, необходимо создают план действий, в который входит:

* тестирование функциональности;
* проверка удобства пользования;
* тест производительности;
* проверка на безопасность;
* UI-тестирование (тестирование интерфейса).

**6.2 Функциональное тестирование сайта и тестирование пользовательского интерфейса (UI-интерфеса)**

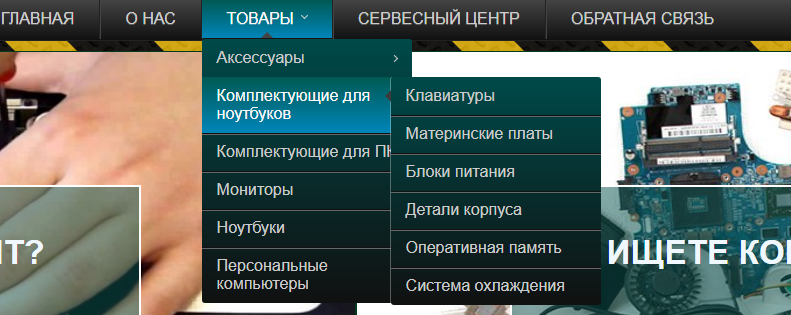
Это важнейший этап тестирования, который нельзя заменить или пропустить. Ошибка в функциональности неизбежно приведет к потере клиентов. Например, если в функции корзины интернет-магазина допущена ошибка, то посетитель не сможет купить товар. При функциональном тестировании проверяются:

* навигация;
* поиск и оформление заказа, покупки и оплаты товара;
* форма регистрации или входа в личный кабинет;
* возможности редактирования сайта (добавление позиций, редактирование контента и т.д.).

На рисунках 6.1 и 6.2 предоставлены возможности навигационной панели по сайту.



Рисунок 6.1 – Основной вид меню

 Рисунок 6.2 – Вид меню, при наведение на каталог товаров

На рисунке 6.3 изображена поисковая строка, а на рисунке 6.4 результат поиска. Результатом поиска является запрос, прописанный в поисковой строке.



Рисунок 6.3 – Поисковая строка

На рисунке 6.4 изображен результат поиска по запросу «Видекарта»:

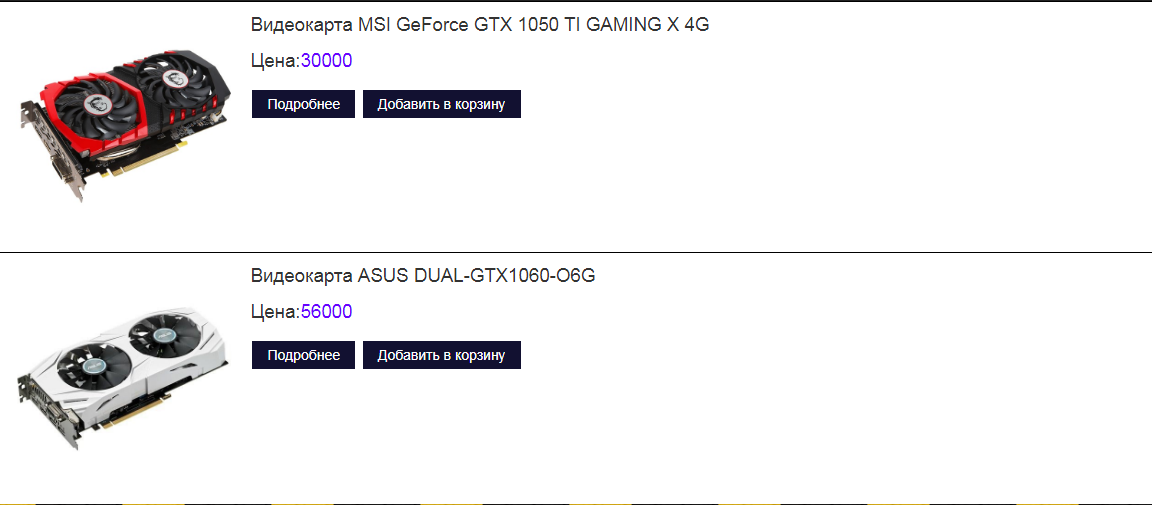


Рисунок 6.4 – Результат поиска

После того как пользователь нашел необходимый ему товар, он может добавить его в корзину. Корзина предоставлена на рисунке 6.5.

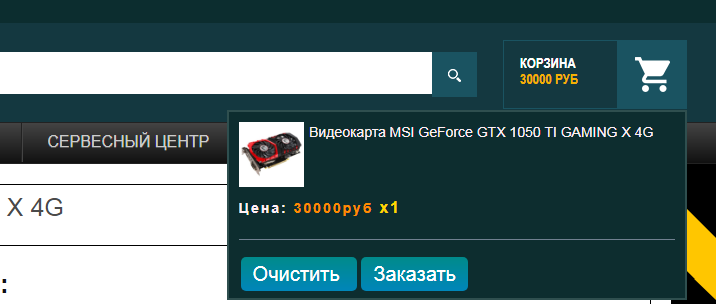
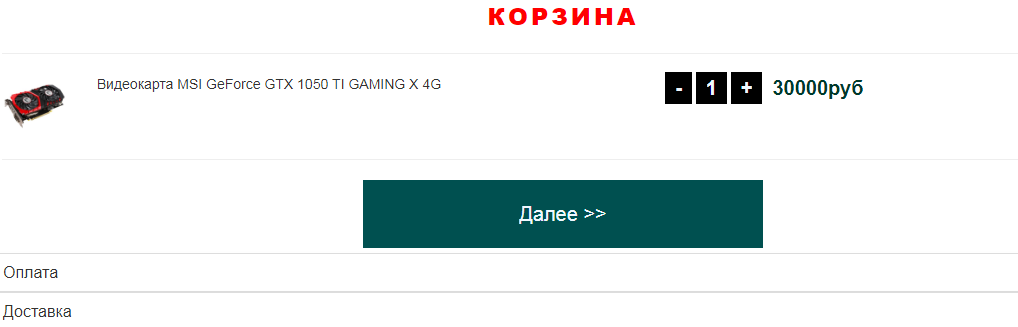


Рисунок 6.5 – Корзина

На рисунках 6.6 и 6.7 изображены формы оформления заказа. Заказ товара имеет 2 фазы. Первая, это редактирование выбранного товара, а вторая, это внесение в форму контактных данных, типа адреса, номера телефона и т.п.

Рисунок 6.6 – Редактирования товара в корзине

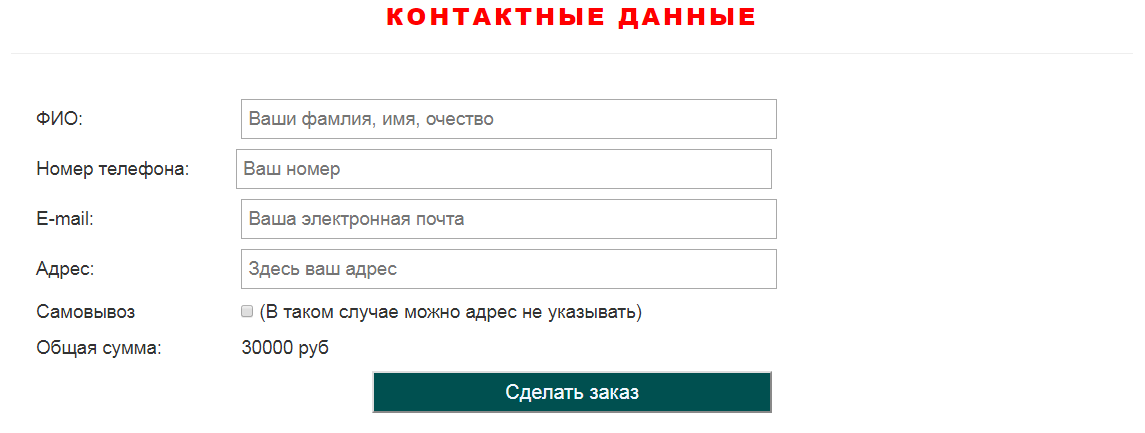


Рисунок 6.7 – Форма для указания контактных данных

Пользователи могут оставлять заявки на ремонт своей техники. Скриншот формы добавления заявки предоставлен на рисунке 6.8.

