МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

Севастопольский государственный университет

кафедра Информационных систем

**Лисянский Александр Игоревич**

Институт информационных технологий и управления в технических системах

курс 4 группа ИC/б-42(о)

09.03.02 Информационные системы (уровень бакалавриата)

КУРСОВОЙ ПРОЕКТ

по дисциплине «Веб - технологии»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Листов 53

Отметка о зачёте \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

асс. Забаштанский А. К.

(должность) (подпись) (инициалы, фамилия)

Севастополь 2015

АННОТАЦИЯ

В документе описана поэтапная разработка личного кулинарного сайта. В пояснительной записке описывается выбор инструментальных средств, разработка раздела администратора, разработка пользовательской части сайта, тестирование разработанного сайта.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc436722147)

[1. ПОСТАНОВКА ЗАДАЧИ 6](#_Toc436722148)

[1.1 Техническое задание 6](#_Toc436722149)

[2. АНАЛИЗ И ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 9](#_Toc436722150)

[3. ПРОЕКТИРОВАНИЕ БД WEB-САЙТА 10](#_Toc436722151)

[4. ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ 12](#_Toc436722152)

[5. РАЗРАБОТКА РАЗДЕЛА АДМИНИСТРАТОРА 14](#_Toc436722153)

[5.1 Проектирование интерфейса раздела администратора 14](#_Toc436722154)

[5.2 Разработка программных модулей раздела администратора 14](#_Toc436722155)

[6. РАЗРАБОТКА РАЗДЕЛА ПОЛЬЗОВАТЕЛЯ 25](#_Toc436722156)

[6.1 Проектирование интерфейса раздела пользователя 25](#_Toc436722157)

[6.2 Разработка программных модулей раздела пользователя 26](#_Toc436722158)

[7. ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО САЙТА 31](#_Toc436722159)

[7.1 Анализ кроссбраузерности сайта 31](#_Toc436722160)

[7.2 Профилирование разработанного сайта 34](#_Toc436722161)

[7.3 Тестовые примеры работы 35](#_Toc436722162)

[ЗАКЛЮЧЕНИЕ 42](#_Toc436722163)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 43](#_Toc436722164)

[ПРИЛОЖЕНИЕ А 44](#_Toc436722165)

[ПРИЛОЖЕНИЕ Б 48](#_Toc436722166)

[ПРИЛОЖЕНИЕ В 49](#_Toc436722167)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

КУРСОВОЙ ПРОЕКТ

Разраб.

Лисянский А. И.

Провер.

*Забаштанский А.К.*

Н. Контр.

Утв.

ПОЯСНИТЕЛЬНАЯ

ЗАПИСКА

Лит.

Листов

47

Группа ИC/б-42(о)

# ВВЕДЕНИЕ

В настоящее время существует множество современных веб-технологий, которые предоставляют разработчикам неограниченные возможности для реализации своих идей. Появляются новые программные платформы, которые упрощают процесс разработки сайтов.

Целью курсового проектирования является разработка личного кулинарного сайта. Рассматриваемая тема актуальна, так как большинство сайтов о кулинарии не имеют возможности добавлять новые рецепты, на них нет форума пользователей, нет поиска по подходящим продуктам.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* изучить MVC фреймворк;
* дополнительно ознакомится с CSS фрейворком;
* написать интернет-сайта.

Пояснительная записка состоит из сети разделов. В разделе «Постановка задачи» содержится техническое задание, согласно которому будет разрабатываться интернет-магазин. В разделе «Анализ и описание предметной области» описана предметная область разрабатываемого сайта. В разделе «Проектирование БД Web-сайта» описаны таблицы, содержащиеся в базе данных, а также определены атрибуты для каждой таблицы. В разделе «Выбор инструментальных средств» описаны средства, которые использовались для разработки сайта. В разделе «Разработка раздела администратора» описывается проектирование интерфейса раздела администратора и разработка программных модулей раздела администратора. В разделе «Разработка раздела пользователя» описывается проектирование интерфейса раздела пользователя и разработка программных модулей раздела пользователя. В разделе «Тестирование разработанного сайта» выполнен анализ кроссбраузерности сайта, выполнено профилирование разработанного сайта, а также представлены тестовые примеры работы сайта.

1. ПОСТАНОВКА ЗАДАЧИ

Целью курсового проектирование является закрепление, систематизация и обобщение знаний по дисциплине «Веб - технологии», приобретение практических навыков написания интернет-магазина с помощью MVC фреймворка. Для достижения поставленной цели необходимо выполнить следующие задачи:

* анализ предметной области;
* проектирование базы данных сайта;
* создания HTML макета WEB-сайта;
* разработка раздела администратора;
* разработка раздела пользователя.

Поставленные задачи будут решатся в соответствии с техническим заданием.

* 1. Техническое задание

В рамках данного курсового проекта разрабатывается личный кулинарный сайт.

Разрабатываемый сайт делится на раздел пользователя и администрация. Структура пользовательской части сайта:

* «Главная страница»;
* «Поиск рецептов»;
* «Новые рецепты»;
* «На праздник»;
* «На каждый день»;
* «Форум»;
* «Мастер-класс»;
* «Регистрация»;
* «Войти»;
* «Выйти».

Структура раздела администратора:

* «Главная страница»;
* «Добавить рецепт»;
* «Список рецептов»;
* «Поиск рецептов»;
* «Мастер-класс»;
* «Войти(admin)»;
* «Выход (admin)».

В пользовательской части сайта меню отличается в случаях, когда пользователь авторизован и не авторизован. В том случае если пользователь не авторизован, то в меню отображается «Регистрация» и «Войти», если пользователь авторизован, то не отображается пункты «Регистрация» и «Войти», зато отображается пункт «Выйти».

На «Главной странице» любого раздела будет указана информация, что вы находитесь на главной странице определенного раздела.

На странице «Поиск рецепта» представлена возможность поиска как по названию, так и по содержанию ингредиентов. Возможность поиска по ингредиентам доступна только в разделе пользователя.

В разделах «На праздник» и «На каждый день» представлены рецепты, из соответствующих разделов, так как пользователю предоставлена возможность выбирать между праздничными и повседневными блюдами.

На странице «Новые рецепты» будут представлены новинки в порядке убывания даты добавления.

