Содержание

[Введение 5](#_Toc67936973)

[1 Теоретический раздел 7](#_Toc67936974)

[1.1 Анализ предметной области 7](#_Toc67936975)

[1. 2 Назначения и цели создания системы 10](#_Toc67936976)

[1.3 Постановка задачи 11](#_Toc67936979)

[1.4 Описание входной и выходной информации 13](#_Toc67936980)

[1.5 Инфологическая модель предметной области 14](#_Toc67936981)

[1.6 Логическая модель данных 15](#_Toc67936982)

[1.7 Нормализация модели данных 16](#_Toc67936983)

[1.8 Обоснование выбора вида программного обеспечения 18](#_Toc67936984)

[1.9 Физическая модель данных 21](#_Toc67936985)

[1.10 Анализ рынка 25](#_Toc67936986)

[2 Практический раздел 28](#_Toc67936987)

[2.1 Требования к информационной и программной совместимости 28](#_Toc67936988)

[2.2 Разработка пользовательского интерфейса 30](#_Toc67936989)

[2.3 Тестирование программы 33](#_Toc67936990)

[2.4 Анализ надежности и качества 34](#_Toc67936991)

[Заключение 36](#_Toc67936992)

[Список использованный источников 38](#_Toc67936993)

[Приложение А 39](#_Toc67936994)

[Приложение Б 4](#_Toc67936995)0

# Введение

Использование Интернет технологий в коммерческой деятельности — одна из наиболее актуальных задач в современном мире. ХХI век уже невозможно представить без интернета и интернет технологий. Всемирная паутина прочно вошла в нашу жизнь.

Наиболее активными пользователями сети интернет являются молодежь, жители крупных городов, а также люди с высоким уровнем дохода. Следовательно, можно сказать, что было бы крайне неразумно оставлять без внимания потребительские запросы этой аудитории, не попытавшись удовлетворить хотя бы часть таких запросов прямо в Интернете.

В отличие от обычного магазина, интернет-магазин открыт 24 часа в сутки, 7 дней в неделю, доступен любому пользователю в любой точке планеты и не требует финансовых затрат на аренду помещения, значительно снижает издержки на персонал.

Интернет-магазин сочетает невысокую стоимость и высокую эффективность: создание Интернет-магазина помогает компании завоевать или укрепить свои позиции на рынке товаров и услуг, увеличить клиентскую базу и повысить популярность бренда. Иными словами, создание интернет-магазина приносит владельцам реальную прибыль. Именно поэтому в настоящее время все большее количество компаний приходит к решению о создании Интернет-магазина.

Популярность Интернет магазинов растёт с каждым днём. Уже ни для кого не секрет, что хорошо раскрученный Интернет магазин приносит прибыли не меньше, чем его обычный аналог, причём содержать и изготовить Интернет магазин на порядок выгодней — нет затрат на дорогую аренду и других расходов.

Целью курсовой работы является создание автоматизированного рабочего места по учёту продаж в салоне сотовой связи, обеспечивающего удобство, безопасность и интуитивную понятность интерфейса пользователю, а также безошибочную, хорошо организованную работу персонала магазина.

Для достижения поставленной цели ставятся следующие задачи:

1. Изучить функциональные возможности интернет-магазина.
2. Ознакомиться с рынком труда.
3. Определить целевую аудиторию.
4. Мониторинг потенциальных конкурентов из числа существующих интернет-магазинов.
5. Изучить основные принципы проектирования БД и приложения.
6. Изучить основные разработки пользовательского интерфейса.
7. Разработать техническое задание на проект.
8. Разработать интернет-магазин по изученному материалу.
9. Найти потенциальных пользователей интернет-магазина.
10. Внедрить данный интернет-магазин на производство.
11. Дальнейшая поддержка интернет-магазина по потребностям заказчика.

# 1 Теоретический раздел

## 1.1 Анализ предметной области

Рабочее место по учёту продаж в салоне сотовой связи «MobExpress» – это сайт, в котором торгуют товарами с помощью сети Интернет. Данное рабочее место позволяет покупателям онлайн выбирать товар, способ оплаты, метод доставки товара и способ его оплаты.

Данный сайт по учёту продаж в салоне сотовой связи создавался с помощью редактора исходного кода Visual Studio Code, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга.

В современном мире не найдёшь такого человека, который не пользуется цифровыми технологиями. В век IT-технологий, каждый человек в своём доме, на работе имеет компьютер, ноутбук или планшет; для разговора на удалённых расстояниях использует сотовые телефоны.

Для использования данного сайта, пользователю не обязательно иметь мощный компьютер, достаточно иметь минимальные его характеристики:

1. Процессор – AMD A8-7600 3.1-3.8GHz Tray
2. Кулер для процессора – Zalman CNPS8000B 130W
3. Материнская плата – ASRock FM2A68M-HD+ (A68)
4. Видеокарта – интегрированная Radeon R7 (384) 720MHz
5. Жесткий диск – Western WD5000AAKX 500GB
6. Корпус + БП – Aerocool Qs-102 400W

Интернет-магазин цифровой техники «MobExpress» быстро и качественно обслуживает клиентов. База данных используется, как администратором и менеджерами, так и клиентами.

На рисунке 1 изображена организационная структура интернет-магазина «MobExpress», которая состоит из: руководителя(администратора), отдела снабжения, отдела продаж, отдела контроля качества, склада, отдела клиентских отношений.

Руководитель (администратор)

Отдел продаж

Отдел снабжения

Отдел контроля качества

Менеджеры

Отдел клиентских отношений

Склад

Рисунок 1 – Организационная структура предприятия

Руководитель (администратор) – производит общее руководство деятельности магазина, контролирует работу отдела продаж.

Менеджеры дополняют товары и категории, отправляют товар клиентам, проверяют доставку и оплату.

Отдел по продажам занимается консультацией клиентов по эксплуатации, приобретённой ими продукции, и её гарантированного обслуживания.

Отдел по контролю качества занимается ведением контроля качества предоставляемого товара в интернет-магазине.

**Преимущества интернет-магазина «MobExpress»:**

1. Цены товаров намного ниже, чем в других интернет-магазинах
2. Быстрый отклик на заявку клиента
3. Интуитивно понятный интерфейс как для администратора, у которого есть все привилегии, так и для менеджеров, с ограниченными правами, также и для покупателей.

Для совершения покупки заинтересовавших товаров, их нужно положить в корзину для покупок. Товар помещается в корзину нажатием кнопки «Купить», далее можно продолжить выбор или перейти в корзину для дальнейшего оформления заказа.

Нажатие кнопки «Купить» ни к чему не обязывает. Всегда можно удалить товар из корзины, отложить его для последующей покупки или вообще не делать заказ.

Посмотреть содержимое корзины можно в любой момент нажатием кнопки «Просмотр корзины» в заголовке окошка корзины или пиктограммы с изображением пользовательской корзины.

После того как пользователь нажал на корзину, под наименованием товара происходит вся процедура оформления заказа и его оплаты.

## 1. 2 Назначения и цели создания системы

## Возможности, которые открывает интернет магазин, неисчерпаемы. Помимо информационной направленности, интернет позволяет осуществлять ведение бизнеса. Этим фактором обусловлена востребованная разработка сайтов в сети по созданию интернет-магазинов. Этот вид бизнеса малозатратен, удобен, позволяет добиться значительных преимуществ в экономии и решить различные производственные задачи.

## **Преимущества интернет-магазина MobExpress:**

* Возможность круглосуточной работы без выходных и праздничных дней.
* Постоянная реальная прибыль от беспрерывной работы.
* Отсутствие арендной платы за торговое помещение и оборудование.
* Отсутствие необходимости оплаты труда многочисленного персонала: продавцов, кассиров, охранников, администраторов и др. Достаточно одного или двух администраторов сайта.
* Неограниченная база потенциальных клиентов, которые могут ознакомиться с ассортиментом, независимо от своего местонахождения.
* Значительная экономия, позволяющая обойтись без вливания больших финансовых средств в строительство обычного магазина.
* Возможность общения с потенциальными клиентами в круглосуточном режиме онлайн, отвечая на интересующие вопросы.

## 1.3 Постановка задачи

Процесс создания интернет-магазина включает в себя следующие этапы постановки задач:

Таблица 1 – Основные задачи проекта

|  |  |  |
| --- | --- | --- |
| № | Задачи | Количество дней |
| 1 | Постановка задачи, определение целей и технологий разработки | 1 |
| 2 | Написание формального технического задания на интернет-магазин | 4 |
| 3 | Создание графического дизайна для всех типов страниц сайта (главная страница, страница с описанием товара, страница каталога товаров, страница добавления товара в корзину) | 10 |
| 4 | Вёрстка страниц web-сайта | 8 |
| 6 | Обучение менеджеров интернет-магазина | 1 |
| 7 | Наполнение базы данных сайта контентом (описание товара, оплата, доставка и т.п.) | 3 |
| 8 | Подготовка транспортного файла с каталогом товаров и начальная загрузка его в интернет-магазин | 1 |
| 9 | Тестирование всех страниц и процедур на корректность и работоспособность | 1 |
| 10 | Внедрение интернет-магазина на производство | 1 |

Интернет- магазин должен:

1. информировать клиентов о предлагаемом ассортименте компьютерных комплектующих (цена товара, производитель, технические характеристики и т. п.);
2. осуществлять online-заказы (регистрация, оформление бланка заказа);
3. носить рекламный характер для привлечения новых клиентов.

Создание интернет-магазина — это наиболее простой и относительно недорогой способ распространения рекламы.

Проект должен отвечать следующим требованиям:

1. Находящаяся информация на сайте должна быть краткой и предоставлять полное описание о товаре.
2. Информация должна быть распределена по разделам и сопровождена фотографиями и иллюстрациями для большей наглядности.
3. Содержать процедуру регистрации, а также системы защиты от несанкционированного и неавторизированного доступа.
4. Содержать обратную связь с продавцом
5. Содержать возможность поиска товаров по ключевым словам.
6. Возможность поделится товаром с другом в социальных сетях.

## 1.4 Описание входной и выходной информации

Входной информацией для интернет-магазина является:

* информация о заказчике, предоставляемая им в процессе регистрации в системе;
* информация о размещении заказчиком заказа покупки товаров и подробности этого заказа;
* информация об оплате заказчиком счета;
* сообщения, направляемые клиентом в службу поддержки.

Выходными данными для интернет-магазина являются:

* счёт, выставляемый системой заказчику для оплаты его заказа;
* список покупок, находящихся в процессе доставки из интернет-магазина
* список покупок, заказанных к отправке заказчикам;
* список посылок, отправленных заказчикам, а также информация о содержащихся в них покупках.

## 1.5 Инфологическая модель предметной области

Инфологическая модель данных, в данном проекте создана для того, чтобы посмотреть характеристики на товар, пользователю необходимо выбрать его из списка интернет-магазина. При клике на товар, идёт запрос в базу данных и открывается сам товар и его комплектующие. При возникшем вопросе, клиент пишет администратору сайта или онлайн консультанту магазина.

При покупке товара клиентом, вся информация об оформлении заказа клиентом также формируется в базе данных, после чего формируется чек на оплату данного товара. После того, как клиент заполнил все формы при оформлении заказа, ему надо оплатить товар, а уже после оплаты, товар готовится к отправке клиенту.

Таблица 2 – Схема информационных потоков

|  |  |  |  |
| --- | --- | --- | --- |
| Отдел  Работы | Пользователь | База данных | Администратор |
| Поиск и просмотр данных |  |  |  |
| Обработка запроса |  |  |  |
| Ответ на вопрос пользователя |  |  |  |
| Оформление заказа |  |  |  |
| Чек на оплату |  |  |  |
| Оплата товара |  |  |  |
| Отправка товара клиенту |  |  |  |

## 1.6 Логическая модель данных

Авторизация

Заполнение полей

-

Главная страница

Начало

Завести аккаунт

Проверка

Внесение исправлений

-

+

+

Проверка

+

Конец

Рисунок 2 – Алгоритм работы

Вход

-

Выбор товара

Заявка на товар

+

-

Оплата товара

Деньги есть

Внесение денег

-

Завершение покупки

+

Чек об оплате

Рисунок 4 – Алгоритм работы

## 1.7 Нормализация модели данных

При разработке модели базы данных использовался язык структурированных запросов (SQL). Данная база данных состоит из 47 таблиц:

* admin;
* banner;
* blog;
* blog\_category;
* brand;
* bundle\_category;
* buisnes\_settings
* category;
* ci\_sessions;
* contact\_massage;
* coupon;
* currency\_settings;
* customer\_product;
* email\_template;
* general\_settings;
* language;
* language\_list;
* logo;
* membership;
* membership\_payment;
* message;
* message\_thread;
* package;
* package\_payment;
* page;
* permission;
* product;
* product\_affilation;
* product\_affilation\_code\_use;
* product\_affilation\_points,total;
* role;
* sale;
* slider;
* slider\_style;
* slides;
* social\_links;
* stock;
* subscribe;
* sub\_category;
* ticket;
* ticket\_message;
* ui\_settings;
* user;
* user\_rating;
* vendor;
* vendor\_invoice;
* wallet\_load.

Данную схему можно просмотреть в приложении А.