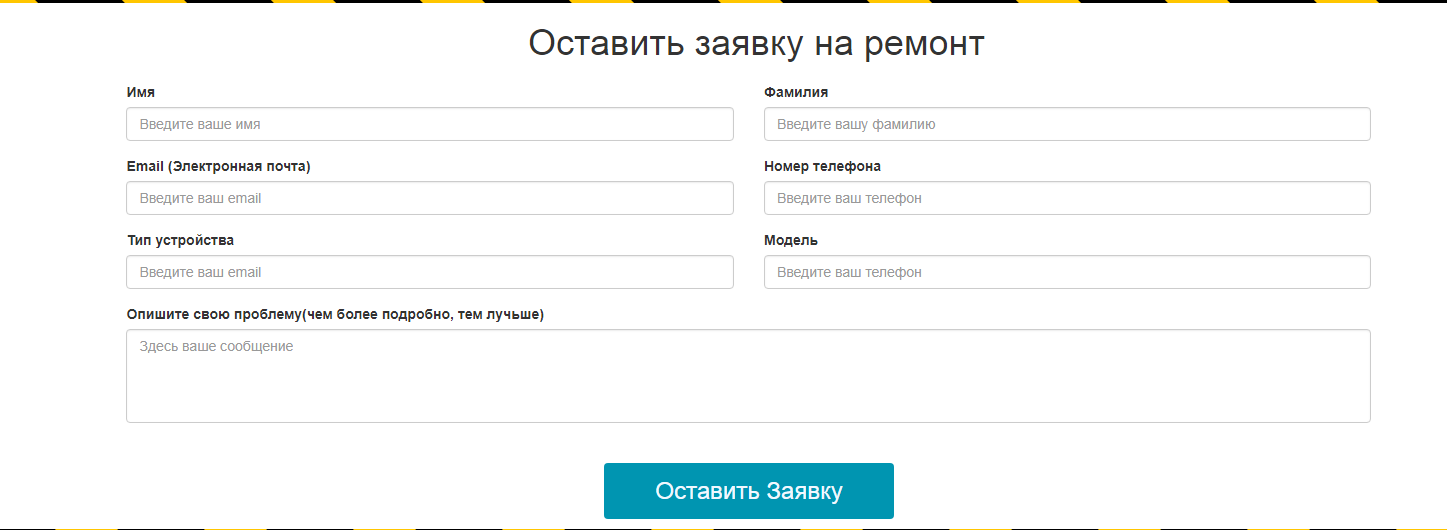


Рисунок 6.8 – Форма заявки на ремонт

Все заявки принимаются и обрабатываются администрацией сайта. На рисунке 6.9 изображена таблица заказанных товаров, конкретным пользователем.

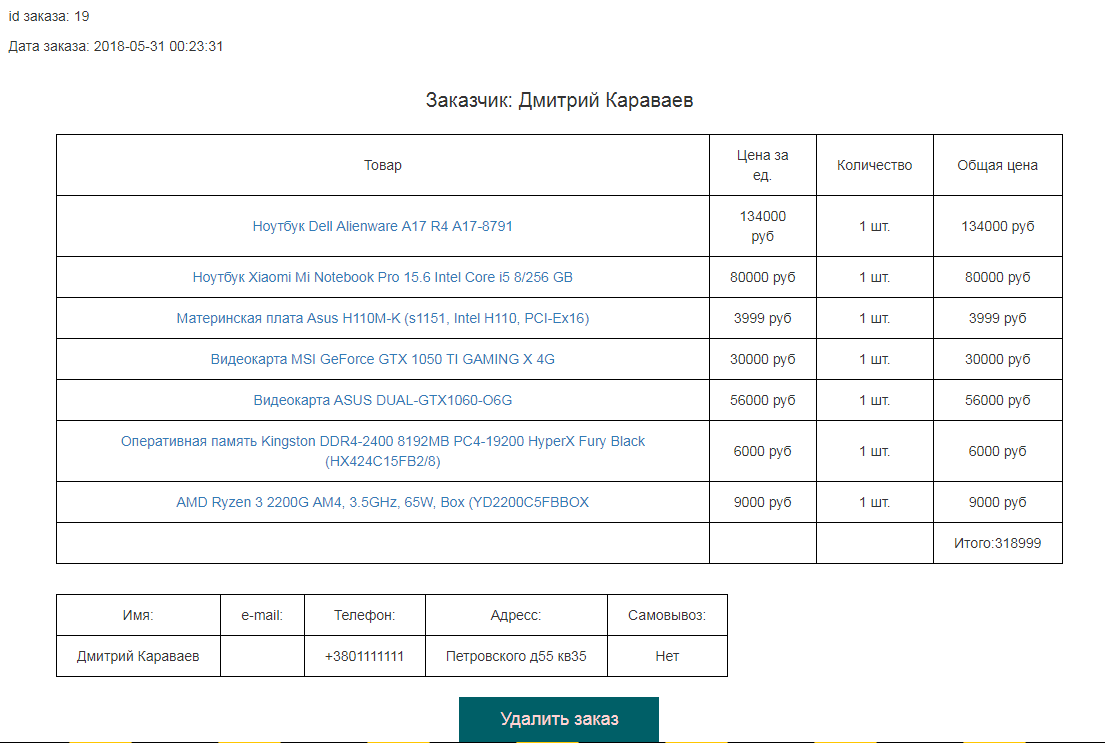


Рисунок 6.9 – таблица приема заказа товар

На рисунке 6.10 предоставлена форма с заявкой на ремонт.

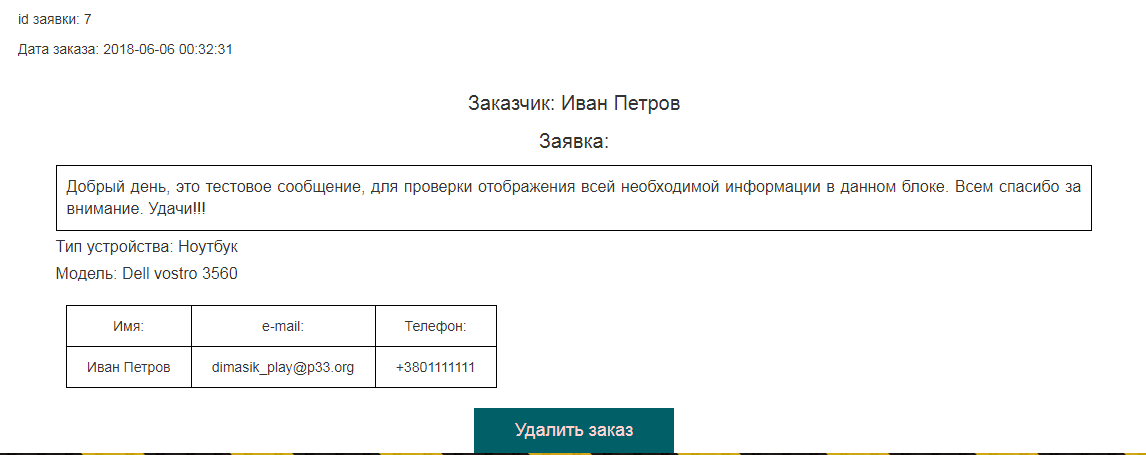


Рисунок 6.10 – таблица приема заказа товар

Регистрация\авторизация на сайте имеет упрощенный и одновременно современный вид, для удобства пользованием сайтом. Скриншоты регистрации\авторизации изображены на рисунках 6.11 и 6.12.

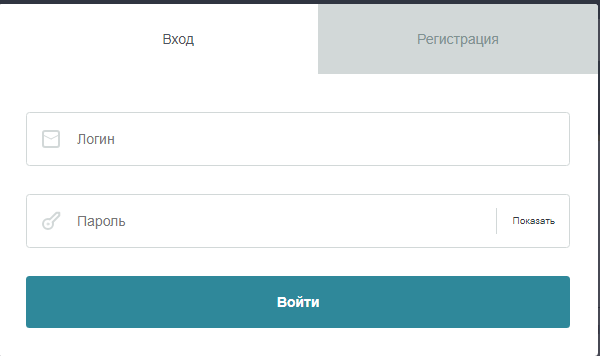


Рисунок 6.11 – авторизация

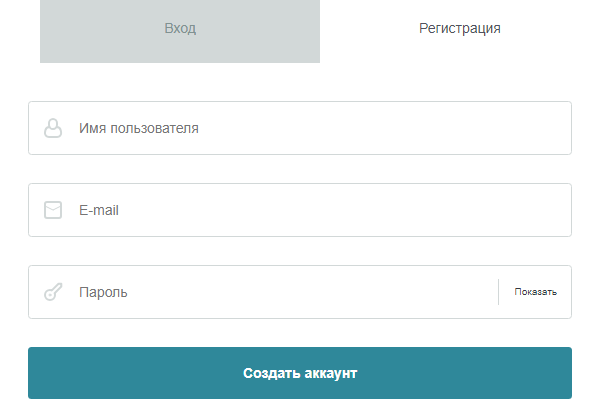


Рисунок 6.12 – регистрация

Редактирование сайта осуществляется администратором сайта, под редактированием подразумеваться добавление и удаления товара, и пунктов меню во вкладки «Товары». На рисунке 6.13 изображена форма добавления товара, а на рисунке 6.14 изображения форма работы с элементами каталога, вкладкой «Товары», которая изображена на рисунке 6.2.

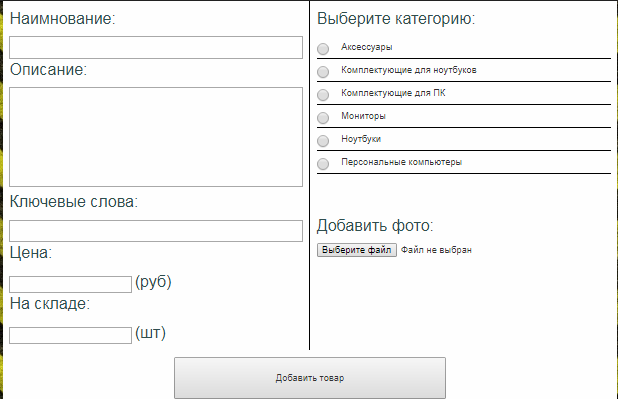


Рисунок 6.13 – форма добавления товара

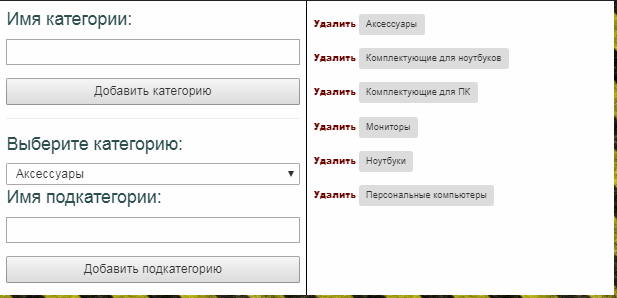


Рисунок 6.14 – форма работы с каталогом

**6.3 Тестирование безопасности сайта**

Тестирование безопасности - это стратегия тестирования, используемая для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.