В разделе «Форум» будет виден список тем, созданных пользователями, возможность создания своей темы, и переход в тему с сообщениями пользователей. Этот раздел не администрируется, так что руководство сайта не несет ответственности за содержание данного раздела.

В разделе «Мастер-класс» в пользовательском режиме будет представлена информация о ближайшем мастер-классе, а также возможность записаться на него. В режиме администратора появиться возможность изменения описания, просмотр и удаление пользователей, записавшихся на мероприятие.

Вверху страницы представлено меню авторизации с возможностью регистрации. Если пользователь авторизирован, меню будет видоизменено на информацию о пользователе и кнопку «Выход» для выхода из своей учетной записи. В качестве хранения пользовательских данных будут использоваться метод шифрованных данных сессий.

1. АНАЛИЗ И ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Кулинария – обширный пласт в жизни человека, так как без еды человек конечно живет, но не долго)). Кулинария многих народов поражает своей глубиной вкусов, уникальностью рецептов. Поэтому люди хотят создавать единое хранилище описания вкусняшек, чтобы доступ к нему был в любое время и в любом месте. Собрать качественные, проверенные рецепты в одном месте – вот главная цель кулинарного сайта, чтобы люди могли готовить свои любимые блюда легко, быстро и с радостью потом все это лопать.

На многих кулинарных сайтах есть масса хороших или даже отличных рецептов, видео по приготовлению конкретного блюда (даже может быть каким-либо знаменитым шеф поваром), но почти все сайты не предоставляют возможность подобрать рецепт по тому, что находиться у тебя в холодильнике. В этом и заключается фишка моего сайта – возможность подбора рецептов по заданным продуктам.

Главная задача сайта показать людям, которые так же любят готовить, что можно насладиться процессом приготовления хороших рецептов, не ища их на просторах интернета и не мучаясь в разборе найденных рецептов, так как у нас есть пошаговые инструкции по приготовлению блюд, есть подбор блюд по вашим продуктам.

1. ПРОЕКТИРОВАНИЕ БД WEB-САЙТА

Для разрабатываемого сайта необходима база данных, которая содержит следующие таблицы:

* раздел;
* продукт;
* тип;
* рецепт;
* рецепт\_продукт;
* пользователь;
* комментарий;
* мастер-класс;
* мастер-класс\_пользователь;
* форум;
* сообщение;
* форум\_сообщение.

Для каждой таблицы определены следующие атрибуты:

* раздел (id, название);
* продукт (id, название);
* тип (id, название);
* рецепт (id, id\_раздела, название, рецепт, картинка, дата);
* рецепт\_продукт (id\_рецепта. id\_продукта, количество, id\_типа);
* пользователь (id, логин, пароль, ФИО, емаил, кометирование);
* комментарий (id, id\_пользователя, id\_рецепта, комментарий, дата);
* мастер-класс (id, тема, описание, дата);
* мастер-класс\_пользователь (id\_мастер-класса, id\_пользователя);
* форум (id, тема);
* сообщение (id, id \_пользователя, сообщение);
* форум\_сообщение (id\_темы, id\_сообщения).

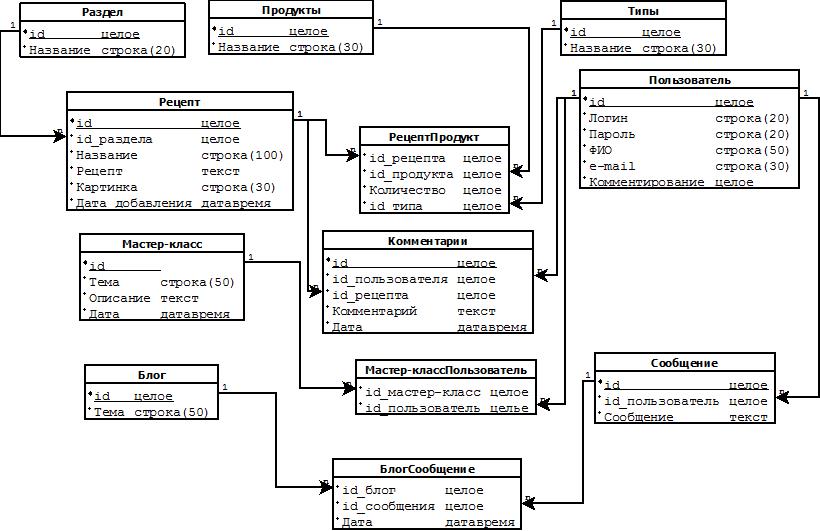


Рисунок 3.1 – Логическая схема базы данных

В соответствии со схемой созданы таблицы база данных. SQL для создания таблиц представлен в приложении Б.

1. ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ

CodeIgniter 3.0.1 - Фреймворк, на основе которого можно создать свою собственную систему управления контентом. Предусмотрено довольно много функций в самой базовой конфигурации. Для работы использует технологию MVC - Модель-Вид-Контроллер.

CodeIgniter фреймворк имеет множество возможностей, некоторые из них перечислены ниже:

* Парадигма MVC;
* Интерфейсы DAO и ActiveRecord для работы с базами данных (PDO);
* Быстрота работы;
* Кэширование страниц и отдельных фрагментов;
* Перехват и обработка ошибок;
* Ввод и валидация форм;
* Использование AJAX и интеграция с jQuery;
* Генерация базового PHP кода для CRUD операций (скаффолдинг);
* Поддержка тем оформления для их лёгкой смены;
* Возможность подключения сторонних библиотек или написания собственных;
* Миграции базы данных.

На основании всего вышеперечисленного для написания сайта был выбран MVC фреймворк CodeIgniter.

Bootstrap – это CSS/HTML фреймворк для создания веб-сайтов. Другими словами, это набор инструментов для создания веб-макета. У него есть ряд преимуществ, что делает его самым популярным среди других подобных фреймворков. Преимущества Bootstrap заключаются в следующем:

* Скорость работы – создание макетов с Bootstrap занимает меньше времени благодаря большому набору готовых к использованию элементов.
* Гибкость – добавление новых элементов не нарушает общую структуру благодаря динамически изменяющейся сетке.
* Легкая изменяемость – правка стилей достигается за счет добавления новых CSS правил, которые переопределяют существующие.
* Большое количество шаблонов – этот момент будет отдельно рассмотрен далее.
* Совместимость с браузерами – Bootstrap совместим с Mozilla Firefox, Google Chrome, Safari, Internet Explorer и Opera.