## 1.8 Обоснование выбора вида программного обеспечения

Для создания курсового проекта была выбрана программа Visual Studio Code. Это редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «легкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш, и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией. Разработка рабочего места специалиста по учёту продаж в салоне сотовой связи «MobExpress», в редакторе VS Code, представлена на рисунке 3.

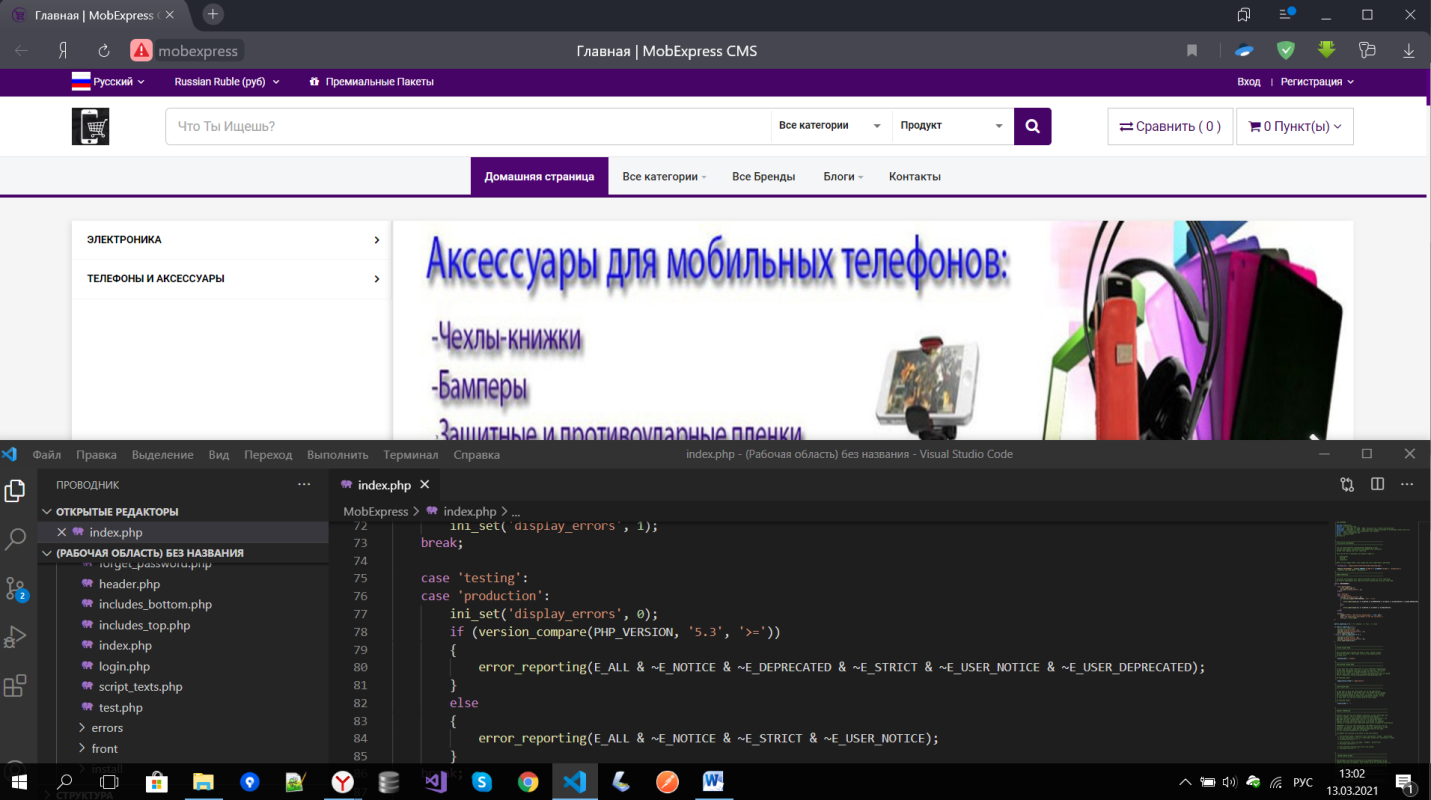


Рисунок 3 – Проектирование сайта

Notepad++: бесплатный редактор исходного кода и замена Блокнота. Notepad++ – это бесплатный (как в "free speech", а также как в "free beer") редактор исходного кода и замена блокнота, который поддерживает несколько языков. Запуск в среде MS Windows, его использование регулируется лицензией GPL; Visual Studio Code: сборка и отладка современных веб-и облачных приложений, от корпорации Майкрософт. Создание и отладка современных веб-и облачных приложений. Код является бесплатным и доступен на вашей любимой платформе - Linux, Mac OSX и Windows.

Notepad++ и код Visual Studio можно в первую очередь классифицировать как средства «текстового редактора».

Visual Studio Code позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ.

В редакторе присутствуют встроенный отладчик, инструменты для работы с Git и средства рефакторинга, навигации по коду, автодополнения типовых конструкций и контекстной подсказки. Продукт поддерживает разработку для платформ ASP.NET и Node.js, и считается легковесным решение, которое позволяет обойтись без полной интегрированной среды разработки. Большим плюсом редактора является поддержка большого количества языков, таких как C++, C#, Python, PHP, JavaScript и других.

Visual Studio Code вышел относительно недавно и уже начал постепенно набирать свою популярность.

Возможности VS Code:

* интеграция настроек из Sublime, Atom. После установки VS Code проверяет наличие других IDE. Если найдена Sublime или Atom VS Code предложит частично перенести настройки и быстрые клавиши в VS Code.
* расширения. Удобная система добавления расширений. Достаточно перейти во вкладку с расширениями и написать в поиске необходимое дополнение.
* Git. Удобная работа с системой контроля версий. Работа с ней такая же, как и в Visual Studio 2019. Отображение коммитов, истории, веток и т.д.
* поиск. Умный поиск с возможностью замены по разным критериям.
* Debug (Запуск приложений). Для веб проектов уже интегрирована отладка. Для других необходимо установить расширение. Отображение использующихся переменных при дебаге. Отображение ошибок (для некоторых языков нужно установить расширение).
* интегрированный терминал. Можно работать через неё с Git и отлаживать приложение.
* Настройка. Большое количество настроек, как и в SublimeText и Atom. Применяются моментально, перезапускать IDE не требуется.
* Визуальное оформление. Можно кардинально изменить вид VS Code. Также есть реализованные темы, которые применяются в один клик.

Таким образом, Visual Studio Code – это самая удобная легковесная среда разработки. Легко расширяемый и настраиваемый интерфейс VS Code является одним из лучших IDE для разработки веб проектов на сегодняшний день.

## 1.9 Физическая модель данных

Таблица 3 – admin (авторизация администратора)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| admin\_id | int(11) | auto\_increment |
| name | longtext | - |
| phone | longtext | - |
| address | longtext | - |
| email | longtext | - |
| password | longtext | - |
| role | varchar(10) | - |
| timestamp | varchar(20) | - |

Таблица 4 – baner (банер)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| banner\_id | int(11) | auto\_increment |
| page | longtext | - |
| place | longtext | - |
| num | longtext | - |
| status | longtext | - |
| link | longtext | - |
| image\_ext | longtext | - |

Таблица 5 – blog (блог)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| blog\_id | int(11) | auto\_increment |
| title | varchar(500) | - |
| summery | varchar(1000) | - |
| autor | varchar(500) | - |
| date | varchar(200) | - |
| description | longtext | - |
| status | varchar(100) | - |
| blog\_category | varchar(25) | - |
| number\_of\_view | int(11) | - |

Таблица 6 – blog\_category (блог категорий)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| blog\_category\_id | int(11) | auto\_increment |
| name | varchar(300) | - |

Таблица 7 – brand (бренды)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| brand\_id | int(11) | auto\_increment |
| name | longtext | - |
| description | longtext | - |
| logo | longtext | - |

Таблица 8 – bundle\_stock (пакет акций)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| stock\_id | int(11) | auto\_increment |
| type | longtext | - |
| product\_bundle | longtext | - |
| quantity | longtext | - |
| rate | longtext | - |
| total | longtext | - |
| reason\_note | longtext | - |
| datetime | longtext | - |
| sale\_id | varchar(30) | - |
| added\_by | varchar(50) | - |

Таблица 9 – business\_settings (бизнес настроки)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| business\_setting\_id | int(11) | auto\_increment |
| type | longtext | - |
| status | varchar(10) | - |
| value | longtext | - |

Таблица 10 – category (категории)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| category\_id | int(11) | auto\_increment |
| category\_name | longtext | - |
| description | longtext | - |
| digital | varchar(10) | - |
| banner | longtext | - |
| data\_brands | longtext | - |
| data\_vendors | longtext | - |
| data subdest | longtext | - |

Таблица 11 – ci\_session (сессия)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| id | varchar(40) | auto\_increment |
| ip\_address | varchar(45) | - |
| timestamp | int(10) | - |
| data | blob | - |

Таблица 12 – contact\_message (контактные сообщения)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Дополнительно |
| contact\_message\_id | int(11) | auto\_increment |
| name | varchar(100) | - |
| subject | varchar(1000) | - |
| email | varchar(200) | - |
| message | longtext | - |
| timestamp | varchar(20) | - |
| view | varchar(10) | - |
| reply | longtext | - |
| other | longtext | - |

Все таблицы созданы с помощью языка структурированных запросов (SQL). Также определённые таблицы связаны реляционными отношениями. Информация из таблиц извлекается с помощью SQL и обрабатывается с помощью интерпретатора PHP. После обработки на клиент передается контент в виде HTML страницы содержащий в себе необходимую информацию.

## 1.10 Анализ рынка

Рынок интернет-торговли в России не так развит, как в странах Западной Европы или Северной Америки. В ходе исследования крупнейших интернет-магазинов России были получены следующие результаты:

1. 80% респондентов когда-либо совершали покупку в интернет-магазине;
2. Самой популярной формой оплаты до сих пор остается оплата наличными при доставке товара, хотя все шире применяется и такая форма оплаты, как электронные платежи;
3. Неуверенность в надежности продавца и качестве услуг по доставке товара — основные факторы, удерживающие потенциальных клиентов интернет-магазинов от покупок в Сети;
4. Социальные медиа оказывают значительное влияние на решение интернет-пользователей о совершении той или иной покупки. Члены социальной сети гораздо больше доверяют мнению других ее участников о продуктах и производителях, чем рекламе.

При изучении структуры спроса уже по традиции обращаются к профильному анализу рынка электроники, ежегодно проводимому маркетологами «М-Видео». Самой активной остается продажа смартфонов, сопутствующих аксессуаров и наушников. Если на протяжении 2013-2016 годов этот сегмент характеризовался некоторой стагнацией, то уже в 2017 году продажи выросли сразу на 2 миллиона штук. Увеличились средний чек товара и размер экрана устройств.

На вторых позициях ритейла отмечается компьютерная техника. Спрос поддерживается частным и корпоративным потребителем. В этом сегменте намечается переход к более удобным форматам техники. Некогда популярные десктопы сменились ноутбуками, а по итогам продаж 2018 потребитель «переключился» на планшеты.

Рынок продажи телевизоров как жизненно необходимый вид техники ожидает мощный катализатор роста. Переход в 2019 году на цифровой формат вещания телевизионных каналов с высокой вероятностью поднимет продажи техники нового поколения. Вполне ожидаемо, что под спрос увеличится и средняя цена продажи. Если в 2017-2018 годах эта цифра изменится с 25 на 35 тысяч рублей.

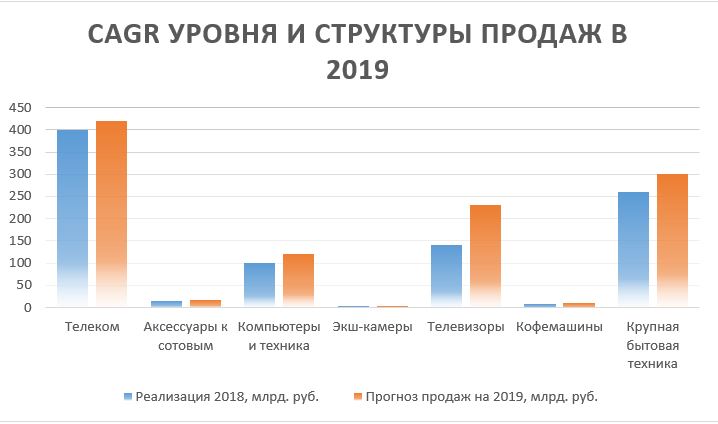


Рисунок 4 – Структура продаж

В анализе рынка электроники на 2019 не приводится прогнозов относительно серьезной реструктуризации отрасли. Лидерами остаются «М-видео», DNS и Эльдорадо. В регионах наблюдается серьезная конкуренция, нередко переходящая на сетевой уровень. Повышение налога на добавленную стоимость и ужесточение требований в сегменте онлайн-продаж как накладные расходы будут учтены в ценообразовании. За все придется платить конечному потребителю. Серьезные преобразования в системе доставки товаров с зарубежных площадок увеличат емкость рынка до 30 процентов. Освобождающаяся ниша рынка будет замещаться уже имеющимся ассортиментом, включая продукцию отечественных производителей. Анализ рынка бытовой техники и электроники поднимает ценовые планки до 5 процентов по сравнению с предыдущим периодом. Именно такие прогнозы ставит портал Hi-Tech Mail.ru.