Основной этап защиты реализован на уровне базы данных, его суть заключается в том, что прежде чем записать данные в таблицу они экранируются и преобразовываются в строковый тип данных. Это сделано с целью защиты сайта от SQL – инъекций и от вредоносных скрипов, которые в последствии могут навредить сайту, а так же функция заменяет символы на переданные параметры массивом. Функция продемонстрирована на листинге 6.1:

Листинг 6.1 – Экранирование данных

private function getQuery($query, $params) {

if ($params) {

for ($i = 0; $i < count($params); $i++) {

$pos = strpos($query, $this->sym\_query);

$arg = "'".$this->mysqli->real\_escape\_string($params[$i])."'";

$query = substr\_replace($query, $arg, $pos, strlen($this->sym\_query));

}

}

return $query;

}

Обработка и вывод ошибок, немаловажный аспект любого сайта. Пользователь должен понимать, почему он не может совершить то, или иное действие.

При авторизации пользователь может увидеть сообщение «Неверный логин, или пароль», если данные введены не верно. На рисунке 6.15 изображена ошибка авторизации пользователя [9].

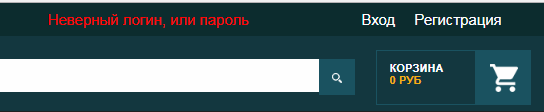


Рисунок 6.15 – ошибка неверный логин, или пароль

Так же при введение неверного запроса в поисковое поле, генерируется ошибка «Поиск не дал результатов». На рисунке 6.16 изображена ошибка при поиске несуществующего товара.

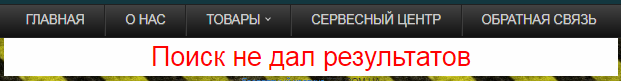


Рисунок 6.16 – ошибка при поиске товара

**ЗАКЛЮЧЕНИЕ**

В процессе проектирования дипломного проекта был разработан сайт «Сервисный центр и интернет-магазин по продажам компьютерной техники», соответствующий всем требованиям [1].

Цель достигнута, поставленные задачи выполнены:

* исследована предметная область;
* проанализирована web - сайты аналогичной тематики;
* выбрана стратегию разработки;
* спроектирована модель сайта;
* разработана web - сайт с использованием php скриптинга;
* рассчитана финансовые затраты на создание проекта.

В процессе работы исследована предметная область. Проведен анализ web - сайтов аналогичной тематики для выявления недостатков системы, которые учитывались при создании сайта. Оптимизирован web-интерфейс и навигация сайта, для того что бы пользователю было удобнее ориентироваться в виртуальном пространстве.

Интернет-магазин имеет удобный и понятный интерфейс, благодаря которому не требуется излишних усилий в освоении работы с сайтом. Вся информация не содержит ничего лишнего, кнопки и элементы меню интуитивно понятны, для чего они используются.

В ходе тестирования сайта результаты проверки показали, что система работает без ошибок, имеется возможность работы с веб-сайтом во всех популярных браузерах. Но как и любой продукт может иметь некие недоработки, которые будут устраняться по мере выявления, во время эксплуатации [4].

Код дипломного проекта можно почитать в приложении А.

Сайт был размещен на хостинге по адресу http://restart-atrasite.zzz.com.ua.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. ГОСТ 19.503-79 ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению.
2. Борисов Е. Ф. Основы экономики: Учебное пособие / Е.Ф. Борисов. - М. : Юрайт - Издат, 2009. - 316с
3. Экономика: Учебное пособие / Под ред. А.С. Булатова. - М. : Юристъ, 2009. - 896 с.
4. Энж Эрик , Спенсер Стефан , Фишкин Рэнд , Стрикчиола Джесси SEO. Искусство раскрутки сайтов; БХВ-Петербург - Москва, 2014. - 668 c.
5. Коггзолл Д. PHP 5: полное руководство Д. Коггзолл, М. : - 2009. - 752 с.
6. Котеров Д.Н., Костарев, А.Л. PHP 5 - наиболее полное руководство // Д.Н. Котеров, А.Л. Костарев - Санкт-Петербург 2005. - 1120 с.
7. Дубаков, М.А. Веб-мастеринг средствами CSS / М.А. Дубаков. - Санкт-Петербург : БХВ-Петербург, 2002. - 544 c.
8. Нильсен, Я. Веб-дизайн. Перевод с анл. / Я. Нильсен - Санкт-Петербург: Символ-Плюс, 2007. - 512 c.
9. Фрэйн, Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрэйн. - Санкт-Петербург : Питер, 2014. - 56 с.
10. PHP: Hipertext Processor // Documentation

URL: http://php.net/manual/ru/function.mysql-real-escape-string.php (дата обращения: 08.05.2018).

1. PHP: Hipertext Processor // Manual

URL: http://php.net/manual/ru/mysqli.quickstart.statements.php (дата обращения: 08.05.2018).

1. freshdesign // Правильное тестирование веб-сайта, или как обеспечить себе спокойный сон

URL:https://freshdesign.ua/blog/technology/website-testing (дата обращения: 05.06.2018).

1. Министерство труда и социальной политики/Законодательство // Трудовой кодекс Луганской Народной Республики [с изменениями внесенными Законом Луганской Народной Республики от 04.03.2016 № 88-II от 12.08.2016 № 113-II]

URL: https://mintrudlnr.su/docs/241-trudovoy-kodeks-luganskoy-narodnoy-respubliki-s-izmeneniyami-vnesennymi-zakonom-luganskoy-narodnoy-respubliki-ot-04032016-88-ii-ot-12082016-113-ii.html (дата обращения: 03.06.2018).

**ПРИЛОЖЕНИЕ А**

**Исходный код приложения**

Листинг А.1- основные страницы сайта

<?php require\_once 'php/start.php'; ?>

<!DOCTYPE html>

<html lang="en">

<head>

<?php

$title = 'Restart';

require\_once('blocks/head.php');

?>

<link rel="stylesheet" href="style/harware.css">

</head>

<body><?php

require\_once('blocks/header.php'); ?>

<?php

require\_once('blocks/nav\_bar.php');

?>

<div class=" col-md-12" style="">

<?php

require\_once('blocks/slid.php');

?>

</div>

<div class="clear"></div>

<div class="" style = "height:300px; ">

</div>

<?php require\_once 'blocks/hardware.php'; ?>

<?php

require\_once('blocks/js\_sript.php');?>

</body>

</html>

<------------------------------------------------------------------->

<? session\_start(); ?>

<!DOCTYPE html>

<html lang="en">

<head>

<?php

require\_once '/blocks/head.php';

require\_once '/work\_to\_db/func\_db.php';

?>

<link rel="stylesheet" href="style/add\_goods.css">

<head>

<body>

<? require\_once '/blocks/header.php';

require\_once '/blocks/nav\_bar.php';?>

<div class="wrapp">

<div class = "form">

<form enctype="multipart/form-data" action="php/add\_goods.php" method="POST">

<div class = "sub\_form\_block">

<p>Наимнование:</p>

<input class = "sub\_key" name = "subject" type = "text">

<br>

<p>Описание:</p>

<textarea name = "description" type = "text"></textarea><br>

<p>Ключевые слова:</p>

<input class = "sub\_key" name = "keywords" type = "text">

<p>Цена:</p>

<input name = "price" type="text">

<span>(руб)</span>

<p>На складе:</p>

<input name = "stock" type="text">

<span>(шт)</span>

</div>

<div class = "sub\_form\_block">

<p>Выберите категорию:</p>

<?

$Q="SELECT \* FROM main\_categorys ORDER BY categorys";

$result = $db->select($Q);

for($i=0;$i<count($result);$i++){

echo'<div class = "dropdown">';

echo'<input type="radio" class = "radi" value = "'.$result[$i]['categorys'].'" name = "main\_category" class = "change\_category" >

<button class="btn dropdown-toggle" type="button" data-toggle="dropdown" >'.$result[$i]['categorys'].'</button>

<ul class="dropdown-menu">';

$id = [$main\_res[$i]['id']];

$sub\_res = $db->select($Q = "SELECT \* FROM categorys WHERE main\_category ={?}",$id);

for($j=0;$j<count($sub\_res);$j++){

echo'<li><input type="radio" class = "radi" value ="'.$sub\_res[$j]['category'].'" name = "sub\_category" class = "sub\_category" ><a href="#">'.$sub\_res[$j]['category'].'</a></li>';

}

echo'</ul>

</div>';

}

?> <br>

<hr>

<p>Добавить фото:</p>

<input name = "img" type = "file">

</div>

<div class="clear"></div>

<input type = "submit" value = "Добавить товар">

</form>

</div>

</div>

</body>

</html>

<------------------------------------------------------------------->

<?php

require\_once 'php/start.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<?php require\_once('blocks/head.php'); ?>

<link rel="stylesheet" type="text/css" href="style/f\_back.css">

</head>

<body>

<?php

//nav\_bar

require\_once('blocks/nav\_bar.php');

?>

<form action="" class="contact">

<fieldset class="contact-inner">

<p class="contact-input">

<input type="text" name="name" placeholder="Ваше имя…" autofocus>

</p>

<p class="contact-input">

<input type="e-mail" value="<? echo $\_SESSION['mail'] ?>" name="mail" placeholder="Ваш e-mail…" autofocus>

</p>

<p class="contact-input">

<input type="text" value="" name="theme\_msg" placeholder="Тема сообщения…" autofocus>

</p>

<p class="contact-input">

<textarea name="message" placeholder="Ваше сообщение…"></textarea>

</p>

<p class="contact-submit">

<input type="submit" value="Отправить сообщение">

</p>

<p>\*ответ на ваше сообщение придет на указанный вами e-mail </p>

</fieldset>

</form>

<footer class="container-fluid row">

</footer>

<?php require\_once('blocks/js\_sript.php');?>

</body>

</html>

<------------------------------------------------------------------->

<? session\_start();

require\_once '/blocks/head.php';

require\_once '/work\_to\_db/func\_db.php';

$db = DataBase::getDB();

?>

<html>

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="style/form\_reg\_edit.css">

<head>

<body>

<?php require\_once '/blocks/header.php';

require\_once '/blocks/nav\_bar.php';

?>

<div class="wrapp">

<div class = "form">

<form action = "php/add\_category.php" method="post">

<p>Имя категории:</p>

<input name = "category" type = "text">

<br>

<br>

<input type = "submit" value = "Добавить категорию">

</form>

<hr>

<form action = "php/add\_subcategory.php" method="post">

<p>Выберите категорию:</p>

<select name = "category" id = "change\_category">

<?