В Bootstrap существует возможность адаптации разработанного сайта для мобильных устройств. Bootstrap легко и эффективно масштабирует проект с одной базой кода, от телефонов и планшетов до настольных компьютеров.

Исходя из вышеперечисленного в качестве CSS фреймворка был выбран Bootstrap.

библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

При разработке сайта использовался сервер Apache 2.4.9, версия РНР 5.4.12. Для реализации базы данных использовался MySQL 5.6. В качестве среды разработки будет использоваться NetBeans.

1. РАЗРАБОТКА РАЗДЕЛА АДМИНИСТРАТОРА
   1. Проектирование интерфейса раздела администратора

В случае входа в систему от имени администратора меню будет выглядеть как на рисунке 5.1.

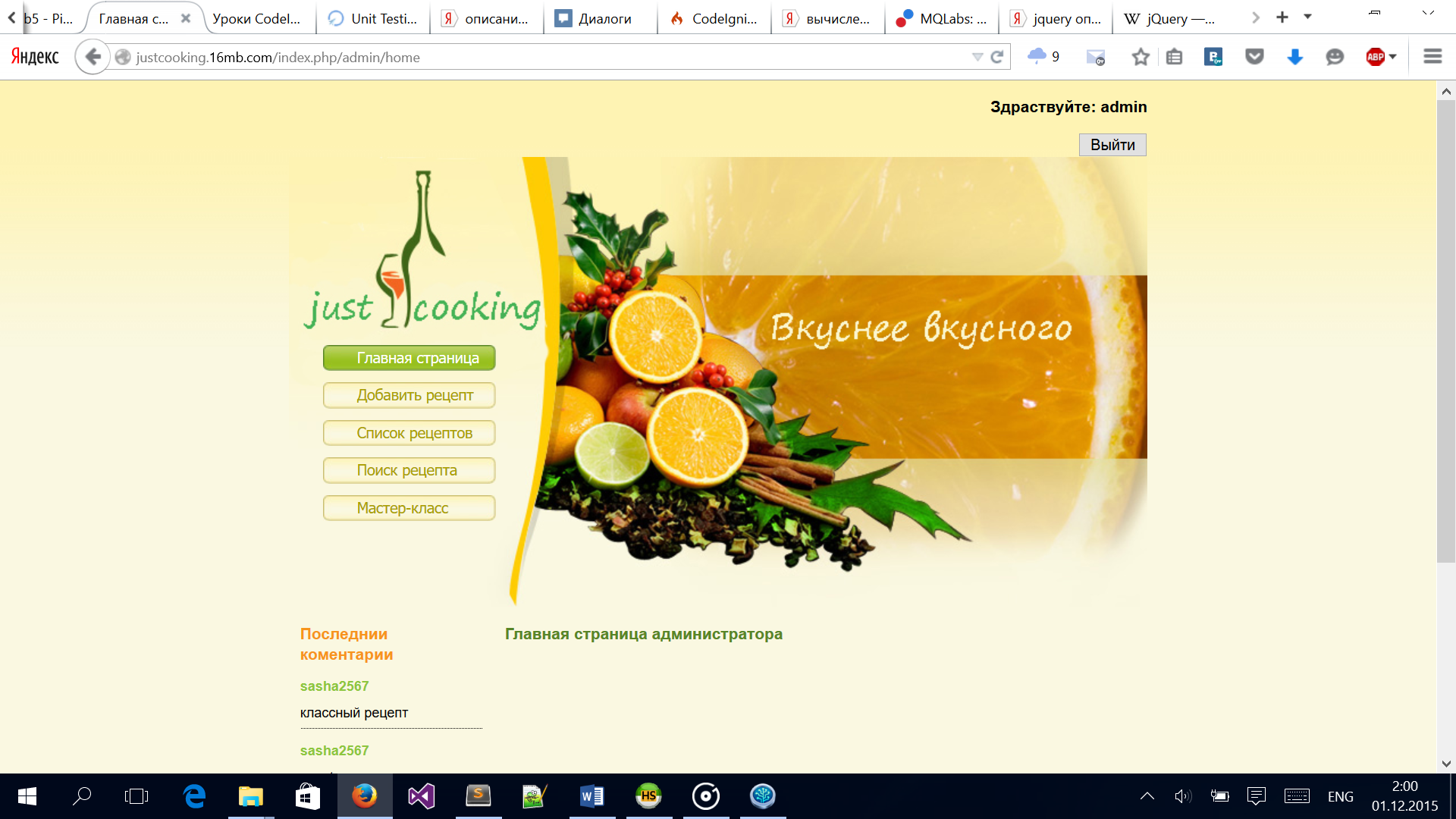


Рисунок 5.1 – Меню раздела администратора

Код отображающий меню раздела администратора представлен ниже.

<div id="templatemo\_header">

<div id="templatemo\_menu">

<ul>

<li><a href="<?=base\_url();?>admin/home" <?php if($uri == 'admin/home') echo 'class="current"';?>>Главная&nbsp;страница</a></li>

<li><a href="<?=base\_url();?>admin/addrecipe" <?php if($uri == 'admin/addrecipe') echo 'class="current"';?>>Добавить&nbsp;рецепт</a></li>

<li><a href="<?=base\_url();?>admin/recipelist" <?php if($uri == 'admin/recipelist') echo 'class="current"';?>>Список&nbsp;рецептов</a></li>

<li><a href="<?=base\_url();?>admin/searchs/index" <?php if($uri == 'admin/search/index') echo 'class="current"';?>>Поиск&nbsp;рецепта</a></li>

<li><a href="<?=base\_url();?>admin/master/index" <?php if($uri == 'admin/master/index') echo 'class="current"';?>>Мастер-класс</a></li>

</ul>

</div>

</div>

* 1. Разработка программных модулей раздела администратора

Для реализации раздела администратора были разработаны различные модели, представления и формы.

Для реализации возможности добавления и изменения рецептов на соответствующих страницах была реализована модель Recipe.php. Код модели представлен ниже.