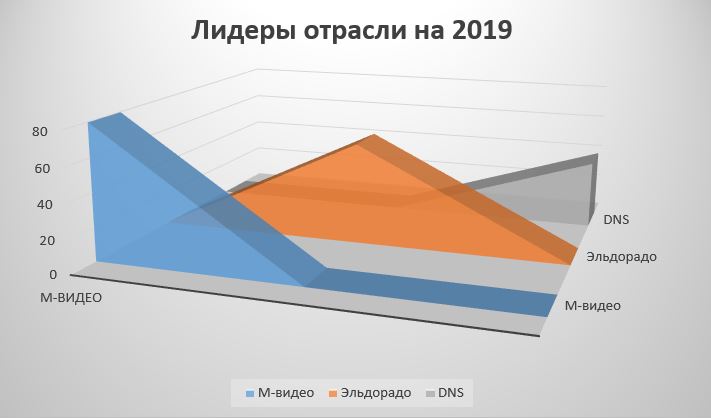


Рисунок 5 – Лидеры продаж

Для определения актуальности, функционала, работоспособности и удобства интернет-магазином «MobExpress», разработанном в данном курсовом проекте, были выбраны следующие сайты интернет-магазинов для анализа:

1. <https://www.dns-shop.ru>
2. https://www.eldorado.ru

Первым шагом для сравнения данных сайтов, является информационная разработка веб-сайтов и разработанного продукта.

На данном шаге, анализ проводился по следующим критериям – функционал сайта.

Таблица 12 – Сравнение web-страниц

|  |  |  |  |
| --- | --- | --- | --- |
|  | [dns-shop.ru](https://www.dns-shop.ru) | eldorado.ru | spravka.000webhostapp.com/index.php |
| Разнообразный ассортимент товаров | + | + | + |
| Наличие комментариев о товаре | + | + | + |
| Возможность рассказать о интернет-магазине в социальных сетях | + | + | + |
| Наличие панели «Помощь» | + | + | + |
| Реклама | + | + | + |

Следующим шагом анализа является структура и юзабилити интернет-магазина. Это самая важная функция сайта, о которой многие разработчики сайтов. При отсутствии данной функции, пользователь может долго не задержаться на сайте, если не будет простого интерфейса с удобством поиска товара. Тем самым, магазин понесёт огромные убытки.

Таблица 13 – Анализ структуры сайтов

|  |  |  |  |
| --- | --- | --- | --- |
|  | [dns-shop.ru](https://www.dns-shop.ru) | eldorado.ru | spravka.000webhostapp.com/index.php |
| Наличие навигации | + | + | + |
| Наличие обратной связи | + | + | + |
| Онлайн-консультант | - | + | + |
| Поиск по сайту | + | + | + |
| Адаптивность web-дизайна | + | - | - |
| Быстрый переход по сайту | + | + | - |

# 2 Практический раздел

## 2.1 Требования к информационной и программной совместимости

Интернет-магазин «MobExpress» был разработан в текстовом редакторе Visual Studio Code на скриптовом языке программирования php.

PHP – cкриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Технические характеристики персонального компьютера, на котором разрабатывался сайт:

* процессор: Intel(R) Core(TM) i3-7100U CPU @2.40GHz 2.40GHz
* оперативная память: 4,00 ГБ
* дисковое пространство: 1 ТБ
* разрешение монитора: 1920×1080
* клавиатура
* мышь

Необходимые программные средства, необходимые при эксплуатации программы:

* операционная система: Windows XP/7/Vista/8, 8.1/10
* браузер с поддержкой JavaScript

Технические средства для использования интернет-магазина MobExpress:

* тактовая частота процессора 1,6 ГГц и выше
* ОЗУ 1 ГБ
* 10 МБ свободного дискового пространства
* видеоадаптер, совместимый с Direct 9 и поддерживающий разрешение экрана 1920×1080 точек и выше
* клавиатура
* мышь

## 2.2 Разработка пользовательского интерфейса

После того, как была выбрана ссылка на сайт, появляется главная страница магазина. На рисунке 5 представлен весь ассортимент товара, который присутствует в интернет-магазине.

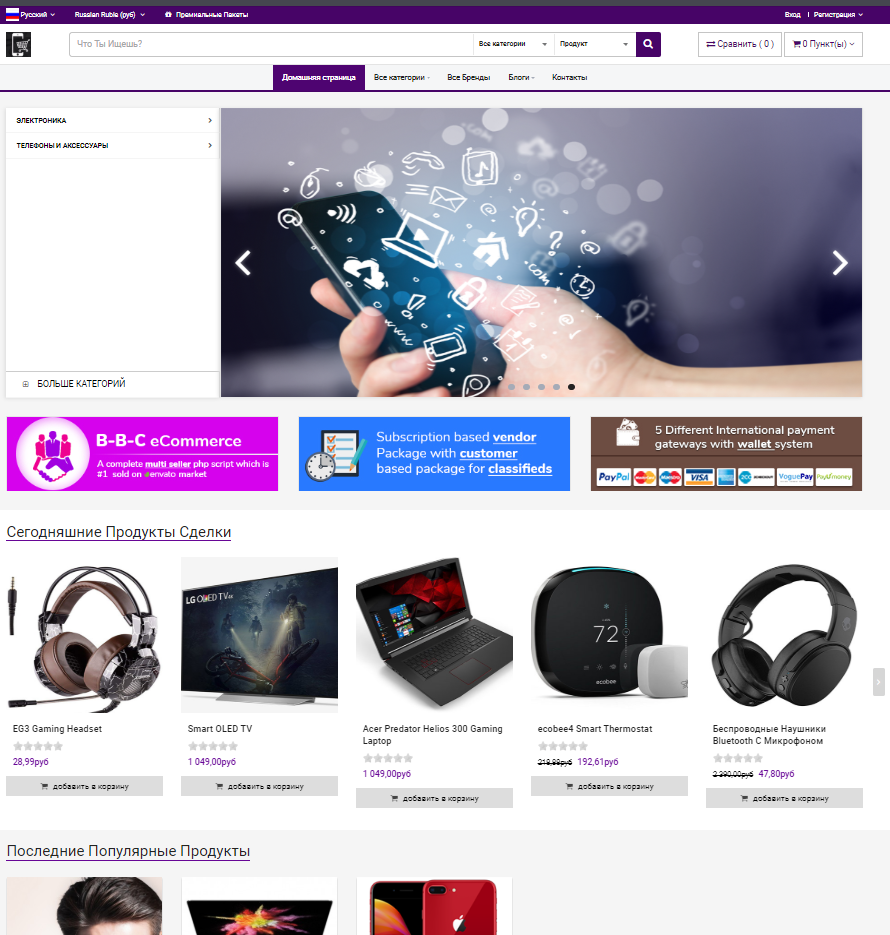


Рисунок 5 – Стартовая страница интернет-магазина

Также, на данном сайте, можно выбрать любой товар из данных категорий. Для этого, как показано на рисунке 6, нужно нажать на данную категорию товара и выбрать понравившуюся модель.

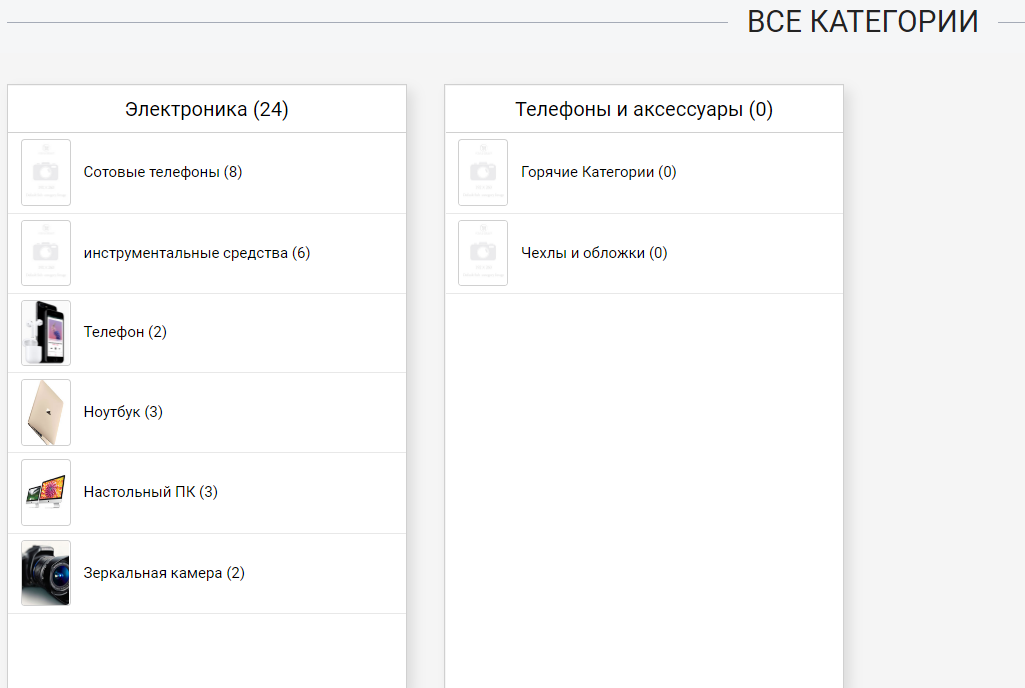


Рисунок 6 – Категории товаров

Ещё один способ найти товар – найти его по брендам. На рисунке 7 показаны доступные бренды по каждой из категорий товара.

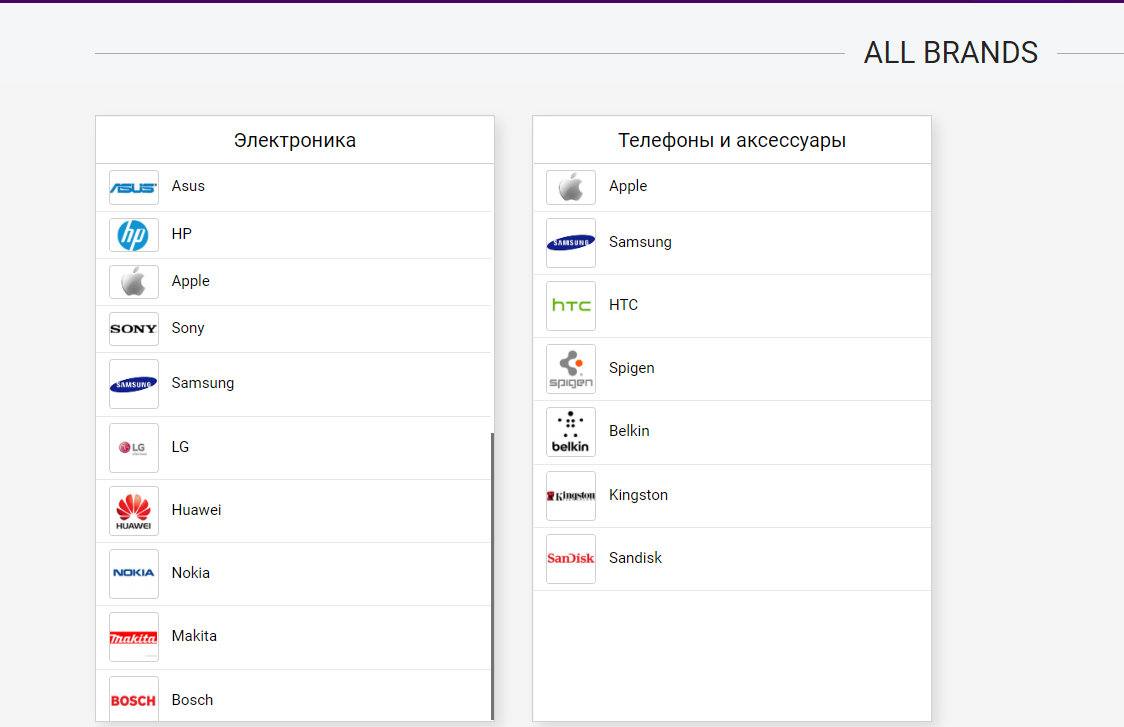


Рисунок 7 – Поиск по параметрам

Перед покупкой товара, нужно посмотреть его описание, характеристики и отзывы. На рисунке 8 показано какие характеристики товара видит пользователь, а также сколько было просмотров и отзывов данного товара. Также он может положить товар в корзину, не заходя в более подробное описание товара.