$Q="SELECT \* FROM main\_categorys ORDER BY categorys";

$result = $db->select($Q);

for($i=0;$i<count($result);$i++)

echo'<option>'.$result[$i]['categorys'].'</option>';

?>

</select>

<p>Имя подкатегории:</p>

<input name = "subcategory" type = "text">

<br>

<br>

<input type = "submit" value = "Добавить подкатегорию">

</form>

</div>

<div class="form">

<?

$Q="SELECT \* FROM main\_categorys ORDER BY categorys";

$result = $db->select($Q);

for($i=0;$i<count($result);$i++){

echo'<div class = "dropdown">';

echo'<a class = "deleted" href="php/deleted\_categorys.php?id='.$result[$i]['id'].'&query=main">Удалить</a>

<button class="btn dropdown-toggle" type="button" data-toggle="dropdown" >'.$result[$i]['categorys'].'</button>

<ul class="dropdown-menu"> ';

$id = [$main\_res[$i]['id']];

$sub\_res = $db->select($Q = "SELECT \* FROM categorys WHERE main\_category ={?}",$id);

for($j=0;$j<count($sub\_res);$j++){

echo'<li><a href="#">'.$sub\_res[$j]['category'].'</a ><a class = "deleted del\_sub" href="php/deleted\_categorys.php?id='.$sub\_res[$j]['id'].'&query=sub">[x]</a></li>';

}

echo'</ul>

</div>';

}

?>

</div>

</div>

</body>

<html>

<------------------------------------------------------------------->

<?php session\_start();?>

<!DOCTYPE html>

<html lang="en">

<head>

<?php

require\_once '/blocks/head.php';

require\_once '/work\_to\_db/func\_db.php';

?>

<link rel="stylesheet" type="text/css" href="style/goods.css">

</head>

<?php

$db = DataBase::getDB();

$main\_get = $\_GET['main\_id'];

$sub\_get = $\_GET['sub\_id'];

//echo "<h1>>>".$main\_get."<<</h1>";

if(isset($main\_get)){

$Q = "SELECT \* FROM `goods` WHERE main\_category = {?}";

$res = $db->select($Q,[$main\_get]);

}

else {

$Q = "SELECT \* FROM `goods` WHERE category = {?}";

$res = $db->select($Q,[$sub\_get]);

}

require\_once '/blocks/header.php';

require\_once '/blocks/nav\_bar.php';

//echo "<h1>>>".$main\_get[0]['subject']."<<</h1>";

//echo "<h1>".$sub\_get."</h1>";

if(count($res)>0)

for($i=0;$i<count($res);$i++){

echo'

<div class="all\_goods">

<div class="goods">

<img src="'.$res[$i]['img'].'" class = "img\_goods">

<div class="discription">

<p>'.$res[$i]['subject'].'</p>

<p>Цена:<span>'.$res[$i]['price'].' руб</span></p>

</div>

<div class = "operation">

<a href="/product\_detals.php?id='.$res[$i]['id'].'">Подробнее</a>

<a href="php/add\_basck.php?id='.$res[$i]['id'].'">Добавить в корзину</a>';

if($\_SESSION['status']=='admin')

echo'<a href="php/deleted.php?d\_id='.$res[$i]['id'].'">Удалить</a>';

echo'</div>

</div>

</div>

<div style = "clear:both;">

<p></p>

';

}

else echo"<p class = 'no\_goods'>Товаров нет</p>";

?>

</div>

</html>

<------------------------------------------------------------------->

<script type="text/javascript">

</script>

<div id="navigation">

<nav class="top\_m container-fluid" id="mainmenu">

<ul class="nav navbar-nav " id="menu">

<li class="item\_nav"><a href="/">ГЛАВНАЯ</a></li>

<li class="item\_nav"><a href="">О НАС</a></li>

<li class="parent"><a href="">товары</a>

<ul>

<?php require\_once '/view\_menu.php'; ?>

</ul>

</li>

<li class="item\_nav" ><a href="">СЕРВЕСНЫЙ ЦЕНТР</a></li>

<li class="item\_nav"><a href="/f\_back.php">ОБРАТНАЯ СВЯЗЬ</a></li>

<? if($\_SESSION['status']=='admin')

echo'

<li class="item\_nav"><a href="User.php">ЗАКАЗЫ(0)</a></li>

<li class="nav navbar-nav parent " ><a href="">Дополнительно</a>

<ul>

<li class="item\_nav"><a href="User.php">ПОЛЬЗОВАТЕЛИ</a></li>

<li class="item\_nav"><a href="add\_goods.php">Работа с товаром</a></li>

<li class="item\_nav"><a href="form\_menu\_regedit.php">Работа с меню</a></li>

</ul>

</li>';

?>

</ul>

</div>

</nav>

</div>

<------------------------------------------------------------------->

<?php session\_start();

if(!$\_SESSION['goods']) header("Location: index.php");

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<link rel="stylesheet" type="text/css" href="style/order\_goods.css">

<?php

require\_once '/work\_to\_db/func\_db.php';

$db = DataBase::getDB();

require\_once 'blocks/head.php';

?>

</head>

<body>

<?php

require\_once 'blocks/header.php';

require\_once 'blocks/nav\_bar.php';

?>

<div class="order\_goods container-fluid ">

<p class = "order\_cart">Корзина</p>

<hr>

<?php

$db = DataBase::getDB();

if($\_SESSION['goods'] )

foreach($\_SESSION['goods'] as $k => $val){

$P = $db->selectRow("SELECT \* FROM goods WHERE id = {?}",[$val]);

echo '<div class="row">

<div class = "col-sm-1 col-xl-1 col-lg-1">

<img src="'.$P['img'].'" alt="">

</div>

<div class = "col-sm-6 col-xl-6 col-lg-6">

<span>'.$P['subject'].'</span>

</div>

<div class = "col-sm-4 col-xl-4 col-lg-4">

<a class = "count\_range" href = "php/count\_order.php?id='.$P['id'].'&q=spare">-</a>

<span class = "order\_count">'.$\_SESSION['count\_p'][$val].'</span>

<a class = "count\_range" href = "php/count\_order.php?id='.$P['id'].'&q=add">+</a><span class = "order\_price">'.($P['price']\*$\_SESSION['count\_p'][$val]).'руб </span>

</div>

<div class = "col-sm-1 col-xl-1 col-lg-1">

<a href = "php/delet\_goods.php?id='.$P['id'].'"><img style = "width:50%" src = "images/delete.png"></a>

</div>

</div>

<hr>';

}

?>

<a href="../order\_send.php" class = 'order\_button'>Далее >></a>

</div>

<?php require\_once "blocks/discription\_order.php" ?>

</body>

</html>

<------------------------------------------------------------------->

<?php session\_start();

if(!$\_SESSION['goods']) header("Location: index.php");

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<link rel="stylesheet" type="text/css" href="style/order\_goods.css">

<?php

require\_once '/work\_to\_db/func\_db.php';

$db = DataBase::getDB();

require\_once 'blocks/head.php';

?>

</head>

<body>

<?php

require\_once 'blocks/header.php';

require\_once 'blocks/nav\_bar.php';

?>

<div class="order\_goods container-fluid ">

<p class = "order\_cart">Контактные данные</p>

<hr>

<?php $db = DataBase::getDB(); ?>

<div class="contact\_form" >

<form action="order\_success.php" method="post">

<p><span class = "label\_form">ФИО:</span><input type="text" placeholder="Ваши фамлия, имя, очество"></p>

<p><span class = "label\_form">Номер телефона:</span><input type="text" placeholder = "Ваш номер"></p>

<p><span class = "label\_form">E-mail:</span><input type="text" placeholder = "Ваша электронная почта"></p>

<p><span class = "label\_form">Адрес:</span><input type="text" placeholder="Здесь ваш адрес"> </p>

<p><span class = "label\_form">Самовывоз </span><input type="checkbox"><span> (В таком случае можно адрес не указывать)</span></p>

<p><span class = "label\_form">Общая сумма:</span><span><?php echo $\_SESSION['price'] ?> руб</span></p>

<input type="submit" class = 'order\_button' value="Сделать заказ">

</form>

</div>

</div>

<?php require\_once "blocks/discription\_order.php" ?>

</body>

</html>

<------------------------------------------------------------------->

<?php session\_start();

unset($\_SESSION['goods']);

unset($\_SESSION['count\_p']);

unset($\_SESSION['price']);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<link rel="stylesheet" type="text/css" href="style/order\_goods.css">