<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Recipe extends CI\_Model {

var $table = 'recipe';

var $product\_table = 'products';

var $recipe\_product\_table = 'recipe\_product';

var $type\_table = 'types';

var $key\_id = 'recipe\_id';

var $section\_id = 'section\_id';

var $product\_id = 'product\_id';

function \_\_construct(){

parent::\_\_construct();

}

/\*\*

\* add data

\*/

function add($data){

$id = count($this->getlist()) + 1;

$data[$this->key\_id] = $id;

$this->db->insert($this->table, $data);

return $id;

}

function addIngredients($data)

{

foreach ($data as $value) {

$this->db->insert($this->recipe\_product\_table, $value);

}

}

/\*\*

\* edit data

\*/

function edit($id, $data){

$this->db->where($this->key\_id, $id);

$this->db->update($this->table, $data);

}

/\*\*

\* delete data

\*/

function delete($id){

$this->db->where($this->key\_id, $id);

$this->db->delete($this->table);

}

/\*\*

\* get info about

\*/

function get($id){

$this->db->where($this->key\_id, $id);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get info about

\*/

function getsection($id){

$this->db->where($this->section\_id, $id);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get info about

\*/

function getproduct($id){

$this->db->where($this->key\_id, $id);

$query = $this->db->get($this->recipe\_product\_table);

$temp = $query->result\_array();

$result = array();

foreach ($temp as $value) {

$this->db->where($this->product\_id, $value[$this->product\_id]);

$query = $this->db->get($this->product\_table);

$tmp = $query->result\_array();

foreach ($tmp as $var) {

$tmp = $var;

break;

}

$type = $value['type\_id'];

$this->db->where('type\_id', $type);

$query = $this->db->get($this->type\_table);

$tmpp = $query->result\_array();

foreach ($tmpp as $var) {

$tmpp = $var;

break;

}

$type = $tmpp['type'];

$tmp = $tmp['description'];

$result[] = array(

'product' => $tmp,

'count' => $value['count'],

'type' => $type

);

}

return $result;

}

/\*\*

\* get info about

\*/

function getproductlist()

{

$query = $this->db->get($this->product\_table);

return $query->result\_array();

}

/\*\*

\* get info about

\*/

function getproductid($value='')

{

$this->db->like('description', $value);

$query = $this->db->get($this->product\_table);

return $query->result\_array();

}

function getsearch($value='', $id = 0)

{

$this->db->like('description', $value);

$this->db->limit (10, ($id - 1) \* 10);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get info about

\*/

function getrecipeid($value = null)

{

if($value != null){

$str = "(".$value[0];

$i = 1;

for ($i=1; $i < count($value); $i++) {

$str .= ", ".$value[$i];

}

$str .= ")";

$query = $this->db->query("SELECT DISTINCT `recipe\_id` FROM `web\_recipe\_product` А WHERE NOT exists (SELECT `product\_id` FROM `web\_products` WHERE `product\_id` in ".$str." AND NOT exists (SELECT `recipe\_id`, `product\_id` FROM `web\_recipe\_product` В WHERE В.`product\_id` = `web\_products`.`product\_id` AND А.`recipe\_id` =В.`recipe\_id`))");

return $query->result\_array();

}

}

/\*\*

\* get info about

\*/

function gettypeslist()

{

$query = $this->db->get($this->type\_table);

return $query->result\_array();

}

/\*\*

\* get list of

\*/

function getlist(){

$query = $this->db->get($this->table);

return $query->result\_array();

}

}

Для реализации возможности разрешения или запрета комментирования и удобной работы с данными пользователя была реализована модель User.php.

<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class User extends CI\_Model {

var $table = 'user';

var $key\_id = 'user\_id';

var $uname = 'username';

function \_\_construct(){

parent::\_\_construct();

}

/\*\*

\* add data

\*/

function add($data){

$this->db->insert($this->table, $data);

}

/\*\*

\* edit data

\*/

function edit($id, $data){

$this->db->where($this->key\_id, $id);

$this->db->update($this->table, $data);

}

/\*\*

\* delete data

\*/

function delete($id){

$this->db->where($this->key\_id, $id);

$this->db->delete($this->table);

}

/\*\*

\* get info

\*/

function get($id){

$this->db->where($this->key\_id, $id);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get info

\*/

function getUser($username)

{

$this->db->where($this->uname, $username);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get list

\*/

function getlist(){

$query = $this->db->get($this->table);

return $query->result\_array();

}

}

Для более удобной работы с разделами «мастер-класс» и «форум» были написаны следующие модели:

Модель Coment.php

<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Coment extends CI\_Model {

var $table = 'coment';

var $key\_id = 'coment\_id';

var $user\_id = 'user\_id';

var $recipe\_id = 'recipe\_id';

function \_\_construct(){

parent::\_\_construct();

}

/\*\*

\* add data

\*/

function add($data){

$this->db->insert($this->table, $data);

}

/\*\*

\* edit data

\*/

function edit($id, $data){

$this->db->where($this->key\_id, $id);

$this->db->update($this->table, $data);

}

/\*\*

\* delete data

\*/

function delete($id){

$this->db->where($this->key\_id, $id);

$this->db->delete($this->table);

}

/\*\*

\* get info about

\*/

function get($id){

$this->db->where($this->key\_id, $id);

$query = $this->db->get($this->table);

return $query->result\_array();

}

function getuser($id){

$this->db->where($this->user\_id, $id);

$query = $this->db->get($this->table);

return $query->result\_array();

}

function getrecipe($id){

$this->db->where($this->recipe\_id, $id);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get list of

\*/

function getlist(){

$query = $this->db->get($this->table);

return $query->result\_array();

}

Модель Master.php

<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Masters extends CI\_Model {

var $table = 'master';

var $key\_id = 'master\_id';

var $table\_master = 'master\_user';

var $table\_user = 'user';

var $user\_key\_id = 'user\_id';

function \_\_construct(){

parent::\_\_construct();

}

/\*\*

\* add data

\*/

function add($data){

$this->db->insert($this->table, $data);

}

/\*\*

\* add data

\*/

function adduser($id)

{

$this->db->order\_by('date', 'desc');

$query = $this->db->get($this->table);

$query = $query->result\_array();

$query = $query[0];

$data = array(

'master\_id' => $query['master\_id'],

'user\_id' => $id

);

$this->db->insert($this->table\_master, $data);

}

/\*\*

\* add data

\*/

function deleteuser($id)

{

$this->db->order\_by('date', 'desc');