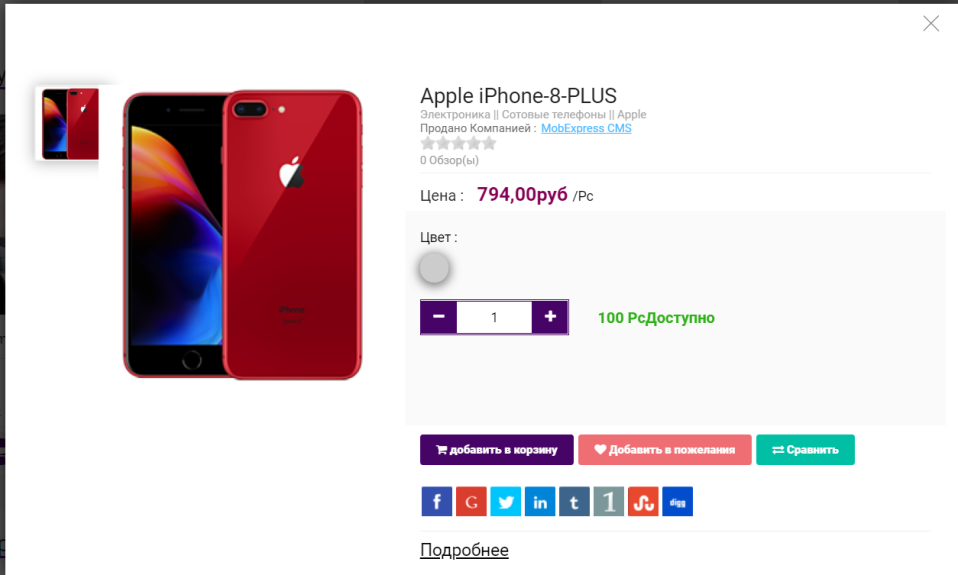


Рисунок 8 – Краткое описание товара

После того, как пользователь нажал на определённый товар, он попадает на страницу с полным описанием и характеристиками товара. Как показано на рисунке 9, на данной странице можно найти описание товара, его характеристики, а также отзывы от других клиентов, которые уже приобрели данный товар.

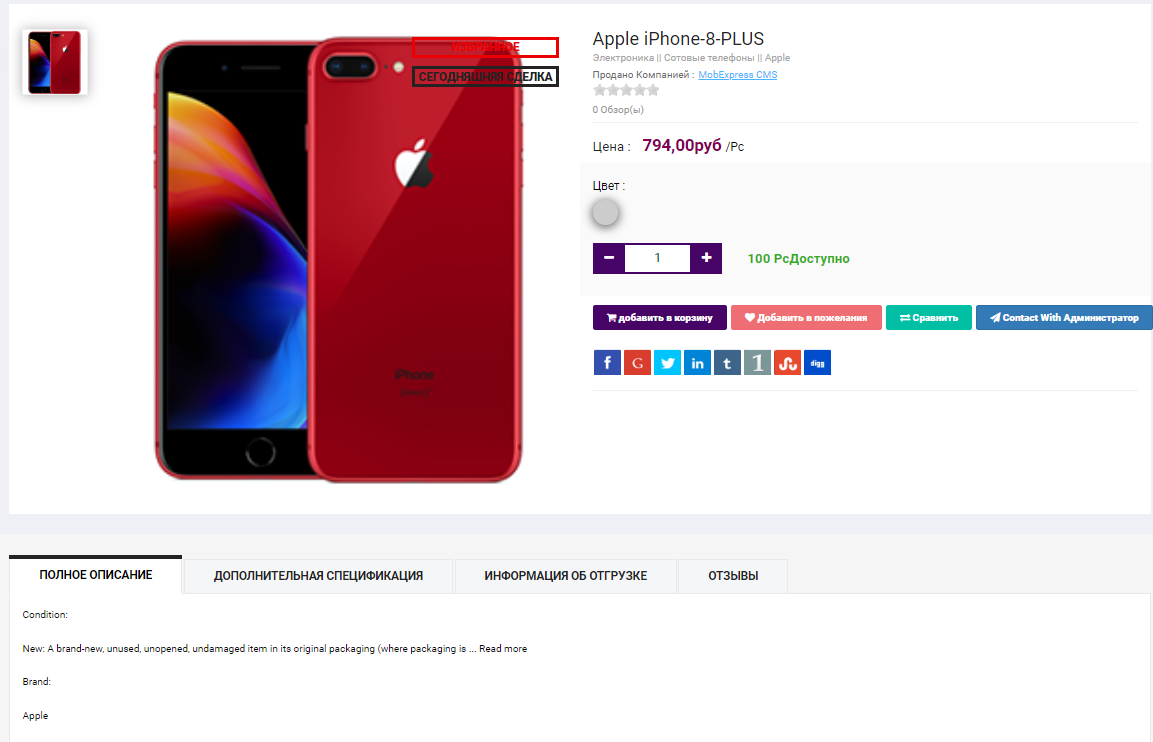


Рисунок 9 – Информация по товару

## 2.3 Тестирование программы

Когда товар выбран, и клиент доволен его описанием, характеристиками и отзывами, настаёт момент покупать товар. Для того чтобы купить товар, необходимо нажать на кнопку «Купить», и он автоматически попадает в корзину. После того, как клиент войдёт в корзину, он увидит свои товары. Также он может увеличить или уменьшить количество, покупаемого им товара, нажав на соответствующие знаки («+» – прибавить товар, а «-» – уменьшить товар). На рисунке 10 показано, что клиент видит при входе в корзину. Да данном шаге, также можно ввести купон на скидку, если у клиента такой имеется. Для того, чтобы продолжить клиент нажимает на кнопку «Следующий».

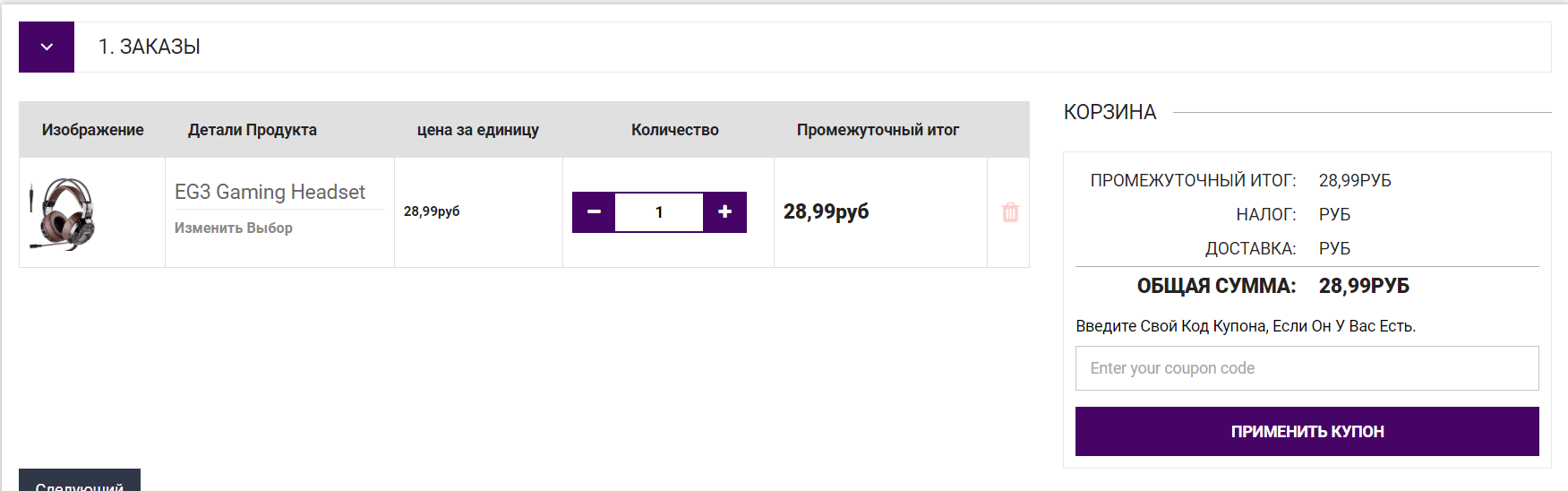


Рисунок 10 – Корзина товаров

На рисунке 11 показан следующий шаг при покупке товара – «Контактная информация». Здесь пользователь вводит свои данные, куда присылать данный товар. Для завершения заполнения формы для заказа, нужно нажать на кнопку «Следующий».

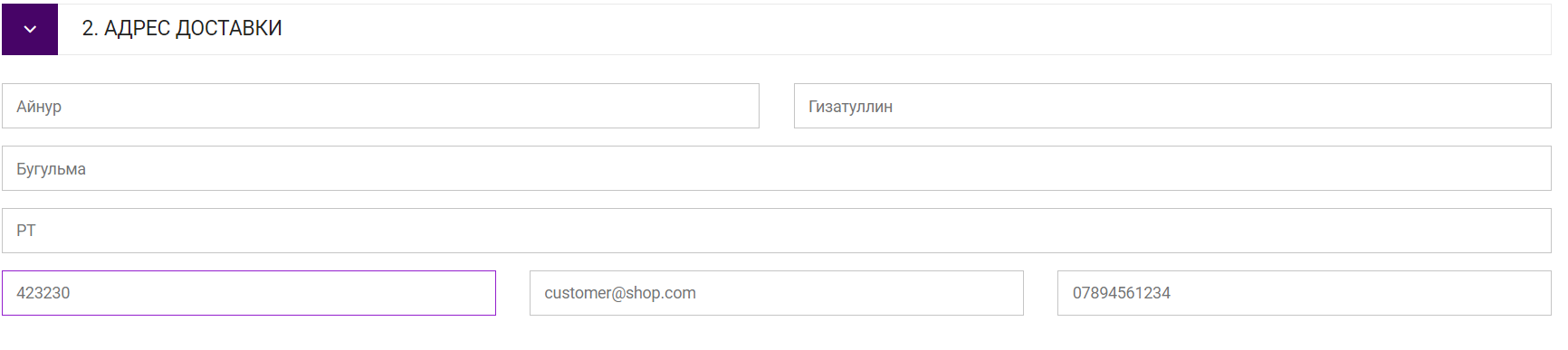


Рисунок 11 – Контактная информация

Рисунок 12 является последним завершающим шагом при покупке товара. На этом шаге данный товар попадает на модерацию администратору, а пользователю остаётся только выбрать вариант оплаты товара, и оплатить его. После того, как оплата была произведена, в статусе товара у администратора, будет помечено, что товар оплачен.

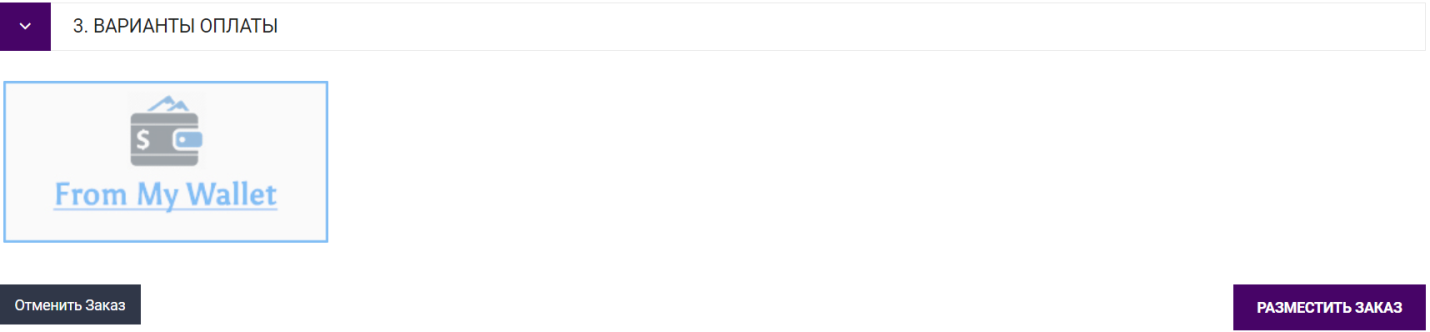


Рисунок 12 – Оплата товара

## 2.4 Анализ надежности и качества

Перед тем, как купить товар, пользователь должен авторизоваться в системе. Если он не зарегистрирован, нужно клиенту зарегистрироваться, перейдя по соответствующей ссылке.

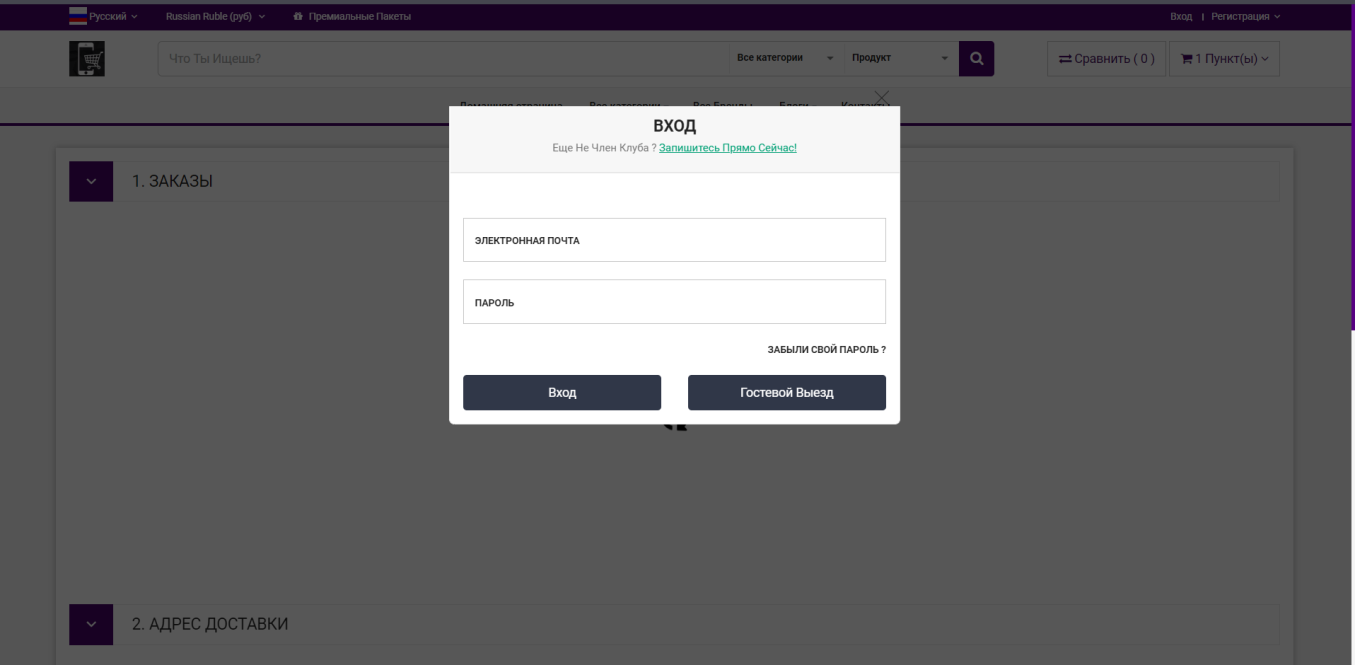


Рисунок 13 – Авторизация при покупке

При регистрации пользователя, ему нужно заполнить нужные поля. Если они будут пусты, то появятся предупреждения о заполнении полей. Для регистрации указывать фамилию не обязательно, если пользователь его не имеет.