<?php

require\_once '/work\_to\_db/func\_db.php';

$db = DataBase::getDB();

require\_once 'blocks/head.php';

?>

</head>

<body>

<?php

require\_once 'blocks/header.php';

require\_once 'blocks/nav\_bar.php';

?>

<div class="order\_goods container-fluid ">

<p class = "order\_cart">Заказ успешно добавлен</p>

<hr>

<div class="info\_order">

<p>С Вами свяжутся в течение суток</p>

<p>

<a href="/">Вернутся на главную</a><span> | </span>

<a href="">Перейти к заказам</a>

</p>

</div>

</div>

<?php require\_once "blocks/discription\_order.php" ?>

</body>

</html>

<------------------------------------------------------------------->

<?php session\_start();?>

<!DOCTYPE html>

<html>

<head>

<?php

require\_once '/work\_to\_db/func\_db.php';

$db = DataBase::getDB();

$Q = "SELECT \* FROM `goods` WHERE id = {?}";

$res = $db-> select($Q,[$\_GET['id']]);

$title = $res[0]['subject'];

require\_once '/blocks/head.php';

?>

<link rel="stylesheet" href="style/product\_detals.css">

</head>

<body>

<?php

require\_once '/blocks/header.php';

require\_once '/blocks/nav\_bar.php';

?>

<div class="product">

<div class = 'detals'>

<span class = 'subject'><? echo $res[0]['subject']; ?></span><br>

<hr>

<img class = "image" src=<? echo"'".$res[0]['img']."'"?>>

Описание:</p><hr>

<span class = 'discription'><? echo $res[0]['description']; ?></span>

<div style = 'clear: both;'></div>

<hr>

<div class = "bar\_operation">

<p class = 'price'>Цена: <span class="price\_value"><? echo $res[0]['price']; ?></span><span> руб</span></p> <span class = "operation"><a href="php/add\_basck.php?id=<? echo $res[0]['id'] ?> ">Добавить в корзину</a></span>

</div>

</div>

<?php

if ($\_SESSION['status']=='admin')

echo'<span class = "admin\_operation"><a href="">Редактировать</a></span>

<span class = "admin\_operation"><a href="">Удалить</a></span>'; ?>

</div>

<script type="text/javascript">

$(document).ready(function() { // Ждём загрузки страницы

$(".image").click(function(){ // Событие клика на маленькое изображение

var img = $(this); // Получаем изображение, на которое кликнули

var src = img.attr('src'); // Достаем из этого изображения путь до картинки

$("body").append("<div class='popup'>"+ //Добавляем в тело документа разметку всплывающего окна

"<div class='popup\_bg'></div>"+ // Блок, который будет служить фоном затемненным

"<img src="+src+" class='popup\_img' />"+ // Само увеличенное фото

"</div>");

$(".popup").fadeIn(800); // Медленно выводим изображение

$(".popup\_bg").click(function(){ // Событие клика на затемненный фон

$(".popup").fadeOut(800); // Медленно убираем всплывающее окно

setTimeout(function() { // Выставляем таймер

$(".popup").remove(); // Удаляем разметку высплывающего окна

}, 800);

});});});

</script>

</body>

</html>

<------------------------------------------------------------------->

<? session\_start(); ?>

<!DOCTYPE html>

<html>

<head>

<?php

require\_once 'work\_to\_db/func\_db.php';

require\_once 'blocks/head.php';

$db = DataBase::getDB(); // Создаём объект базы данных

?>

<link rel="stylesheet" type="text/css" href="style/goods.css">

</head>

<body>

<?php

$S = $\_POST['search'];

$S = "%".$S."%";

require\_once 'blocks/header.php';

require\_once 'blocks/nav\_bar.php';

$Q ="select \* from `goods` where `key\_words` like {?}";

$res = $db->select($Q,[$S]);

if(count($res)>0)

for($i=0;$i<count($res);$i++){

echo'

<div class="all\_goods">

<div class="goods">

<img src="'.$res[$i]['img'].'" class = "img\_goods">

<div class="discription">

<p>'.$res[$i]['subject'].'</p>

<p>Цена:<span>'.$res[$i]['price'].'</span></p>

</div>

<div class = "operation">

<a href="/product\_detals.php?id='.$res[$i]['id'].'">Подробнее</a>

<a href="php/add\_basck.php?id='.$res[$i]['id'].'">Добавить в корзину</a>';

if($\_SESSION['status']=='admin')

echo'<a href="php/deleted.php?d\_id='.$res[$i]['id'].'">Удалить</a>';

echo'</div>

</div>

</div>

<div style = "clear:both;">

<p></p>';

}

else echo"<p class = 'no\_goods'>Поиск не дал результатов</p>";

?>

</body>

</html>

<------------------------------------------------------------------->

<?

session\_start();

require\_once "/work\_to\_db/func\_db.php";

require\_once "blocks/head.php";

$id = $\_GET['id'];

?>

<link rel="stylesheet" href="../style/mega\_menu.css">

<style>

#main {

background-color: #f2f2f2;

padding: 20px;

-webkit-border-radius: 4px;

-moz-border-radius: 4px;

-ms-border-radius: 4px;

-o-border-radius: 4px;

border-radius: 4px;

border-bottom: 4px solid #ddd;}

#real-estates-detail #author img {

-webkit-border-radius: 100%;

-moz-border-radius: 100%;

-ms-border-radius: 100%;

-o-border-radius: 100%;

border-radius: 100%;

border: 5px solid #ecf0f1;

margin-bottom: 10px;}

#real-estates-detail .sosmed-author i.fa {

width: 30px;

height: 30px;

border: 2px solid #bdc3c7;

color: #bdc3c7;

padding-top: 6px;

margin-top: 10px;}

.panel-default .panel-heading {

background-color: #fff;}

#real-estates-detail .slides li img {

height: 450px;}

.prof\_foto input[type='submit']{

width: 90%;

padding: 10px ;

background:#1c3657;

border-style: none;

color: #fff;

font-size: 14pt;}

</style>

<?php require\_once "/blocks/nav\_bar.php" ?>

<div class="container">

<div id="main">

<div class="row" id="real-estates-detail">

<div class="col-lg-4 col-md-4 col-xs-12">

<div class="panel panel-default">

<div class="panel-heading">

</div>

<div class="panel-body">

<div class="text-center" id="author">

<img src="images/user.png">

<h3><?php echo $\_SESSION['login']; ?></h3>

<hr>

<form class = 'prof\_foto'>

<input type="file" name="foto\_profile" ><br>

<input type="submit" value = "Обновить фото" name="">

</form>

</div>

</div>

</div>

</div>

<div class="col-lg-8 col-md-8 col-xs-12">

<div class="panel">

<div class="panel-body">

<ul id="myTab" class="nav nav-pills">

<li class="active"><a href="#detail" data-toggle="tab">О пользователе</a></li>

</ul>

<div id="myTabContent" class="tab-content">

<hr>

<div class="tab-pane fade active in" id="detail">

<h4> Профиль</h4>

<table class="table table-th-block">

<tbody>

<tr><td class="active">Статус:</td><td><?php echo $\_SESSION['status']; ?></td></tr>

<tr><td class="active">Почта:</td><td><?php echo $\_SESSION['mail']; ?></td></tr>

<tr><td class="active">ID:</td><td><?php echo $\_SESSION['id']; ?></td></tr>

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div><!-- /.main -->

</div><!-- /.container -->

Листинг А.2 – Блоки сайта

<div class="discription">

<div class="panel-group" id="accordion" role="tablist" aria-multiselectable="true">

<div class="acord\_panel panel panel-default">

<div class="panel-heading" role="tab" id="headingonOne">

<h4 class="panel-title">

<a class="collapsed" role="button" data-toggle="collapse" data-parent="#accordion" href="#collapse2" aria-expanded="false" aria-controls="collapse2">

Оплата

</a>

</h4>

</div>

<div id="collapse2" class="panel-collapse collapse" role="tabpanel" aria-labelledby="heading2">

<div class="panel-body">

<p><strong> 1. Наличная форма оплаты</strong></p>

<p>Вы можете оплатить Ваш заказ или ремонт наличными в нашем магазине .</p>

Вы можете оплатить Ваш заказ или ремонт наличными при получении . Обратная доставка оплаты за товар производится за счет клиента.