$query = $this->db->get($this->table);

$query = $query->result\_array();

$query = $query[0];

$data = array(

$this->key\_id => $query[$this->key\_id],

$this->user\_key\_id => $id

);

$this->db->delete($this->table\_master, $data);

}

/\*\*

\* get info

\*/

function get(){

$this->db->order\_by('date', 'desc');

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get info

\*/

function getUser($userid)

{

$this->db->where($this->user\_key\_id, $userid);

$query = $this->db->get($this->table\_user);

return $query->result\_array();

}

/\*\*

\* get list

\*/

function getlist(){

$this->db->order\_by('date', 'desc');

$query = $this->db->get($this->table);

$query = $query->result\_array();

$query = $query[0];

$this->db->where($this->key\_id, $query[$this->key\_id]);

$query = $this->db->get($this->table\_master);

return $query->result\_array();

}

}

Модель Forum.php

<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Forum extends CI\_Model {

var $table = 'theme';

var $table\_theme = 'theme\_messege';

var $table\_messege = 'messege';

var $table\_user = 'user';

var $key\_id = 'theme\_id';

var $messege\_id = 'messege\_id';

var $user\_id = 'user\_id';

function \_\_construct(){

parent::\_\_construct();

}

/\*\*

\* add data

\*/

function addtheme($data){

$this->db->insert($this->table, $data);

$this->db->where('theme', $data['theme']);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* add data

\*/

function addmessege($t\_id,$u\_id,$m){

$query = $this->db->get($this->table\_messege);

$c = count($query->result\_array()) + 1;

$data = array(

$this->messege\_id => $c,

$this->user\_id => $u\_id,

'messege' => $m

);

$this->db->insert($this->table\_messege, $data);

$data = array(

$this->messege\_id => $c,

$this->key\_id => $t\_id

);

$this->db->insert($this->table\_theme, $data);

}

/\*\*

\* get info

\*/

function gettheme($id = 1){

$this->db->limit (10, ($id - 1) \* 10);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get info

\*/

function getthemebyid($id = 1){

$this->db->where ($this->key\_id,$id);

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get info

\*/

function get(){

$query = $this->db->get($this->table);

return $query->result\_array();

}

/\*\*

\* get list

\*/

function getmessege($id){

$this->db->select('messege.messege\_id, messege.user\_id, messege.messege, user.username');

$this->db->from('theme\_messege');

$this->db->join('messege', 'messege.messege\_id = theme\_messege.messege\_id');

$this->db->join('user', 'user.user\_id = messege.user\_id');

$this->db->where('theme\_messege.theme\_id', $id);

$query = $this->db->get();

return $query->result\_array();

}

}

С помощью PHPUnit были написаны тесты моделей. Код теста для модели Goods.php представлен ниже.

<?php

class GoodsTest extends CDbTestCase {

public $fixtures = array(

'goods' => 'Goods',

);

//тестирование добавление записи в таблицу

public function testAdd() {

$goods = new Goods();

$goods->setAttributes(array(

'goodsname' => 'name 1',

'price' => '900',

'imagefile' => '1.jpg',

'description' => 'description 1',

), false);

$this->assertTrue($goods->save(false));

// проверить наличие товара

$goods = Goods::model()->findByPk($goods->id);

$this->assertTrue($goods instanceof Goods);

$this->assertEquals('name 1', $goods->goodsname);

}

//тестирование выполнения измененияя записи

public function testUpdate() {

$goods = new Goods();

$goods = Goods::model()->findByPk($goods->id);

$goods->setAttributes(array(

'goodsname' => $goods->goodsname,

'price' => $goods->price,

'imagefile' => $goods->imagefile,

'description' => 'change description',

), false);

$this->assertTrue($goods->save(false));

// проверить изменение товара

$goods = Goods::model()->findByPk($goods->id);

$this->assertTrue($goods instanceof Goods);

$this->assertEquals('change description', $goods->description);

}

//тестирование удаления записи из таблицы

public function testDelete() {

$goods = new Goods();

$goods = Goods::model()->findByPk($goods->id);

$goods->delete();

$goods = Goods::model()->findByPk($goods->id);

$this->assertTrue($goods instanceof Goods);

$this->assertEquals('change description', $goods->description);

}

}

?>

Тестирование модели News.php проводилось аналогично. На рисунке 5.2 представлены результаты выполнения тестов.

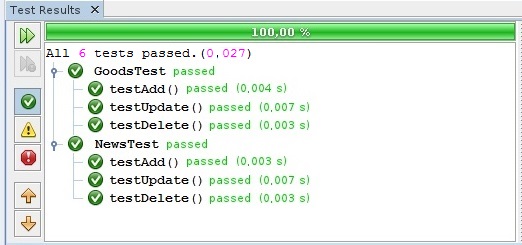


Рисунок 5.2 – Результаты выполнения тестов

Тесты выполнены успешно, значит работа с базой организована верно.

1. РАЗРАБОТКА РАЗДЕЛА ПОЛЬЗОВАТЕЛЯ
   1. Проектирование интерфейса раздела пользователя

В случае если пользователь не авторизован, то меню будет выглядеть как на рисунке 6.1.



Рисунок 6.1 – Меню для неавторизованного пользователя

Если пользователь зашел в систему под своим email и паролем, то меню будет выглядеть как на рисунке 6.2. Вторая строка основного меню содержит подменю.



Рисунок 6.2 – Меню для авторизованного пользователя

Код отображающий меню раздела пользователя представлен ниже.