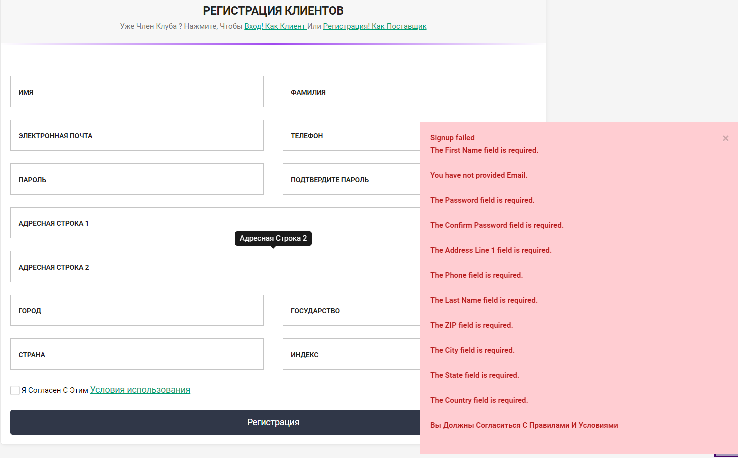


Рисунок 14 – Регистрация пользователя

При успешной регистрации пользователю будет выведено сообщение об успешной его регистрации. А также пользователь будет автоматически перенаправлен на страницу входа.

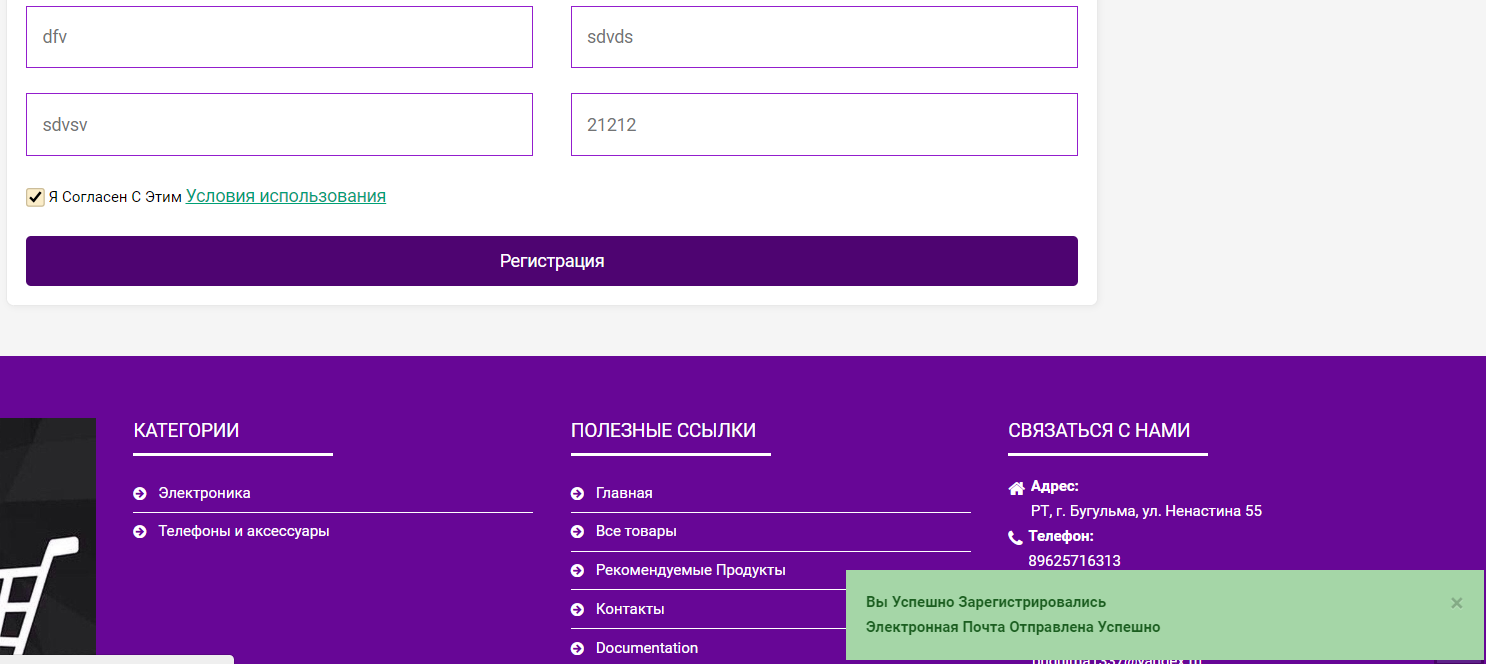


Рисунок 15 – Успешная регистрация пользователя

# Заключение

Результатом проделанной работы стал готовое к работе автоматизированное рабочее место по учёту продаж в салоне сотовой связи. В ходе проведения работы были выделены основные задачи сайта, важные особенности целевой аудитории; проанализированы различные пути решения и выбраны наиболее подходящие в данной ситуации. Оценивая проделанную работу можно сделать выводы:

* сайт предоставляет пользователям из целевой группы наиболее нужные им сервисы, а именно – возможность ознакомиться со списком продуктов и высказать пожелания администрации сайта;
* дизайн сайта соответствует предполагаемым предпочтениям целевой группы, времени и целям нахождения потенциальных клиентов на сайте;
* реализован сервис поиска по сайту, что считается комфортным и даже необходимым дополнением;
* реализована навигация с помощью меню по страницам сайта. Ссылки легко отличимы от основного текста, находятся в одном месте;
* реализована сортировка продукта, что помогает различным категориям целевой аудитории легко ориентироваться в товарах;
* уделено особое внимание безопасности пользователей, контролю над корректностью входных данных, защита от переполнения базы данных;
* реализован интерфейс администратора, что позволяет работать с сайтом малоквалифицированным работникам и совершать меньше ошибок при работе с базой данных.

Работа была выполнена успешно за исключением небольших недоработок, устранение которых было бы полезно для продвижения сайта. Среди них можно указать:

* отсутствие форума, чата, других средств активного общения;
* плохо работает геолокация;
* отсутствие прослеживания нахождения товара.

При должном продвижении, сайт мог бы пользоваться успехом у определенной категории пользователей. Для этого необходимо перевести его на более мощный веб сервер и более вместительную и оперативную базу данных. Также нужно продумать рекламные акции, способ оповещения потенциальных клиентов и другие коммерческие аспекты.

Создавая рабочее место по учёту продаж в салоне сотовой связи «MobExpress», были приобретены практические навыки обследования предметной области, логической и физической разработки, проанализированы функциональные возможности интернет-магазинов, отдельные компоненты применены на практике.

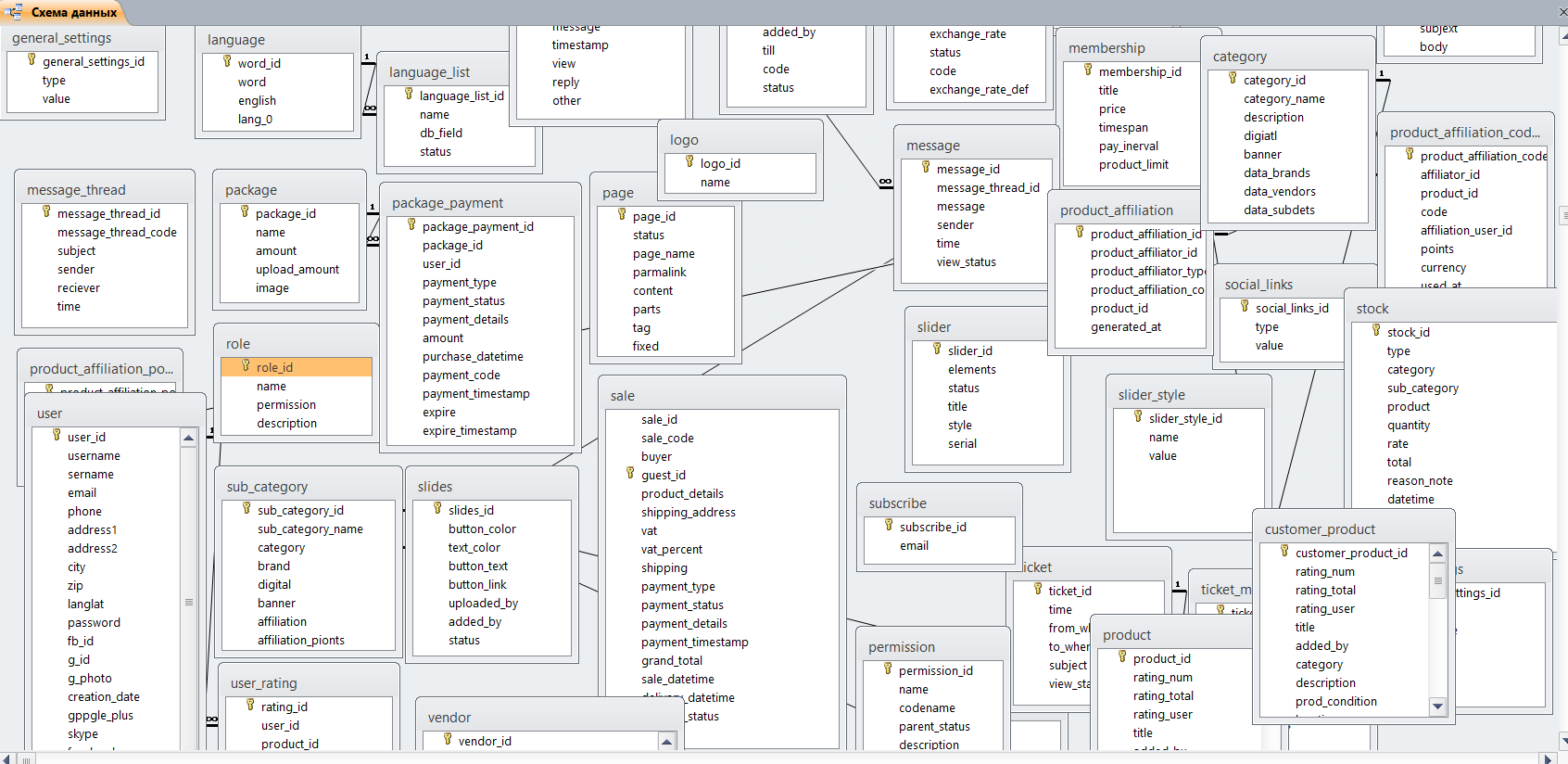
Подводя итог работы, можно сделать вывод, что данное рабочее место по учёту продаж в салоне сотовой связи является актуальным в нынешнее время тем, что не требует арендной платы за помещение и большого штата сотрудников. В данном проекте был реализован интуитивно-понятный пользовательский интерфейс, а также выложены товары, имеющие цены ниже рыночной стоимости.

# Список использованный источников

1. Акила К., Еремеевский А. Интернет-магазин с нуля. Полное пошаговое руководство; Питер - М., 2017. - **713** c.
2. Аткинсон, Л. MySQL. Библиотека профессионала / Л. Аткинсон. — М.: Диалектика, 2016. — 624 с.
3. Деловой Интернет; САРКОН - М., 2016. - 256 c.
4. Дунаев, В. Сценарии для Web-сайта. PHP и JavaScript / В. Дунаев. — СПб.: БХВ-Петербург, 2016. — 576 с.
5. Лебедев А. Руководство Лебедев А. - Москва.: Издательство Студии Артемия Лебедева, 2017. - 452 с.
6. Ратшиллер Т., Геркен Т. PHP4: разработка Web-приложений. - СПб: Питер, 2017. - 384 с.
7. Спейнауэр С. Справочник Web-мастера. СПб.: BHV, 2017. - 368 с.
8. Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySQL. - К.: "ДиаСофт", 2016. - 672 с
9. Ульман Ларри PHP и MySQL. Cоздание интернет-магазинов; Вильямс - М., 2017. - 544 c.