</div>

</div>

</div>

<div class="acord\_panel panel panel-default">

<div class="panel-heading" role="tab" id="headingTwo">

<h4 class="panel-title">

<a class="collapsed" role="button" data-toggle="collapse" data-parent="#accordion" href="#collapse3" aria-expanded="false" aria-controls="collapse3">

Доставка

</a>

</h4>

</div>

<div id="collapse3" class="panel-collapse collapse" role="tabpanel" aria-labelledby="heading3">

<div class="panel-body">

<p><strong>1. Самовывоз из магазина</strong></p>

<p>возможен по будням с 8.00 до 13.00 после резервирования товара.</p>

<p><strong>2. Доставка на дом</strong></p>

<p>возможена по будням в договоренное время до 17:00. Данная услуга стоит 100руб</p>

</div>

</div>

</div>

</div>

</div>

<------------------------------------------------------------------->

<div id="hardware" class="container">

<div class ="row hardware" >

<div class="icon\_block col-lg-4 col-md-4 col-sm-12">

<p class = "cell\_icon\_hardware"><img src="img/icon-body.png" alt=""><a href="">Детали корпуса</a></p>

<p class = "cell\_icon\_hardware"><img src="img/icon-cooler.png" alt=""><a href="">Кулера</a></p>

</div>

<div class="icon\_block col-lg-4 col-md-4 col-sm-12">

<p class = "cell\_icon\_hardware"><img src="img/icon-matrix.png" alt=""><a href="">Матрицы</a></p>

<p class = "cell\_icon\_hardware"><img src="img/icon-keyboard.png" alt=""><a href="">Клавиатуры</a></p>

</div>

<div class="icon\_block col-lg-4 col-md-4 col-sm-12">

<p class = "cell\_icon\_hardware"><img src="img/icon-power.png" alt=""><a href="">Блоки питания</a></p>

<p class = "cell\_icon\_hardware"><img src="img/icon-ram.png" alt=""><a href="">Оперативная память

</a></p>

</div>

</div>

</div>

<script type="text/javascript">

$(document).ready(function(){

$("#service").on("click","a", function (event) {

event.preventDefault();

var id = $(this).attr('href'),

top = $(id).offset().top;

$('body,html').animate({scrollTop: top}, 1500);

});

});

</script>

<------------------------------------------------------------------->

<meta charset="UTF-8">

<title><? echo $title ?></title>

<script src="../jquery/jquery.js"></script>

<script src="js\_scripts/scroll.js"></script>

<link rel="stylesheet" href="../bootstrap/css/bootstrap.css"/>

<link rel="stylesheet" href="../bootstrap/css/bootstrap-theme.css" />

<script src="../bootstrap/js/bootstrap.js"></script>

<script src="../bootstrap/js/npm.js"></script>

<script src="../jquery/jquery.js"></script>

<script src="../js\_scripts/main.js"></script>

<link rel="stylesheet" href="../style/cart\_b.css">

<link rel="stylesheet" type="text/css" href="../style/style.css">

<link rel="stylesheet" type="text/css" href="../style/reset.css">

<link rel="stylesheet" type="text/css" href="../style/form\_date.css">

<link rel="stylesheet" href="../style/prev\_cont.css">

<link rel="stylesheet" href="../style/mega\_menu.css">

<meta name="viewport" content="width=device-width, initial-scale=1">

<------------------------------------------------------------------->

<header class="">

<div>

<?php

if(!$\_SESSION['login'])

echo'<nav class="main-nav">

<ul>

<li><a class="cd-signin" href="#0">Вход</a></li>

<li><a class="cd-signup" href="#0">Регистрация</a></li>

</ul>

</nav>';

else echo '

<nav class="main-nav">

<ul>

<li><a class="cd-signin" href = "../user.php?"'.$\_SESSION['id'].' > Ваш профиль: '.$\_SESSION['login'].'</a></li>

<li><a class="cd-signin" href="../php/unset.php" style = "color:#ff0000"> Выйти </a></li>

</ul>

</nav>';

?>

</div>

<div class="clear"></div>

<div class="h-middle">

<div class="content">

<a href="nhttps://notebookoff.et/" class="logo"><img src="../images/logo.png" alt="Ремонт ноутбуков в Киеве сервисный центр НоутбукОФФ"></a>

<div class="popup-basket clr" >

<?php

if($\_SESSION['goods'] ){

echo '<div class="cart\_b">';

$db = DataBase::getDB();

if($\_SESSION['goods'] )

foreach($\_SESSION['goods'] as $k => $val){

$P = $db->selectRow("SELECT \* FROM goods WHERE id = {?}",[$val]);

echo '<p>

<img src="'.$P['img'].'" alt=""><span class = "subject\_b" >'.$P['subject'].'</span>

<p class = "over\_price">Цена: <span class = "price\_b">'.$P['price'].'руб </span><span class = "count\_b">x'.$\_SESSION['count\_p'][$val].'</span></p>

</p>

<hr>';

echo '<a href="../php/unset\_bask.php" >Очистить </a> <a href="../order\_goods.php" >Заказать</a>';

echo '</div>';

} ?>

<strong class="title">Корзина<span class="value" id="cart-total"><?

if($\_SESSION['price'])

echo $\_SESSION['price'];

else echo "0"?> руб</span></strong>

<span class="toggle"></span>

</div>

<div class="search clr" id="search" >

<form action = "../result\_search.php" method="POST">

<input type="text" class="t-val" name="search" value="" placeholder="Поиск... введите в строку Ваш запрос">

<div class="ctrl-btns clr">

<button><img src="../images/icon-search.png" alt=""></button>

</div>

</form>

</div>

</div>

</div>

</header>

<div class="clear"></div>

<?php if(!$\_SESSION['login']) require\_once 'reg\_auth\_form.php'; ?>

<------------------------------------------------------------------->

<div id="navigation">

<nav class="top\_m container-fluid" id="mainmenu">

<ul class="nav navbar-nav " id="menu">

<li class="item\_nav"><a href="/">ГЛАВНАЯ</a></li>

<li class="item\_nav"><a href="">О НАС</a></li>

<li class="parent"><a href="">товары</a>

<ul>

<?php require\_once '/view\_menu.php'; ?>

</ul>

</li>

<li class="item\_nav" ><a href="">СЕРВЕСНЫЙ ЦЕНТР</a></li>

<li class="item\_nav"><a href="/f\_back.php">ОБРАТНАЯ СВЯЗЬ</a></li>

<? if($\_SESSION['status']=='admin')

echo'

<li class="nav navbar-nav parent"><a href="User.php">Cообщения(0)</a>

<ul>

<li class="item\_nav"><a href="add\_goods.php">Заказы на товар(0)</a></li>

<li class="item\_nav"><a href="form\_menu\_regedit.php">Заявки на ремонт(0)</a></li>

</ul>

</li>

<li class="nav navbar-nav parent " ><a href="">Дополнительно</a>

<ul>

<li class="item\_nav"><a href="User.php">ПОЛЬЗОВАТЕЛИ</a></li>

<li class="item\_nav"><a href="add\_goods.php">Работа с товаром</a></li>

<li class="item\_nav"><ahref="form\_menu\_regedit.php">Работа с меню </a></li>

</ul>

</li>'; ?>

</ul>

</div>

</nav>

</div>

<------------------------------------------------------------------->

<div class="cd-user-modal"> <!-- все формы на фоне затемнения-->

<div class="cd-user-modal-container">

<ul class="cd-switcher">

<li><a href="#0">Вход</a></li>

<li><a href="#0">Регистрация</a></li>

</ul>

<div id="cd-login"> <!-- форма входа -->

<form class="cd-form" action="../php/login.php" method="post"><p class="fieldset">

<label class="image-replace cd-email" for="signin-email">Ваш логин</label>

<input name = 'login' class="full-width has-padding has-border" id="signin-email" type="text" placeholder="Логин">

<span class="cd-error-message"></span>

</p>

<p class="fieldset">

<label class="image-replace cd-password" for="signin-password">Пароль</label>

<input name = 'password' class="full-width has-padding has-border" id="signin-password" type="password" placeholder="Пароль">

<a href="#0" class="hide-password">Показать</a>

<span class="cd-error-message">Здесь сообщение об ошибке!</span>

</p>

<p class="fieldset"> <input class="full-width" type="submit" value="Войти">

</p>

</form>

<p class="cd-form-bottom-message"><a href="#0">Забыли свой пароль?</a></p>

<a href="#0" class="cd-close-form">Close</a>

</div> <!-- cd-login -->

<div id="cd-signup"> <!-- форма регистрации -->

<form class="cd-form" action="../php/regisr.php" method="post">

<p class="fieldset">

<label class="image-replace cd-username" for="signup-username">Ваш логин</label>

<input name = "login" class="full-width has-padding has-border" id="signup-username" type="text" placeholder="Имя пользователя">

<span class="cd-error-message">Здесь сообщение об ошибке!</span>

</p>

<p class="fieldset">

<label class="image-replace cd-email" for="signup-email">E-mail</label>

<input name = "mail" class="full-width has-padding has-border" id="signup-email" type="text" placeholder="E-mail">

<span class="cd-error-message">Здесь сообщение об ошибке!</span>

</p>

<p class="fieldset">

<label class="image-replace cd-password" for="signup-password">Пароль</label>

<input name = "password" class="full-width has-padding has-border" id="signup-password" type="password" placeholder="Пароль">

<a href="#0" class="hide-password">Показать</a><span class="cd-error-message">Здесь сообщение об ошибке!</span></p>

<p class="fieldset">

<input class="full-width has-padding" type="submit" value="Создать аккаунт">

</p>

</form>

<a href="#0" class="cd-close-form">Close</a>

</div> <!-- cd-signup -->

<div id="cd-reset-password">

<p class="cd-form-message">Забыли пароль? Пожалуйста, введите адрес своей электронной почты. Вы получите ссылку, чтобы создать новый пароль.</p>

<form class="cd-form">

<p class="fieldset">

<label class="image-replace cd-email" for="reset-email">E-mail</label>

<input class="full-width has-padding has-border" id="reset-email" type="email" placeholder="E-mail">

<span class="cd-error-message">Здесь сообщение об ошибке!</span>

</p>

<p class="fieldset">

<input class="full-width has-padding" type="submit" value="Восстановить пароль">

</p>

</form>

<p class="cd-form-bottom-message"><a href="#0">Вернуться к входу</a></p>

</div> <!-- cd-reset-password -->

<a href="#0" class="cd-close-form">Закрыть</a>

</div></div>

<------------------------------------------------------------------->

<div id="service" class="prev\_content" >

<div class="slide">

<p><a href="">Нужен ремонт?</a></p>

<img src="../images/repair.jpg" alt="">

</div>

<div class="slide">

<p><a href="#hardware">Ищете комплетующие?</a></p>

<img src="../images/hardware.jpg" alt="">

</div>

</div>

<script type="text/javascript">

$(document).ready(function(){

var H = $(window).innerHeight();