<div id="mainmenu">

<?php

$user = new SiteUser();

if (is\_numeric(Yii::app()->session['UserId']))

$user = SiteUser::model()->findByPk(Yii::app()->session['UserId']);

if ($\_GET['language'] === 'ru') {

$this->widget('zii.widgets.CMenu', array(

'items' => array(

array('label' => 'Главная страница', 'url' => array('/site/index&language=ru')),

array('label' => 'Привет, ' . $user->name, 'url' => array('/site/basket&language=ru'), 'visible' => is\_numeric(Yii::app()->session['UserId']),

'items' => array(

array('label' => 'Изменить данные', 'url' => array('/site/change&language=ru')),

array('label' => 'Корзина, ' . Yii::app()->session['amount\_basket'], 'url' => array('/site/basket&language=ru')),

array('label' => 'Выход (' . $user->name . ')', 'url' => array('/site/userlogout&language=ru')),

)),

array('label' => 'Новости и акции', 'url' => array('/site/news&language=ru')),

array('label' => 'Товары', 'url' => array('/site/goods&language=ru&page=1')),

array('label' => 'Контакты', 'url' => array('/site/contact&language=ru')),

array('label' => 'Информация для потребителя', 'url' => array('/site/info&language=ru')),

array('label' => 'Регистрация', 'url' => array('/site/registration&language=ru'), 'visible' => !Yii::app()->session['UserId']),

array('label' => 'Войти', 'url' => array('/site/userlogin&language=ru'), 'visible' => !Yii::app()->session['UserId']),

array('label' => 'Администрирование', 'url' => array('/site/login&language=ru'), 'visible' => !Yii::app()->session['UserId'])),

));

} else {

$this->widget('zii.widgets.CMenu', array(

'items' => array(

…

));

} ?>

</div>

* 1. Разработка программных модулей раздела пользователя

Для реализации раздела пользователя были разработаны различные модели, представления и формы.

Для реализации возможности просмотра новостей и акций на соответствующей странице было реализовано представление News.php. Код представления представлен ниже.

<?php

if ($\_GET['language'] === 'ru') {

$this->pageTitle = Yii::app()->name;

$this->pageTitle = 'Новости и акции';

$this->breadcrumbs = array(

'Новости и акции',

);

} else {

$this->pageTitle = Yii::app()->name;

$this->pageTitle = 'News and shares';

$this->breadcrumbs = array(

'News and shares',

);

}

$news = new News();

$news = $news->findAll(); //Поиск все записей таблицы новости

?>

<div class="form">

<?php

$form = $this->beginWidget('CActiveForm', array(

'id' => 'news-form',

'enableClientValidation' => true,

'clientOptions' => array(

'validateOnSubmit' => true,

),

));

?>

<?php if (count($news) > 0) { //если в таблице есть записи, то вывести их?>

<?php foreach ($news as $br) { ?>

<form method="POST" action="">

<div id="contener-news">

<input type='text' name='newsId' value='<?= $br->id ?>' style="display:none"/>

<div id="contener-news-title"><?= "$br->title" ?></div>

<table>

<tr>

<td><img src='images/news/<?= $br->imagefile ?>' width="100%"></td>

<td><?= $br->text ?></td>

</tr>

</table>

<?php

if ($\_GET['language'] === 'ru')

echo CHtml::submitButton('Подробно', array('id' => 'button-out-contener', 'name' => 'details'));

else

echo CHtml::submitButton('Details', array('id' => 'button-out-contener', 'name' => 'details'));

?>

</div>

</form>

<?php }

} else { //Вывод сообщения если нет новостей

?>

<h4>На данный момент нет новостей</h4>

<?php }

$this->endWidget(); ?>

</div>

Для просмотра товара на странице «Товары» было реализовано представление Goods.php. Код представления представлен ниже.

<?php

if ($\_GET['language'] === 'ru') {

$this->pageTitle = Yii::app()->name;

$this->pageTitle = 'Товары';

$this->breadcrumbs = array(

'Товары',

);

} else {

$this->pageTitle = Yii::app()->name;

$this->pageTitle = 'Goods';

$this->breadcrumbs = array(

'Goods',

);

}

//Постраничное отображение

$criteria = new CDbCriteria();

$criteria->order = 'id DESC';

$count = Goods::model()->count($criteria);

$pagination = new CPagination($count);

$pagination->pageSize = 3; // Количество элементов на страницу

$pagination->pageVar = $\_GET['page']; // Имя GET переменной хранения текущего индекса страницы.

$pagination->applyLimit($criteria);

$pagination->route = 'site/goods'; //маршрут для отображения постраничной навигации.

$goods = Goods::model()->findAll($criteria);

?>

<div class="form">

<?php

$form = $this->beginWidget('CActiveForm', array(

'id' => 'goods-form',

'enableClientValidation' => true,

'clientOptions' => array(

'validateOnSubmit' => true,

),

));

//Вывод записей о товаре

if (count($goods) > 0) { /\*Если есть записи в таблице, то выводим (вывод товаров аналогичен выводу новостей) \*/?>

<div id="contener-goods">

…

</div>

<?php } else { ?>

<h4>В каталоге нет товаров!</h4>

<?php }

$this->endWidget(); ?>

</div><!-- form -->

<div id="nav"> //Отображения номеров страниц

<?php

$this->widget('CLinkPager', array(

'pages' => $pagination,

));

?>

</div>

Для просмотра страниц «Контакт» и «Информация для пользователя» реализованы представления Contact.php и Info.php. Код представления представлен ниже.

<?php

if ($\_GET['language'] === 'ru') {

$this->pageTitle = Yii::app()->name;

$this->pageTitle = 'Контакты';

$this->breadcrumbs = array(

'Контакты',

);

} else {

$this->pageTitle = Yii::app()->name;

$this->pageTitle = 'Contact';

$this->breadcrumbs = array(

'Contact',

);

}

?>

<?php

$content = new Content(); //Отображение контента страницы из БД

$content = Content::model()->findByPk('contact');

<div id="div-content" ><?= $content->pagecontent ?></div>

Для реализации возможности регистрации пользователя на соответствующей странице была реализована модель SiteUser.php и форма RegistrationForm.php. Код модели SiteUser.php представлен ниже.