# 

# Приложение А



# Приложение Б

Главная страница – index.php

<?php

/\*\*

 \* CodeIgniter

 \*

 \* An open source application development framework for PHP

 \*

 \* This content is released under the MIT License (MIT)

 \*

 \* Copyright (c) 2014 - 2015, British Columbia Institute of Technology

 \*

 \* Permission is hereby granted, free of charge, to any person obtaining a copy

 \* of this software and associated documentation files (the "Software"), to deal

 \* in the Software without restriction, including without limitation the rights

 \* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

 \* copies of the Software, and to permit persons to whom the Software is

 \* furnished to do so, subject to the following conditions:

 \*

 \* The above copyright notice and this permission notice shall be included in

 \* all copies or substantial portions of the Software.

 \*

 \* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

 \* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

 \* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

 \* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

 \* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

 \* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

 \* THE SOFTWARE.

 \*

 \* @package CodeIgniter

 \* @author  EllisLab Dev Team

 \* @copyright   Copyright (c) 2008 - 2014, EllisLab, Inc. (http://ellislab.com/)

 \* @copyright   Copyright (c) 2014 - 2015, British Columbia Institute of Technology (http://bcit.ca/)

 \* @license http://opensource.org/licenses/MIT  MIT License

 \* @link    http://codeigniter.com

 \* @since   Version 1.0.0

 \* @filesource

 \*/

/\*

 \*---------------------------------------------------------------

 \* APPLICATION ENVIRONMENT

 \*---------------------------------------------------------------

 \*

 \* You can load different configurations depending on your

 \* current environment. Setting the environment also influences

 \* things like logging and error reporting.

 \*

 \* This can be set to anything, but default usage is:

 \*

 \*     development

 \*     testing

 \*     production

 \*

 \* NOTE: If you change these, also change the error\_reporting() code below

 \*/

    include\_once './application/libraries/vendor/autoload.php';

    define('ENVIRONMENT', isset($\_SERVER['CI\_ENV']) ? $\_SERVER['CI\_ENV'] : 'production');

    //define('CFG\_TIME\_ZONE', 'Asia/Dacca');

/\*

 \*---------------------------------------------------------------

 \* ERROR REPORTING

 \*---------------------------------------------------------------

 \*

 \* Different environments will require different levels of error reporting.

 \* By default development will show errors but testing and live will hide them.

 \*/

switch (ENVIRONMENT)

{

    case 'development':

        error\_reporting(-1);

        ini\_set('display\_errors', 1);

    break;

    case 'testing':

    case 'production':

        ini\_set('display\_errors', 0);

        if (version\_compare(PHP\_VERSION, '5.3', '>='))

        {

            error\_reporting(E\_ALL & ~E\_NOTICE & ~E\_DEPRECATED & ~E\_STRICT & ~E\_USER\_NOTICE & ~E\_USER\_DEPRECATED);

        }

        else

        {

            error\_reporting(E\_ALL & ~E\_NOTICE & ~E\_STRICT & ~E\_USER\_NOTICE);

        }

    break;

    default:

        header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);

        echo 'The application environment is not set correctly.';

        exit(1); // EXIT\_ERROR

}

$error\_reporting = 0; // 0 = unknown, 1 = full , 2 = none

if ($error\_reporting == 1) {

    /\*full error reporting\*/

    ini\_set('display\_errors', 1);

    ini\_set('display\_startup\_errors', 1);

    error\_reporting(E\_ALL);

} else if ($error\_reporting == 2) {

    /\*no error reporting\*/

    ini\_set('display\_errors', 0);

    ini\_set('display\_startup\_errors', 0);

    error\_reporting(~E\_ALL);

}

/\*

 \*---------------------------------------------------------------

 \* SYSTEM FOLDER NAME

 \*---------------------------------------------------------------

 \*

 \* This variable must contain the name of your "system" folder.

 \* Include the path if the folder is not in the same directory

 \* as this file.

 \*/

    $system\_path = 'system';

/\*

 \*---------------------------------------------------------------

 \* APPLICATION FOLDER NAME

 \*---------------------------------------------------------------

 \*

 \* If you want this front controller to use a different "application"

 \* folder than the default one you can set its name here. The folder

 \* can also be renamed or relocated anywhere on your server. If

 \* you do, use a full server path. For more info please see the user guide:

 \* http://codeigniter.com/user\_guide/general/managing\_apps.html

 \*

 \* NO TRAILING SLASH!

 \*/

    $application\_folder = 'application';

/\*

 \*---------------------------------------------------------------

 \* VIEW FOLDER NAME

 \*---------------------------------------------------------------

 \*

 \* If you want to move the view folder out of the application

 \* folder set the path to the folder here. The folder can be renamed

 \* and relocated anywhere on your server. If blank, it will default

 \* to the standard location inside your application folder. If you

 \* do move this, use the full server path to this folder.

 \*

 \* NO TRAILING SLASH!

 \*/

    $view\_folder = '';

/\*

 \* --------------------------------------------------------------------

 \* DEFAULT CONTROLLER

 \* --------------------------------------------------------------------

 \*

 \* Normally you will set your default controller in the routes.php file.

 \* You can, however, force a custom routing by hard-coding a

 \* specific controller class/function here. For most applications, you

 \* WILL NOT set your routing here, but it's an option for those

 \* special instances where you might want to override the standard

 \* routing in a specific front controller that shares a common CI installation.

 \*

 \* IMPORTANT: If you set the routing here, NO OTHER controller will be

 \* callable. In essence, this preference limits your application to ONE

 \* specific controller. Leave the function name blank if you need

 \* to call functions dynamically via the URI.

 \*

 \* Un-comment the $routing array below to use this feature

 \*/

    // The directory name, relative to the "controllers" folder.  Leave blank

    // if your controller is not in a sub-folder within the "controllers" folder

    // $routing['directory'] = '';

    // The controller class file name.  Example:  mycontroller

    // $routing['controller'] = '';

    // The controller function you wish to be called.

    // $routing['function'] = '';

/\*

 \* -------------------------------------------------------------------

 \*  CUSTOM CONFIG VALUES

 \* -------------------------------------------------------------------

 \*

 \* The $assign\_to\_config array below will be passed dynamically to the

 \* config class when initialized. This allows you to set custom config

 \* items or override any default config values found in the config.php file.

 \* This can be handy as it permits you to share one application between

 \* multiple front controller files, with each file containing different

 \* config values.

 \*

 \* Un-comment the $assign\_to\_config array below to use this feature

 \*/

    // $assign\_to\_config['name\_of\_config\_item'] = 'value of config item';

// --------------------------------------------------------------------

// END OF USER CONFIGURABLE SETTINGS.  DO NOT EDIT BELOW THIS LINE

// --------------------------------------------------------------------

/\*

 \* ---------------------------------------------------------------

 \*  Resolve the system path for increased reliability

 \* ---------------------------------------------------------------

 \*/

    // Set the current directory correctly for CLI requests

    if (defined('STDIN'))

    {

        chdir(dirname(\_\_FILE\_\_));

    }

    if (($\_temp = realpath($system\_path)) !== FALSE)

    {

        $system\_path = $\_temp.'/';

    }

    else

    {

        // Ensure there's a trailing slash

        $system\_path = rtrim($system\_path, '/').'/';

    }

    // Is the system path correct?

    if ( ! is\_dir($system\_path))

    {

        header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);

        echo 'Your system folder path does not appear to be set correctly. Please open the following file and correct this: '.pathinfo(\_\_FILE\_\_, PATHINFO\_BASENAME);

        exit(3); // EXIT\_CONFIG

    }

/\*

 \* -------------------------------------------------------------------

 \*  Now that we know the path, set the main path constants

 \* -------------------------------------------------------------------

 \*/

    // The name of THIS file

    define('SELF', pathinfo(\_\_FILE\_\_, PATHINFO\_BASENAME));

    // Path to the system folder

    define('BASEPATH', str\_replace('\\', '/', $system\_path));

    // Path to the front controller (this file)

    define('FCPATH', dirname(\_\_FILE\_\_).'/');

    // Name of the "system folder"

    define('SYSDIR', trim(strrchr(trim(BASEPATH, '/'), '/'), '/'));

    // The path to the "application" folder

    if (is\_dir($application\_folder))

    {

        if (($\_temp = realpath($application\_folder)) !== FALSE)

        {

            $application\_folder = $\_temp;

        }

        define('APPPATH', $application\_folder.DIRECTORY\_SEPARATOR);

    }

    else

    {

        if ( ! is\_dir(BASEPATH.$application\_folder.DIRECTORY\_SEPARATOR))

        {

            header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);

            echo 'Your application folder path does not appear to be set correctly. Please open the following file and correct this: '.SELF;

            exit(3); // EXIT\_CONFIG

        }

        define('APPPATH', BASEPATH.$application\_folder.DIRECTORY\_SEPARATOR);

    }

    // The path to the "views" folder

    if ( ! is\_dir($view\_folder))

    {

        if ( ! empty($view\_folder) && is\_dir(APPPATH.$view\_folder.DIRECTORY\_SEPARATOR))

        {

            $view\_folder = APPPATH.$view\_folder;

        }

        elseif ( ! is\_dir(APPPATH.'views'.DIRECTORY\_SEPARATOR))

        {

            header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);

            echo 'Your view folder path does not appear to be set correctly. Please open the following file and correct this: '.SELF;

            exit(3); // EXIT\_CONFIG

        }

        else

        {

            $view\_folder = APPPATH.'views';

        }

    }

    if (($\_temp = realpath($view\_folder)) !== FALSE)

    {

        $view\_folder = $\_temp.DIRECTORY\_SEPARATOR;

    }

    else

    {

        $view\_folder = rtrim($view\_folder, '/\\').DIRECTORY\_SEPARATOR;

    }

    define('VIEWPATH', $view\_folder);

    /\*custom defines\*/

    define('KB', 1024);

    define('MB', 1048576);

    define('GB', 1073741824);

    define('TB', 1099511627776);

/\*

 \* --------------------------------------------------------------------

 \* LOAD THE BOOTSTRAP FILE

 \* --------------------------------------------------------------------

 \*

 \* And away we go...

 \*/

require\_once BASEPATH.'core/CodeIgniter.php';

Регистрация – resister/index.php

<section class="page-section color get\_into">

    <div class="container">

        <div class="row margin-top-0">

            <div class="col-sm-8 col-sm-offset-2">

                <div class="logo\_top">

                    <a href="<?php echo base\_url()?>">

                        <img class="img-responsive" src="<?php echo $this->crud\_model->logo('home\_bottom\_logo'); ?>" alt="Shop" style="z-index:200">

                    </a>

                </div>

                        <?php

                    echo form\_open(base\_url() . 'home/registration/add\_info/', array(

                        'class' => 'form-login',

                        'method' => 'post',

                        'id' => 'sign\_form'

                    ));

                    $fb\_login\_set = $this->crud\_model->get\_type\_name\_by\_id('general\_settings','51','value');

                    $g\_login\_set = $this->crud\_model->get\_type\_name\_by\_id('general\_settings','52','value');

                ?>

                <div class="row box\_shape">

                  <div class="title">

                      <?php echo translate('customer\_registration');?>

                        <div class="option">

                        <?php echo translate('already\_a\_member\_?\_click\_to\_');?>

                        <?php

                                 if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','58','value') !== 'ok') { ?>

                              <a href="<?php echo base\_url(); ?>home/login\_set/login">

                                  <?php echo translate('login');?>!

                              </a>

                        <?php

                                             }

                           else { ?>

                                <a href="<?php echo base\_url(); ?>home/login\_set/login">

                                    <?php echo translate('login');?>! <?php echo translate('as\_customer');?>

                                </a>

                              <?php echo translate('\_or\_');?>

                                <a href="<?php echo base\_url(); ?>home/vendor\_logup/registration">

                                    <?php echo translate('sign\_up');?>! <?php echo translate('as\_vendor');?>

                                </a>

                              <?php

                                            }

                                        ?>

                        </div>

                      </div>

                      <hr>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control required" name="username" type="text" placeholder="<?php echo translate('first\_name');?>" data-toggle="tooltip" title="<?php echo translate('first\_name');?>">

                          </div>

                      </div>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control required" name="surname" type="text" placeholder="<?php echo translate('last\_name');?>" data-toggle="tooltip" title="<?php echo translate('last\_name');?>">

                          </div>

                      </div>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control emails required" name="email" type="email" placeholder="<?php echo translate('email');?>" data-toggle="tooltip" title="<?php echo translate('email');?>">

                          </div>

                          <div id='email\_note'></div>

                      </div>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control" name="phone" type="text" placeholder="<?php echo translate('phone');?>" data-toggle="tooltip" title="<?php echo translate('phone');?>">

                          </div>

                      </div>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control pass1 required" type="password" name="password1" placeholder="<?php echo translate('password');?>" data-toggle="tooltip" title="<?php echo translate('password');?>">

                          </div>

                      </div>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control pass2 required" type="password" name="password2" placeholder="<?php echo translate('confirm\_password');?>" data-toggle="tooltip" title="<?php echo translate('confirm\_password');?>">

                          </div>

                          <div id='pass\_note'></div>

                      </div>

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

                          <div class="form-group">

                              <input class="form-control required" name="address1" type="text" placeholder="<?php echo translate('address\_line\_1');?>" data-toggle="tooltip" title="<?php echo translate('address\_line\_1');?>">

                          </div>

                      </div>

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

                          <div class="form-group">

                              <input class="form-control required" name="address2" type="text" placeholder="<?php echo translate('address\_line\_2');?>" data-toggle="tooltip" title="<?php echo translate('address\_line\_2');?>">