H/=2;

H = Math.round(H);

var doc = $('.slide img').css('height',H);

console.log(H);

});

</script>

Листинг А.3 – Класс работы с базой данных

<?php

class DataBase {

private static $db = null; // Единственный экземпляр класса, чтобы не создавать множество подключений

private $mysqli; // Идентификатор соединения

private $sym\_query = "{?}"; // "Символ значения в запросе"

/\* Получение экземпляра класса. Если он уже существует, то возвращается, если его не было, то создаётся и возвращается (паттерн Singleton) \*/

public static function getDB() {

if (self::$db == null) self::$db = new DataBase();

return self::$db;}

/\* private-конструктор, подключающийся к базе данных, устанавливающий локаль и кодировку соединения \*/

private function \_\_construct() {

$this->mysqli = new mysqli("localhost", "root", "", "restart");

$this->mysqli->query("SET lc\_time\_names = 'ru\_RU'");

$this->mysqli->query("SET NAMES 'utf8'");

if ($mysqli->connect\_errno) {

printf("Не удалось подключиться: %s\n", $mysqli->connect\_error);

exit();}}

private function getQuery($query, $params) {

if ($params) {

for ($i = 0; $i < count($params); $i++) {

$pos = strpos($query, $this->sym\_query);

$arg = "'".$this->mysqli->real\_escape\_string($params[$i])."'";

$query = substr\_replace($query, $arg, $pos, strlen ( $this->sym\_query));}}

return $query;}

public function select($query, $params = false) {

$result\_set = $this->mysqli->query($this->getQuery($query, $params));

if (!$result\_set) return false;

return $this->resultSetToArray($result\_set);

}

public function selectRow($query, $params = false) {

$result\_set = $this->mysqli->query($this->getQuery($query, $params));

if ($result\_set->num\_rows != 1) return false;

else return $result\_set->fetch\_assoc();}

public function selectCell($query, $params = false) {

$result\_set = $this->mysqli->query($this->getQuery($query,$params));

if ((!$result\_set) || ($result\_set->num\_rows != 1)) return false;

else {

$arr = array\_values($result\_set->fetch\_assoc());

return $arr[0];} }

public function query($query, $params = false){

$success = $this->mysqli->query($this->getQuery($query,$params));

if ($success) {

if ($this->mysqli->insert\_id === 0) return true;

else return $this->mysqli->insert\_id;}

else return false;}

private function resultSetToArray($result\_set) {

$array = array();

while (($row = $result\_set->fetch\_assoc()) != false) {

$array[] = $row;}

return $array;}

public function \_\_destruct() {

if ($this->mysqli) $this->mysqli->close();}}

?>

Листинг А.4 – Стили сайта

body{

background: #F5F5DC;}

a{

text-decoration: none!important;}

header {

background: #0c2d2f;

height: 130px;}

.row{

margin-right:0!important;}

.regisr\_auth{

float: right;}

.regisr\_auth span{

margin: 2px 10px;}

.h-middle {

padding: 9px 0;

background: #143840;}

.logo img {

display: block;

width: 15%;

float: left; }

.popup-basket {

float: right;

margin: 8px 0 8px 26px;

position: relative;

cursor: pointer;}

.popup-basket .title {

float: left;

height: 69px;

padding: 14px 39px 11px 16px;

font: 700 14px "Open Sans",sans-serif;

color: #fff;

text-transform: uppercase;

border: 1px solid #1a5361;}

.content {

max-width: 1280px;

width: 95%;

margin: 0 auto;

position: relative;}

.popup-basket {

float: right;

margin: 8px 0 8px 26px;

position: relative;}

.search {

float: right;

width: 64.53%;

margin: 20px 0;

padding-right: 90px;

position: relative;}

.search .ctrl-btns button {

float: left;

width: 45px;

height: 45px;

padding: 0;

position: relative;

line-height: 45px;

text-align: center;

cursor: pointer;

border: none;

background: none;}

.search .ctrl-btns button img {

vertical-align: middle;}

.search .t-val {

display: block;

width: calc(100% + 45px);

color: rgb(52, 52, 52);

padding: 13px 23px;

font: 400 14px "Open Sans", sans-serif;

border-width: initial;

border-style: none;

border-color: initial;

border-image: initial;}

.content::after {

content: "";

display: block;

clear: both;}

.search .ctrl-btns {

position: absolute;

top: 0px;

right: 0px;

background: rgb(26, 83, 97);}

.popup-basket .title .value {

display: block;

color: #f4b102;}

.popup-basket .toggle {

float: left;

width: 69px;

height: 69px;

background: #1a5361 url(../images/icon-basket.png) center no-repeat;}

@media(max-width:1200px){

.logo\_photo{

height: 45px;}

.bask\_photo{

height: 45px;}

.login,.login a{

font-size: 14pt;}

.search input{

height: 45px;

font-size: 10pt!important;}}

@media(max-width:1190px){

.search {

width: 55%;}}

.clear{

clear: both;}

.product{

padding: 10px;

width: 65%;

margin: 0 auto;

background:#fff;}

.product img{

display: block;

width: 35%;

float: left;

border: 1px solid #000;

padding: 10px;

margin-right: 5px;}

.detals {

border: 1px solid #000;

margin: 10px;

padding: 5px;}

.detals p{

font-weight: bold;

font-size: 26px;

color: #111;}

.bar\_operation{

display: inline-block;

margin: auto;

width: 100%}

.bar\_operation span {

font-size: 25px;}

.operation a{

color:white;

font-size: 18px;

letter-spacing: 2px;

font-weight: 800;}

.operation {

display: inline-block;

padding: 10px 0px;

margin: 2px;

background:#1b5261;

width: 100%;

text-align: center;}

.operation:hover{

background: #008db4;}

.price{

font-size: 20px;}

.price\_value{

color: orange;

font-weight: 800;}

.image {

cursor:pointer;}

.popup {

position: absolute;

height:100%;

width:100%;

top:0;

left:0;

display:none;

text-align:center;}

.popup\_bg {

background:rgba(0,0,0,0.4);

position:fixed;

z-index:119;

height:100%;

width:100%;}

.popup\_img {

position: relative;

margin:0 auto;

z-index:120;

height:94%;

margin:1% 0 0 0;}

.subject{

font-size: 25px;

color: #4a4a4a;}

.discription{

text-align: justify;

font-size: 18px;

word-spacing: 5px;}

.detals hr{

border-top:1px solid #000;}

.admin\_operation a:hover{

background: #f00;}

.admin\_operation a{

margin: 5px;

padding: 5px 10px;

background: #007462;

color: #fff;}

.order\_goods img{

width: 100%;}

.order\_goods{

width: 60%;

margin: auto;

background: #fff;}

.acord\_panel{

margin: 0!important;}

.panel-default > .panel-heading{

background:none!important;

border-radius: none!important;}

.order\_cart{

text-align: center;

font-size: 24px;

letter-spacing: 3px;

color: red;

text-transform: uppercase;

font-weight: 800;}

.order\_button {

display: block;

width: 400px;

margin:5px auto;

text-align: center;

padding: 20px 0px;

background: #005050;

font-size: 20px;

color: aliceblue;}

.order\_button:hover{

color: aliceblue;

background: #500000;}

.count\_range{

background: #000000;

font-weight: bolder;

color: aliceblue;

padding: 5px 10px;

font-size:20px;}

.order\_count{

background: #000000;

font-weight: bolder;

color: aliceblue;

padding: 5px 10px;

font-size:20px;}

.order\_price{

font-weight: bolder;

color: #00392c;

padding: 5px 10px;

font-size:20px;}

.prev\_content{

width: 100%;

overflow: hidden;}

.slide{

margin-top: 10px;

width: 50%;

float: left;

height: 100%;}

.slide p{

top: 40%;

width: 50%;

font-size: 25pt;

position: absolute;

text-align: center;

vertical-align: middle;}

.slide img{

width: 100%;}

@media(max-width:1190px){

.slide p{

font-size: 14pt;}}

input[type="button"]:not(.default), input[type="submit"]:not(.default) {

-webkit-border-radius: 3px;

-webkit-background-clip: padding-box;

-moz-border-radius: 3px;

-moz-background-clip: padding;

border-radius: 3px;

background-clip: padding-box;

-webkit-transition: color 0.2s ease, border 0.2s ease, background 0.2s ease, -webkit-box-shadow 0.2s ease;

-moz-transition: color 0.2s ease, border 0.2s ease, background 0.2s ease, -moz-box-shadow 0.2s ease;