<?php

class SiteUser extends CActiveRecord {

//Функция возвращает имя таблицы

public function tableName() {

return 'siteuser';

}

/\*Правила для валидации атрибутов модели, описываются только атрибуты, которые вводит пользователь\*/

public function rules() {

return array(

array('name, surname, otchestvo, country, city, address, email, password', 'required'),

array('name, surname, otchestvo', 'length', 'max' => 30),

array('country, city, email, password', 'length', 'max' => 50),

array('address', 'length', 'max' => 150),

array('id, name, surname, otchestvo, country, city, address, email, password', 'safe', 'on' => 'search'),

);

}

//Сопоставление атрибутам labels (name=>label)

public function attributeLabels() {

return array(

'id' => 'ID',

'name' => 'Name',

'surname' => 'Surname',

'otchestvo' => 'Otchestvo',

'country' => 'Country',

'city' => 'City',

'address' => 'Address',

'email' => 'Email',

'password' => 'Password',

);

}

…

} ?>

Код формы RegistrationForm.php реализован аналогично с GoodsForm.php. Приведем только часть кода отвечающую за валидацию формы.

public function rules() {

return array(

// поля обязательные для заполнения

array('name, surname, otchestvo, country, city, address, email, password', 'required'),

array('email', 'authenticate'),

//поля должны быть без пробелов и без цифр

array('name, surname, otchestvo', 'match', 'pattern' => '#^[а-яА-яa-zA-Z]+$#ius', 'message' => 'Должны быть только буквы и без пробелов'),

array('name, surname, otchestvo', 'match', 'pattern' => '#^[^0-9]+$#ius', 'message' => 'Не должно быть цифр'),

//сравнение email с шаблоном

array('email', 'email'),

//поля не содержат цифр

array('country, city', 'match', 'pattern' => '#^[^0-9\_]+$#ius', 'message' => 'Не должно быть цифр'),

//пароль содержит латинские буквы, цифры и специальные символы

array('password', 'match', 'pattern' => '#^[a-zA-Z0-9]+$#ius', 'message' => 'Пароль должен содержать цифры и латинские буквы'),

);

}

С помощью PHPUnit были написаны тесты моделей Basket.php и SiteUser.php. Код тестирования моделей аналогичен, коду тестирования модели Goods.php. На рисунке 6.3 представлены результаты выполнения тестов.

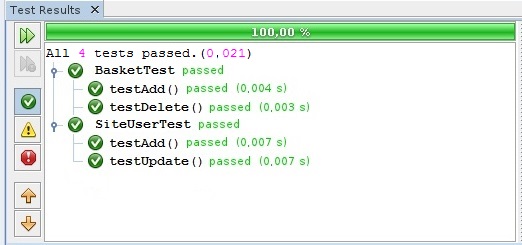


Рисунок 6.3 – Результаты выполнения тестов

Тесты выполнены успешно, значит работа с базой организована верно.

1. ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО САЙТА
   1. Анализ кроссбраузерности сайта

Проверка кроссбраузерности сайта в браузерах Google Chrome, Mozilla Firefox, Opera выполнялась в онлайн сервисе Browsershots, а для Internet Explorer в сервисе [IE NetRenderer](http://www.ipinfo.info/netrenderer/). Далее будут представлены результаты тестирования кроссбраузерности.

Отображение главной страницы и страницы «Товары» в Google Chrome 11 показано на рисунке 7.1.

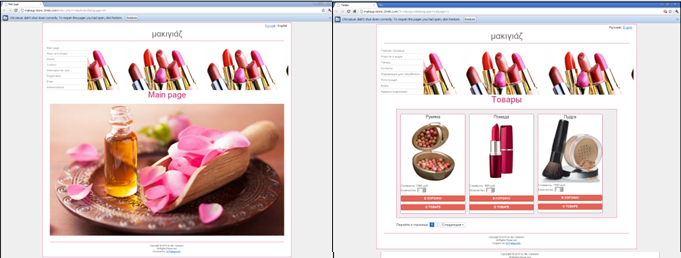


Рисунок 7.1 – Отображение страниц в Google Chrome 11

На рисунке 7.2 показано отображение страниц в браузере Firefox 38.0.

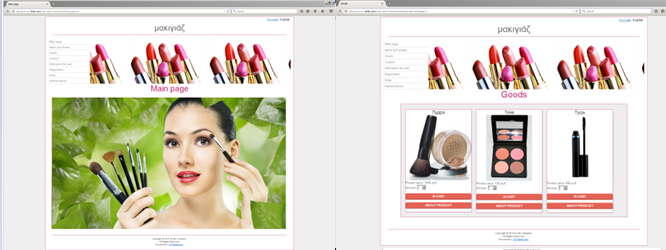


Рисунок 7.2 – Отображение страниц в Firefox 38.0

Главная страница в браузере Opera 12.2 выглядит как на рисунке 7.3.

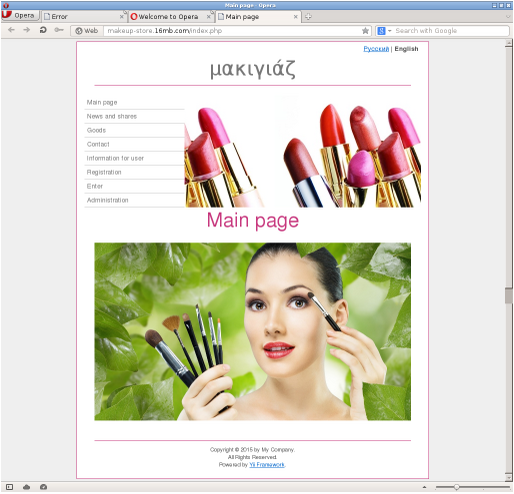


Рисунок 7.3 – Отображение главной страницы в Opera 12.2

Страница «Товары» и главная страница в браузере Internet Explorer 8 выглядит как на рисунке 7.4.

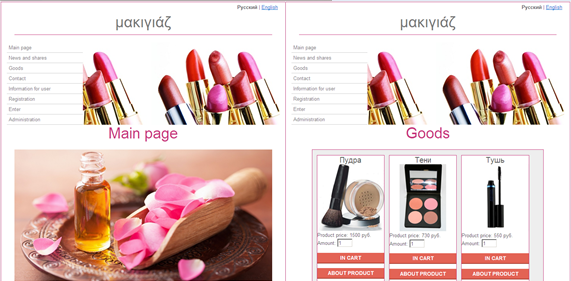


Рисунок 7.4 – Отображение страниц в Internet Explorer 8

Интерфейс разработанного сайта отображается в различных браузерах идентично.

Для проверки кроссбраузерности JavaScript кода в различных браузерах использовалась система Saucelabs.

Выполним проверку на корректность сохранения контента измененного с помощью редактора Wysihtml5 (реализовано на JavaScript). Выполним вход от имени администратора, изменим контент страницы и проверим наличие изменения в разделе пользователя.

Результаты изменения контента страницы «Информация для пользователя» в браузере Google Chrome 30 отображено на рисунке 7.5.