                          </div>

                      </div>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control required" type="text" name="city" placeholder="<?php echo translate('city');?>" data-toggle="tooltip" title="<?php echo translate('city');?>">

                          </div>

                      </div>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control required" type="text" name="state" placeholder="<?php echo translate('state');?>" data-toggle="tooltip" title="<?php echo translate('state');?>">

                          </div>

                      </div>

                      <div class="col-md-6">

                          <div class="form-group">

                              <input class="form-control required" type="text" name="country" placeholder="<?php echo translate('country');?>" data-toggle="tooltip" title="<?php echo translate('country');?>">

                          </div>

                      </div>

                      <div class="col-md-6">

                          <div class="">

                              <input class="form-control required" name="zip" type="text" placeholder="<?php echo translate('zip');?>" data-toggle="tooltip" title="<?php echo translate('zip');?>">

                          </div>

                      </div>

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

                          <input  name="terms\_check" type="checkbox" value="ok" >

                          <?php echo translate('i\_agree\_with');?>

                          <a href="<?php echo base\_url();?>home/legal/terms\_conditions" target="\_blank">

                              <?php echo translate('terms\_&\_conditions');?>

                          </a>

                      </div>

                      <?php

                                    if($this->crud\_model->get\_settings\_value('general\_settings','captcha\_status','value') == 'ok'){ ?>

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

                              <div class="outer required">

                                  <div class="form-group">

                                      <?php echo $recaptcha\_html; ?>

                                  </div>

                              </div>

                          </div>

                        <?php

                                      }

                                    ?>

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

                            <span class="btn btn-theme-sm btn-block btn-theme-dark pull-right logup\_btn" data-ing='<?php echo translate('registering..'); ?>' data-msg="">

                                <?php echo translate('register');?>

                            </span>

                        </div>

                        <!--- Facebook and google login -->

                        <?php

                          if($fb\_login\_set == 'ok' || $g\_login\_set == 'ok'){ ?>

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

                                <h2 class="login\_divider"><span><?php echo translate('sign\_in\_with\_facebook');?>or</span></h2>

                            </div>

                            <?php if($fb\_login\_set == 'ok'){ ?>

                                <div class="col-md-12 col-lg-6 <?php if($g\_login\_set !== 'ok'){ ?>mr\_log<?php } ?>">

                                    <?php if (@$user): ?>

                                        <a class="btn btn-theme btn-block btn-icon-left facebook" href="<?= $url ?>">

                                            <i class="fa fa-facebook"></i>

                                            <?php echo translate('sign\_in\_with\_facebook');?>

                                        </a>

                                    <?php else: ?>

                                        <a class="btn btn-theme btn-block btn-icon-left facebook" href="<?= $url ?>">

                                            <i class="fa fa-facebook"></i>

                                            <?php echo translate('sign\_in\_with\_facebook');?>

                                        </a>

                                    <?php endif; ?>

                                </div>

                            <?php }

                            if($g\_login\_set == 'ok'){ ?>

                                <div class="col-md-12 col-lg-6 <?php if($fb\_login\_set !== 'ok'){ ?>mr\_log<?php } ?>">

                                    <?php if (@$g\_user): ?>

                                        <a class="btn btn-theme btn-block btn-icon-left google" style="background:#ce3e26" href="<?= $g\_url ?>">

                                            <i class="fa fa-google"></i>

                                            <?php echo translate('sign\_in\_with\_google');?>

                                        </a>

                                   <?php else: ?>

                                        <a class="btn btn-theme btn-block btn-icon-left google" style="background:#ce3e26" href="<?= $g\_url ?>">

                                            <i class="fa fa-google"></i>

                                            <?php echo translate('sign\_in\_with\_google');?>

                                        </a>

                                    <?php endif; ?>

                                </div>

                            <?php

                            }

                          }

                        ?>

                    </div>

                </form>

            </div>

        </div>

    </div>

</section>

<style>

    .get\_into .terms a{

        margin:5px auto;

        font-size: 14px;

        line-height: 24px;

        font-weight: 400;

        color: #00a075;

        cursor:pointer;

        text-decoration:underline;

    }

    .get\_into .terms input[type=checkbox] {

        margin:0px;

        width:15px;

        height:15px;

        vertical-align:middle;

    }

</style>

Верхний блок – block-header.php

<!-- Header top bar -->

<div class="top-bar">

    <div class="container">

        <div class="top-bar-left">

            <ul class="list-inline">

                <li class="dropdown flags">

                    <a href="#" class="dropdown-toggle" data-toggle="dropdown">

                        <?php

                            if($set\_lang = $this->session->userdata('language')){} else {

                                $set\_lang = $this->db->get\_where('general\_settings',array('type'=>'language'))->row()->value;

                            }

                            $lid = $this->db->get\_where('language\_list',array('db\_field'=>$set\_lang))->row()->language\_list\_id;

                            $lnm = $this->db->get\_where('language\_list',array('db\_field'=>$set\_lang))->row()->name;

                        ?>

                        <img src="<?php echo $this->crud\_model->file\_view('language\_list',$lid,'','','no','src','','','.jpg') ?>" width="20px;" alt=""/> <span class="hidden-xs"><?php echo $lnm; ?></span><i class="fa fa-angle-down"></i></a>

                        <ul role="menu" class="dropdown-menu">

                            <?php

                                $langs = $this->db->get\_where('language\_list',array('status'=>'ok'))->result\_array();

                                foreach ($langs as $row)

                                {

                            ?>

                                <li <?php if($set\_lang == $row['db\_field']){ ?>class="active"<?php } ?> >

                                    <a class="set\_langs" data-href="<?php echo base\_url(); ?>home/set\_language/<?php echo $row['db\_field']; ?>">

                                        <img src="<?php echo $this->crud\_model->file\_view('language\_list',$row['language\_list\_id'],'','','no','src','','','.jpg') ?>" width="20px;" alt=""/>

                                        <?php echo $row['name']; ?>

                                        <?php if($set\_lang == $row['db\_field']){ ?>

                                            <i class="fa fa-check"></i>

                                        <?php } ?>

                                    </a>

                                </li>

                            <?php

                                }

                            ?>

                    </ul>

                </li>

                <li class="dropdown flags" style="z-index: 1001;">

                    <a href="#" class="dropdown-toggle" data-toggle="dropdown">

                        <?php

                            if($currency\_id = $this->session->userdata('currency')){} else {

                                $currency\_id = $this->db->get\_where('business\_settings', array('type' => 'currency'))->row()->value;

                            }

                            $symbol = $this->db->get\_where('currency\_settings',array('currency\_settings\_id'=>$currency\_id))->row()->symbol;

                            $c\_name = $this->db->get\_where('currency\_settings',array('currency\_settings\_id'=>$currency\_id))->row()->name;

                        ?>

                        <span class="hidden-xs"><?php echo $c\_name; ?></span> (<?php echo $symbol; ?>)

                        <i class="fa fa-angle-down"></i>

                    </a>

                    <ul role="menu" class="dropdown-menu">

                        <?php

                            $currencies = $this->db->get\_where('currency\_settings',array('status'=>'ok'))->result\_array();

                            foreach ($currencies as $row)

                            {

                        ?>

                            <li <?php if($currency\_id == $row['currency\_settings\_id']){ ?>class="active"<?php } ?> >

                                <a class="set\_langs" data-href="<?php echo base\_url(); ?>home/set\_currency/<?php echo $row['currency\_settings\_id']; ?>">

                                    <?php echo $row['name']; ?> (<?php echo $row['symbol']; ?>)

                                    <?php if($currency\_id == $row['currency\_settings\_id']){ ?>

                                        <i class="fa fa-check"></i>

                                    <?php } ?>

                                </a>

                            </li>

                        <?php

                            }

                        ?>

                    </ul>

                </li>

                <?php if($this->crud\_model->get\_type\_name\_by\_id('general\_settings','83','value') == 'ok'){ ?>

                    <li class="dropdown flags" style="z-index: 1001;">

                        <a href="<?=base\_url()?>home/premium\_package" class="" >

                            <i class="fa fa-gift"></i> <?php echo translate('premium\_packages');?>

                        </a>

                    </li>

                <?php } ?>

                <?php if(demo()) { ?>

                    <li class="dropdown flags" style="z-index: 1001;">

                        <i class="text-danger blink\_me fa fa-exclamation-triangle" style="font-size: 16px"></i>

                        For demo purpose many operations including deletion,emailing,file uploading are <strong>DISABLED</strong>

                    </li>

                <?php } ?>

            </ul>

        </div>

        <div class="top-bar-right">

            ----- ----- -----

        </div>

    </div>

</div>

<!-- /Header top bar -->

<!-- HEADER -->

<header class="header header-logo-left">

    <div class="header-wrapper">

        <div class="container">

            <div class="flex-row">

                <div class="flex-col-6 flex-col-lg-auto">

                    <!-- Logo -->

                    <div class="logo">

                        <?php

                            $home\_top\_logo = $this->db->get\_where('ui\_settings',array('type' => 'home\_top\_logo'))->row()->value;

                        ?>

                        <a href="<?php echo base\_url();?>">

                            <img src="<?php echo base\_url(); ?>uploads/logo\_image/logo\_<?php echo $home\_top\_logo; ?>.png" alt="SuperShop"/>

                        </a>

                    </div>

                    <!-- /Logo -->

                </div>

                <div class="flex-col flex-order-last flex-order-lg-0">

                    <!-- Header search -->

                    <div class="header-search mt-4 mt-lg-0 px-xl-5">

                        <?php

                            echo form\_open(base\_url() . 'home/text\_search/', array(

                                'method' => 'post',

                                'accept-charset' => "UTF-8"

                            ));

                        ?>

                            <div class="d-flex position-relative">

                                <div class="flex-grow-1">

                                    <input class="form-control" type="text" name="query"  accept-charset="utf-8" placeholder="<?php echo translate('what\_are\_you\_looking\_for');?>?"/>

                                </div>

                                <div>

                                    <select

                                        class="selectpicker header-search-select cat\_select hidden-xs" data-live-search="true" name="category"

                                        data-toggle="tooltip" title="<?php echo translate('select');?>">

                                        <option value="0"><?php echo translate('all\_categories');?></option>

                                        <?php

                                            $categories = $this->db->get('category')->result\_array();

                                            foreach ($categories as $row1) {

                                                if($this->crud\_model->if\_publishable\_category($row1['category\_id'])){

                                        ?>

                                        <option value="<?php echo $row1['category\_id']; ?>"><?php echo $row1['category\_name']; ?></option>

                                        <?php

                                                }

                                            }

                                        ?>

                                    </select>

                                </div>

                                <div>

                                    <?php

                                        if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','58','value') == 'ok') {

                                    ?>

                                    <select

                                        class="selectpicker header-search-select" data-live-search="true" name="type" onchange="header\_search\_set(this.value);"

                                        data-toggle="tooltip" title="<?php echo translate('select');?>">

                                        <option value="product"><?php echo translate('product');?></option>

                                        <option value="vendor"><?php echo translate('vendor');?></option>

                                    </select>

                                </div>

                                <?php

                                    }

                                ?>

                                <button class="shrc\_btn"><i class="fa fa-search"></i></button>

                            </div>

                        </form>

                    </div>

                    <!-- /Header search -->

                </div>

                <div class="flex-col-6 flex-col-lg-auto text-right">

                    <!-- Header shopping cart -->

                    <div class="header-cart">

                        <div class="cart-wrapper">

                            <a href="<?php echo base\_url(); ?>home/compare" class="btn btn-theme-transparent" id="compare\_tooltip" data-toggle="tooltip" data-original-title="<?php echo $this->crud\_model->compared\_num(); ?>" data-placement="right" >

                                <i class="fa fa-exchange"></i>

                                <span class="hidden-md hidden-sm hidden-xs"><?php echo translate('compare'); ?></span>

                                (

                                <span id="compare\_num">

                                    <?php echo $this->crud\_model->compared\_num(); ?>

                                </span>

                                )

                            </a>

                            <a href="#" class="btn btn-theme-transparent" data-toggle="modal" data-target="#popup-cart">

                                <i class="fa fa-shopping-cart"></i>

                                <span class="hidden-xs">

                                    <span class="cart\_num"></span>

                                    <?php echo translate('item(s)'); ?>

                                </span>

                                <i class="fa fa-angle-down"></i>

                            </a>

                            <!-- Mobile menu toggle button -->

                            <a href="#" class="menu-toggle btn btn-theme-transparent"><i class="fa fa-bars"></i></a>

                            <!-- /Mobile menu toggle button -->

                        </div>

                    </div>

                    <!-- Header shopping cart -->

                </div>

            </div>

        </div>

    </div>

    <div class="navigation-wrapper">

        <div class="container">

            <!-- Navigation -->

            <?php

                $others\_list=$this->uri->segment(3);

            ?>

            <nav class="navigation closed clearfix">

                <a href="#" class="menu-toggle-close btn"><i class="fa fa-times"></i></a>

                <ul class="nav sf-menu">

                    <?php if($this->db->get\_where('ui\_settings',array('type'=>'header\_homepage\_status'))->row()->value == 'yes'){?>

                    <li <?php if($asset\_page=='home'){ ?>class="active"<?php } ?>>

                        <a href="<?php echo base\_url(); ?>home">

                            <?php echo translate('homepage');?>

                        </a>

                    </li>

                    <?php } if($this->db->get\_where('ui\_settings',array('type'=>'header\_all\_categories\_status'))->row()->value == 'yes'){?>