-o-transition: color 0.2s ease, border 0.2s ease, background 0.2s ease, box-shadow 0.2s ease;

transition: color 0.2s ease, border 0.2s ease, background 0.2s ease, box-shadow 0.2s ease;

position: relative;

margin: 0 7px;

display: inline-block;

min-width: 144px;

max-width: 100%;

padding: 15px 25px;

font-family: "Arial", "Helvetica Neue", Arial, Helvetica, sans-serif;

font-size: 14px;

font-weight: 600;

text-transform: uppercase;

line-height: 1;

border-width: 1px;

border-style: solid;

background-color: #288689;

color: #717171;}

.top\_auth{ margin-top: 10px; }

.form {

zoom: 1;

display: block;

width: auto;

padding: 25px 0 0;}

.form:before,

.form:after {

content: "";

display: table;}

.form:after { clear: both;}

.form .form-group {

zoom: 1;

position: relative;

margin-bottom: 25px;}

.form .form-group:before,

.form .form-group:after {

content: "";

display: table;}

.form .form-group:after { clear: both;}

.form .form-group:after {

content: '';

display: block;

clear: both;}

.form .form-group[class\*="col-"] input[type=text],

.form .form-group[class\*="col-"] input[type=email],

.form .form-group[class\*="col-"] input[type=password] {

display: inline-block;

width: 100%;

min-width: 0;

max-width: 100%;}

.form .form-control {

-webkit-border-radius: 0;

-webkit-background-clip: padding-box;

-moz-border-radius: 0;

-moz-background-clip: padding;

border-radius: 0;

background-clip: padding-box;

-webkit-box-shadow: none;

-moz-box-shadow: none;

box-shadow: none;

height: auto;

font-size: 17px;

font-weight: 300;

background-color: #ffffff;

border-style: solid;

border-width: 1px 1px 1px 3px;

border-top-color: #d7d7d7;

border-right-color: #d7d7d7;

border-bottom-color: #d7d7d7;}

input[type=text],

input[type=email],

input[type=password] {

-webkit-transition: color 0.2s ease, background 0.3s ease;

-moz-transition: color 0.2s ease, background 0.3s ease;

-o-transition: color 0.2s ease, background 0.3s ease;

transition: color 0.2s ease, background 0.3s ease;

-webkit-border-radius: 0;

-webkit-background-clip: padding-box;

-moz-border-radius: 0;

-moz-background-clip: padding;

border-radius: 0;

background-clip: padding-box;

-webkit-box-shadow: none;

-moz-box-shadow: none;

box-shadow: none;

margin: 0;

vertical-align: top;

display: inline-block;

width: 100%;

font-size: 17px;

color: #8d8d8d;

background-color: #ffffff;

border-style: solid;

border-width: 1px 1px 1px 3px;

border-top-color: #d7d7d7;

border-right-color: #d7d7d7;

border-bottom-color: #d7d7d7;

outline: none;}

input[type=text]:hover,

input[type=email]:hover,

input[type=password]:hover {

color: #000000;}

input[type=text]:focus,

input[type=email]:focus,

input[type=password]:focus {

outline: none;

border-left-color: steelblue;}

.well h3 {

text-shadow: -1px -1px #ade0ff, -2px -2px #cdcbff, -1px 1px #FFF, -2px 2px #FFF, 1px 1px #FFF, 2px 2px #FFF, 1px -1px #FFF, 2px -2px #FFF, -3px -3px 2px #BBB, -3px 3px 2px #BBB, 3px 3px 2px #BBB, 3px -3px 2px #BBB;

color: steelblue;

transition: all 1s;}

input[type='submit']{

display: block;

margin:10px auto;

padding: 20px 150px;}

.sub\_key {padding:5px 10px; width:100%;}

.sub\_form\_block p,.sub\_form\_block span{

color:#2F4F4F;

font-size:18pt;}

.sub\_category\_block{margin-left:100px;}

.main\_category{

cursor: pointer;

display: inline-block;

padding: 5px;}

.wrapp{

width:60%;

margin:0 auto;}

.form{

margin: 0 auto;

width:100%;}

textarea{width: 100%;

height:150px;

resize: none;}

.sub\_form\_block{

float: left;

padding: 10px;

width: 50%;

height: auto;}

.sub\_form\_block:first-child{ border-right: 1px solid #000;}

.sub\_form\_block .dropdown{ border-bottom: 1px solid #000;}

.sub\_form\_block .dropdown button {background: none!important;}

.sub\_form\_block .dropdown-menu {

border: none!important;

box-shadow: none!important;

left:50px;

padding: 10px;}

.sub\_form\_block .dropdown-menu a{

display: inline-block;}

.radi { display: none;}

.checkbox-custom,

::-moz-focus-inner {

padding: 0;

border: 0;}

:-moz-placeholder {

color: #879fa6 !important;}

::-webkit-input-placeholder {

color: #879fa6;}

:-ms-input-placeholder {

color: #879fa6 !important;}

body {

font: 12px/20px 'Lucida Grande', Verdana, sans-serif;

color: #404040;}

input, textarea, select, label {

font-family: inherit;

font-size: 16px;

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;}

.contact {

position: relative;

margin: 50px auto;

padding: 5px;

width: 30%;

background: #eef5f7;

border: 1px solid #cfd5da;

border-bottom-color: #ccd1d6;

border-radius: 3px;

-webkit-box-shadow: 0 1px 1px rgba(0, 0, 0, 0.05);

box-shadow: 0 1px 1px rgba(0, 0, 0, 0.05);}

.contact-inner {

padding: 25px;

background: white;

border-radius: 2px;}

.contact:before, .contact:after, .contact-inner:before, .contact-inner:after {

content: '';

position: absolute;

top: 100%;

left: 50%;

margin-left: -6px;

width: 1px;

height: 1px;

border: outset transparent;

border-width: 12px 14px 0;

border-top-style: solid;

-webkit-transform: rotate(360deg);}

.contact:before {

margin-top: 1px;

border-top-color: #d8e1e6;}

.contact-inner:before {

border-top-color: #ccd1d6;}

.contact-inner:after {

margin-top: -1px; border-top-color: #eef5f7;}

.contact:after {

margin-top: -8px;

border-top-color: white;}

.contact-input {

overflow: hidden;

margin-bottom: 20px;

padding: 5px;

background: #eef7f9;

border-radius: 2px;}

.contact-input > input, .contact-input > textarea {

display: block;

width: 100%;

height: 29px;

padding: 0 9px;

color: #4d5a5e;

background: white;

border: 1px solid #cfdfe3;

border-bottom-color: #d2e2e7;

border-radius: 2px;

-webkit-box-shadow: inset 0 1px 2px rgba(0, 0, 0, 0.05), 0 1px rgba(255, 255, 255, 0.2);

box-shadow: inset 0 1px 2px rgba(0, 0, 0, 0.05), 0 1px rgba(255, 255, 255, 0.2);}

.contact-input > input:focus, .contact-input > textarea:focus {

border-color: #93c2ec;

outline: 0;

-webkit-box-shadow: 0 0 0 8px #e1ecf5;

box-shadow: 0 0 0 8px #e1ecf5;}

.lt-ie9 .contact-input > input, .lt-ie9 .contact-input > textarea {

line-height: 27px;}

.contact-input > textarea {

padding: 4px 8px;

height: 150px;

line-height: 20px;

resize: none;}

.select {

display: block;

position: relative;

overflow: hidden;

background: white;

border: 1px solid #d2e2e7;

border-bottom-color: #c5d4d9;

border-radius: 2px;

background-image: -webkit-linear-gradient(top, #fcfdff, #f2f7f7);

background-image: -moz-linear-gradient(top, #fcfdff, #f2f7f7);

background-image: -o-linear-gradient(top, #fcfdff, #f2f7f7);

background-image: linear-gradient(to bottom, #fcfdff, #f2f7f7);

-webkit-box-shadow: 0 1px 2px rgba(0, 0, 0, 0.06);

box-shadow: 0 1px 2px rgba(0, 0, 0, 0.06);}

.select:before, .select:after {

content: '';

position: absolute;

right: 11px;

width: 0;

height: 0;

border-left: 3px outset transparent;

border-right: 3px outset transparent;}

.select:before {

top: 10px;

border-bottom: 3px solid #7f9298;}

.select:after {

top: 16px;

border-top: 3px solid #7f9298;}

.select > select {

position: relative;

z-index: 2;

width: 112%;

height: 29px;

line-height: 17px;

padding: 5px 9px;

padding-right: 0;

color: #80989f;

background: transparent;

background: rgba(0, 0, 0, 0);

border: 0;

-webkit-appearance: none;}

.select > select:focus { color: #4d5a5e; outline: 0;}

.contact-submit {

text-align: right;}

.contact-submit > input {

display: inline-block;

vertical-align: top;

padding: 0 14px;

height: 29px;

font-weight: bold;

color: #729fb2;

text-shadow: 0 1px rgba(255, 255, 255, 0.5);

background: #deeef4;

border: 1px solid #bed6e3;

border-bottom-color: #accbd9;

border-radius: 15px;

cursor: pointer;

background-image: -webkit-linear-gradient(top, #e6f2f7, #d0e6ee);

background-image: -moz-linear-gradient(top, #e6f2f7, #d0e6ee);

background-image: -o-linear-gradient(top, #e6f2f7, #d0e6ee);

background-image: linear-gradient(to bottom, #e6f2f7, #d0e6ee);

-webkit-box-shadow: inset 0 1px rgba(255, 255, 255, 0.2), 0 1px 1px rgba(0, 0, 0, 0.06), 0 0 0 4px #eef7f9;

box-shadow: inset 0 1px rgba(255, 255, 255, 0.2), 0 1px 1px rgba(0, 0, 0, 0.06), 0 0 0 4px #eef7f9;}

.contact-submit > input:active {

color: #6a95a9;

text-shadow: 0 1px rgba(255, 255, 255, 0.3);

background: #c9dfe9;

border-color: #a3bed0 #b5ccda #b5ccda;

/\*-webkit-box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1), 0 1px rgba(255, 255, 255, 0.2), 0 0 0 4px #eef7f9;\*/

box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1), 0 1px rgba(255, 255, 255, 0.2), 0 0 0 4px #eef7f9;}