                    <li class="hidden-sm hidden-xs <?php if($asset\_page=='all\_category'){ echo 'active'; } ?>">

                        <a href="<?php echo base\_url(); ?>home/all\_category">

                            <?php echo translate('all\_categories');?>

                        </a>

                        <ul>

                            <?php

                                $all\_category = $this->db->get('category')->result\_array();

                                foreach($all\_category as $row)

                                {

                                    if($this->crud\_model->if\_publishable\_category($row['category\_id'])){

                            ?>

                            <li>

                                <a href="<?php echo base\_url(); ?>home/category/<?php echo $row['category\_id']; ?>">

                                    <?php echo $row['category\_name']; ?>

                                </a>

                            </li>

                            <?php

                                    }

                                }

                            ?>

                        </ul>

                    </li>

                    <?php } ?>

                    <li class="hidden-lg hidden-md <?php if($asset\_page=='all\_category'){ echo 'active'; } ?>">

                        <a href="#">

                            <?php echo translate('all\_categories');?>

                        </a>

                        <ul>

                            <?php

                                $all\_category = $this->db->get('category')->result\_array();

                                foreach($all\_category as $row)

                                {

                                    if($this->crud\_model->if\_publishable\_category($row['category\_id'])){

                            ?>

                            <li>

                                <a href="<?php echo base\_url(); ?>home/category/<?php echo $row['category\_id']; ?>">

                                    <?php echo $row['category\_name']; ?>

                                </a>

                            </li>

                            <?php

                                    }

                                }

                            ?>

                        </ul>

                    </li>

                    <li class="hidden-lg hidden-md <?php if($asset\_page=='all\_category'){ echo 'active'; } ?>">

                        <a href="<?php echo base\_url(); ?>home/all\_category">

                            <?php echo translate('all\_sub\_categories');?>

                        </a>

                    </li>

                    <?php if($this->db->get\_where('ui\_settings',array('type'=>'header\_featured\_products\_status'))->row()->value == 'yes'){?>

                    <li class="<?php if($others\_list=='featured'){ echo 'active'; } ?>">

                        <a href="<?php echo base\_url(); ?>home/others\_product/featured">

                            <?php echo translate('featured\_products');?>

                        </a>

                    </li>

                    <?php } if($this->db->get\_where('ui\_settings',array('type'=>'header\_todays\_deal\_status'))->row()->value == 'yes'){?>

                    <li class="<?php if($others\_list=='todays\_deal'){ echo 'active'; } ?>">

                        <a href="<?php echo base\_url(); ?>home/others\_product/todays\_deal">

                            <?php echo translate('todays\_deal');?>

                        </a>

                    </li>

                    <?php }?>

                    <?php if($this->crud\_model->get\_type\_name\_by\_id('general\_settings','82','value') == 'ok'){

                            if($this->db->get\_where('ui\_settings',array('type'=>'header\_bundled\_product\_status'))->row()->value == 'yes'){ ?>

                    <li <?php if($page\_name=='bundled\_product'){ ?>class="active"<?php } ?>>

                        <a href="<?php echo base\_url(); ?>home/bundled\_product">

                            <?php echo translate('bundled\_product');?>

                        </a>

                    </li>

                     <?php } }?>

                    <?php if(0){

                            if(1){ ?>

                    <li <?php if($page\_name=='customer\_product\_bulk\_upload'){ ?>class="active"<?php } ?>>

                        <a href="<?php echo base\_url(); ?>home/customer\_product\_bulk\_upload">

                            <?php echo translate('Bulk upload');?>

                        </a>

                    </li>

                    <?php }} if($this->crud\_model->get\_type\_name\_by\_id('general\_settings','83','value') == 'ok'){

                                if($this->db->get\_where('ui\_settings',array('type'=>'header\_classifieds\_status'))->row()->value == 'yes'){?>

                    <li <?php if($page\_name=='customer\_products'){ ?>class="active"<?php } ?>>

                        <a href="<?php echo base\_url(); ?>home/customer\_products">

                            <?php echo translate('classifieds');?>

                        </a>

                    </li>

                    <?php }} if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','58','value') !== 'ok') {

                                if($this->db->get\_where('ui\_settings',array('type'=>'header\_latest\_products\_status'))->row()->value == 'yes'){

                    ?>

                    <li class="<?php if($others\_list=='latest'){ echo 'active'; } ?>">

                        <a href="<?php echo base\_url(); ?>home/others\_product/latest">

                            <?php echo translate('latest\_products');?>

                        </a>

                    </li>

                    <?php

                        }}

                    ?>

                    <?php

                        if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','68','value') == 'ok') {

                            if($this->db->get\_where('ui\_settings',array('type'=>'header\_all\_brands\_status'))->row()->value == 'yes') {

                    ?>

                    <li <?php if($asset\_page=='all\_brands'){ ?>class="active"<?php } ?>>

                        <a href="<?php echo base\_url(); ?>home/all\_brands">

                            <?php echo translate('all\_brands');?>

                        </a>

                    </li>

                    <?php

                        }

                    }

                    ?>

                    <?php

                        if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','58','value') == 'ok') {

                            if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','81','value') == 'ok'){

                                if($this->db->get\_where('ui\_settings',array('type'=>'header\_all\_vendors\_status'))->row()->value == 'yes') {

                    ?>

                    <li <?php if($asset\_page=='all\_vendor'){ ?>class="active"<?php } ?>>

                        <a href="<?php echo base\_url(); ?>home/all\_vendor/">

                            <?php echo translate('all\_vendors');?>

                        </a>

                    </li>

                    <?php

                                }

                            }

                        }

                    ?>

                    <?php if($this->db->get\_where('ui\_settings',array('type'=>'header\_blogs\_status'))->row()->value == 'yes') {?>

                    <li class="hidden-sm hidden-xs <?php if($asset\_page=='blog'){ echo 'active'; } ?>">

                        <a href="<?php echo base\_url(); ?>home/blog">

                            <?php echo translate('blogs');?>

                        </a>

                        <ul>

                            <?php

                                $blogs=$this->db->get('blog\_category')->result\_array();

                                foreach($blogs as $row){

                            ?>

                            <li>

                                <a href="<?php echo base\_url(); ?>home/blog/<?php echo $row['blog\_category\_id']; ?>">

                                    <?php echo $row['name']; ?>

                                </a>

                            </li>

                            <?php

                                }

                            ?>

                        </ul>

                    </li>

                    <?php }?>

                    <li class="hidden-lg hidden-md <?php if($asset\_page=='blog'){ echo 'active'; } ?>">

                        <a href="#">

                            <?php echo translate('blogs');?>

                        </a>

                        <ul>

                            <?php

                                $blogs=$this->db->get('blog\_category')->result\_array();

                                foreach($blogs as $row){

                            ?>

                            <li>

                                <a href="<?php echo base\_url(); ?>home/blog/<?php echo $row['blog\_category\_id']; ?>">

                                    <?php echo $row['name']; ?>

                                </a>

                            </li>

                            <?php

                                }

                            ?>

                        </ul>

                    </li>

                    <?php

                        if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','58','value') == 'ok' && $this->crud\_model->get\_type\_name\_by\_id('general\_settings','81','value') == 'ok') {

                            if($this->db->get\_where('ui\_settings',array('type'=>'header\_store\_locator\_status'))->row()->value == 'yes') {

                    ?>

                    <li <?php if($asset\_page=='store\_locator'){ ?>class="active"<?php } ?>>

                        <a href="<?php echo base\_url(); ?>home/store\_locator">

                            <?php echo translate('store\_locator');?>

                        </a>

                    </li>

                    <?php

                            }

                        }

                    ?>

                    <?php if($this->db->get\_where('ui\_settings',array('type'=>'header\_contact\_status'))->row()->value == 'yes') {?>

                    <li <?php if($asset\_page=='contact'){ ?>class="active"<?php } ?>>

                        <a href="<?php echo base\_url(); ?>home/contact">

                            <?php echo translate('contact');?>

                        </a>

                    </li>

                    <?php } if($this->db->get\_where('ui\_settings',array('type'=>'header\_more\_status'))->row()->value == 'yes') {?>

                    <li>

                        <a href="#">

                            <?php echo translate('more');?>

                        </a>

                        <ul>

                            <?php

                                if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','58','value') == 'ok') {

                            ?>

                            <li class="<?php if($others\_list=='latest'){ echo 'active'; } ?>">

                                <a href="<?php echo base\_url(); ?>home/others\_product/latest">

                                    <?php echo translate('latest\_products');?>

                                </a>

                            </li>

                            <?php

                                }

                            ?>

                            <?php

                            $this->db->where('status','ok');

                            $all\_page = $this->db->get('page')->result\_array();

                            foreach($all\_page as $row2){

                            ?>

                            <li>

                                <a href="<?php echo base\_url(); ?>home/page/<?php echo $row2['parmalink']; ?>">

                                    <?php echo $row2['page\_name']; ?>

                                </a>

                            </li>

                            <?php

                            }

                            ?>

                        </ul>

                    </li>

                    <?php }?>

                </ul>

            </nav>

            <!-- /Navigation -->

        </div>

    </div>

</header>

<!-- /HEADER -->

<script type="text/javascript">

    $(document).ready(function(){

        $('.set\_langs').on('click',function(){

            var lang\_url = $(this).data('href');

            $.ajax({url: lang\_url, success: function(result){

                location.reload();

            }});

        });

        $('.top-bar-right').load('<?php echo base\_url(); ?>home/top\_bar\_right');

    });

</script>

<?php

if ($this->crud\_model->get\_type\_name\_by\_id('general\_settings','58','value') !== 'ok') {

?>

<style>

.header.header-logo-left .header-search .header-search-select .dropdown-toggle {

    right: 40px !important;

}

</style>

<?php

}

?>

Подключение базы данных – config/database.php

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

/\*

| -------------------------------------------------------------------

| DATABASE CONNECTIVITY SETTINGS

| -------------------------------------------------------------------

| This file will contain the settings needed to access your database.

|

| For complete instructions please consult the 'Database Connection'

| page of the User Guide.

|

| -------------------------------------------------------------------

| EXPLANATION OF VARIABLES

| -------------------------------------------------------------------

|

|   ['dsn']      The full DSN string describe a connection to the database.

|   ['hostname'] The hostname of your database server.

|   ['username'] The username used to connect to the database

|   ['password'] The password used to connect to the database

|   ['database'] The name of the database you want to connect to

|   ['dbdriver'] The database driver. e.g.: mysqli.

|           Currently supported:

|                cubrid, ibase, mssql, mysql, mysqli, oci8,

|                odbc, pdo, postgre, sqlite, sqlite3, sqlsrv

|   ['dbprefix'] You can add an optional prefix, which will be added

|                to the table name when using the  Query Builder class

|   ['pconnect'] TRUE/FALSE - Whether to use a persistent connection

|   ['db\_debug'] TRUE/FALSE - Whether database errors should be displayed.

|   ['cache\_on'] TRUE/FALSE - Enables/disables query caching

|   ['cachedir'] The path to the folder where cache files should be stored

|   ['char\_set'] The character set used in communicating with the database

|   ['dbcollat'] The character collation used in communicating with the database

|                NOTE: For MySQL and MySQLi databases, this setting is only used

|                as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7

|                (and in table creation queries made with DB Forge).

|                There is an incompatibility in PHP with mysql\_real\_escape\_string() which

|                can make your site vulnerable to SQL injection if you are using a

|                multi-byte character set and are running versions lower than these.

|                Sites using Latin-1 or UTF-8 database character set and collation are unaffected.

|   ['swap\_pre'] A default table prefix that should be swapped with the dbprefix

|   ['encrypt']  Whether or not to use an encrypted connection.

|   ['compress'] Whether or not to use client compression (MySQL only)

|   ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections

|                           - good for ensuring strict SQL while developing

|   ['failover'] array - A array with 0 or more data for connections if the main should fail.

|   ['save\_queries'] TRUE/FALSE - Whether to "save" all executed queries.

|               NOTE: Disabling this will also effectively disable both

|               $this->db->last\_query() and profiling of DB queries.

|               When you run a query, with this setting set to TRUE (default),

|               CodeIgniter will store the SQL statement for debugging purposes.

|               However, this may cause high memory usage, especially if you run

|               a lot of SQL queries ... disable this to avoid that problem.

|

| The $active\_group variable lets you choose which connection group to

| make active.  By default there is only one group (the 'default' group).

|

| The $query\_builder variables lets you determine whether or not to load

| the query builder class.

\*/

$active\_group = 'default';

$query\_builder = TRUE;

$db['default'] = array(

    'dsn'   => '',

    'hostname' => 'localhost',

    'username' => 'mysql',

    'password' => 'mysql',

    'database' => 'shop',

    'dbdriver' => 'mysqli',

    'dbprefix' => '',

    'pconnect' => FALSE,

    'db\_debug' => (ENVIRONMENT !== 'production'),

    'cache\_on' => FALSE,

    'cachedir' => '',

    'char\_set' => 'utf8',

    'dbcollat' => 'utf8\_general\_ci',

    'swap\_pre' => '',

    'encrypt' => FALSE,

    'compress' => FALSE,

    'stricton' => FALSE,

    'failover' => array(),

    'save\_queries' => TRUE

);