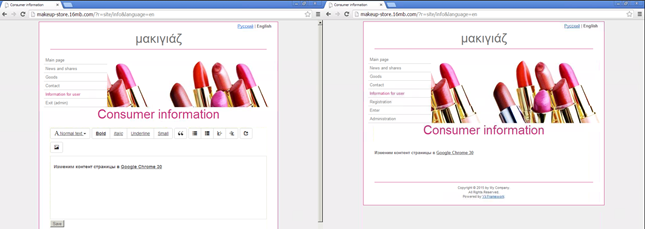


Рисунок 7.5 – Изменение контента в браузере Google Chrome 30

Изменим контент страницы «Контакт» в браузере Firefox 38, результаты изменения отображены на рисунке 7.6

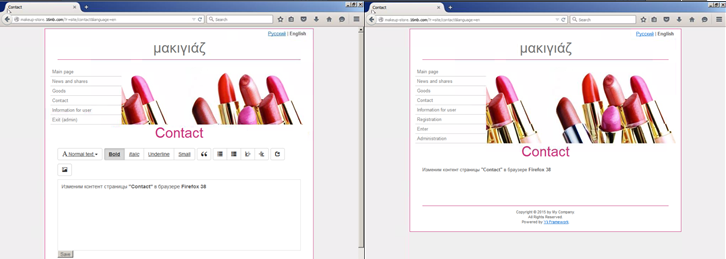


Рисунок 7.6 – Изменение контента в браузере Firefox 38

Результаты изменения контента страницы «Информация для пользователя» в браузере Opera 12 отображено на рисунке 7.7.

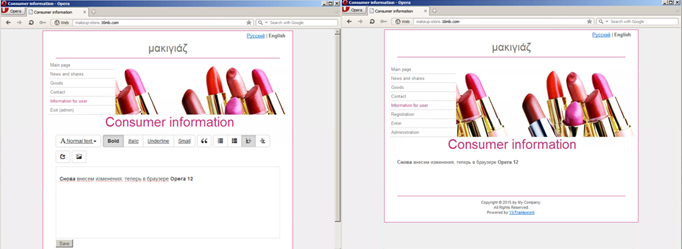


Рисунок 7.7 – Изменение контента в браузере Opera 12

Результаты изменения контента страницы «Контакт» в браузере Internet Explorer 8 отображено на рисунке 7.8.

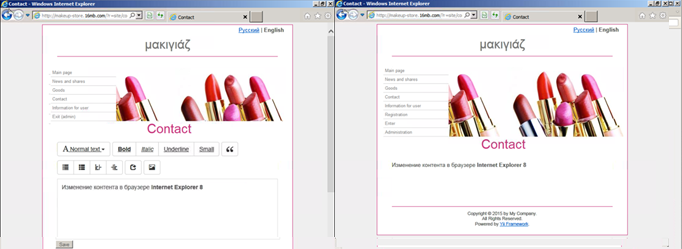


Рисунок 7.8 – Изменение контента в браузере Internet Explorer 8

Функционирование JavaScript-кода в различных браузерах происходит правильно.

* 1. Профилирование разработанного сайта

При выполнении профилирования разработанного сайта было определено время загрузки страниц сайта, результаты представлены в таблице 7.1.

Таблица 7.1 – Время загрузки и объем переданного трафика страниц

|  |  |  |
| --- | --- | --- |
| Наименование страницы | Время загрузки, мс | Объем переданного трафика, КВ |
| Главная страница | 1500 | 5,6 |
| Новости и акции | 718 | 2,9 |
| Товары | 746 | 3,5 |
| Контакт | 1140 | 3,6 |
| Информация для пользователя | 1100 | 2,9 |
| Регистрация | 1420 | 4,9 |
| Войти | 825 | 3,0 |
| Информация о товаре | 538 | 3,3 |
| Информация о новости | 1830 | 5,5 |
| Изменение данных пользователя | 1040 | 3,7 |

* 1. Тестовые примеры работы

Разработанный сайт размещен на бесплатном Web-хостинге hostinger.ru. Url разработанного сайта <http://makeup-store.16mb.com> . При переходе на данную ссылку сайта открывает главная страница интернет-магазина.

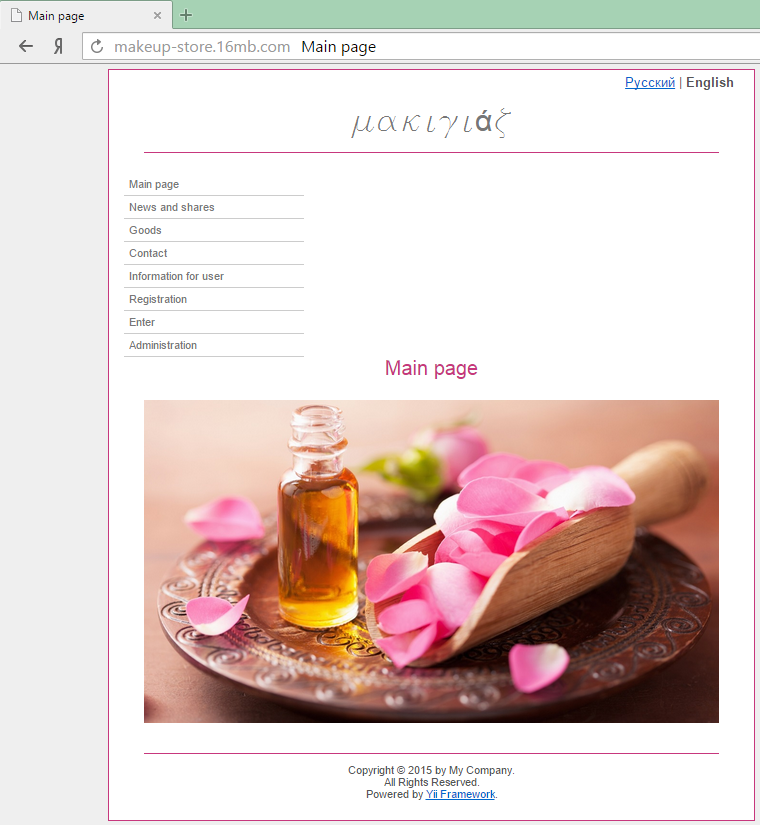


Рисунок 7.9 – Главная страница сайта

Для того чтобы зарегистрироваться на сайте необходимо перейти на страницу «Регистрация» (на английском «Registration»).

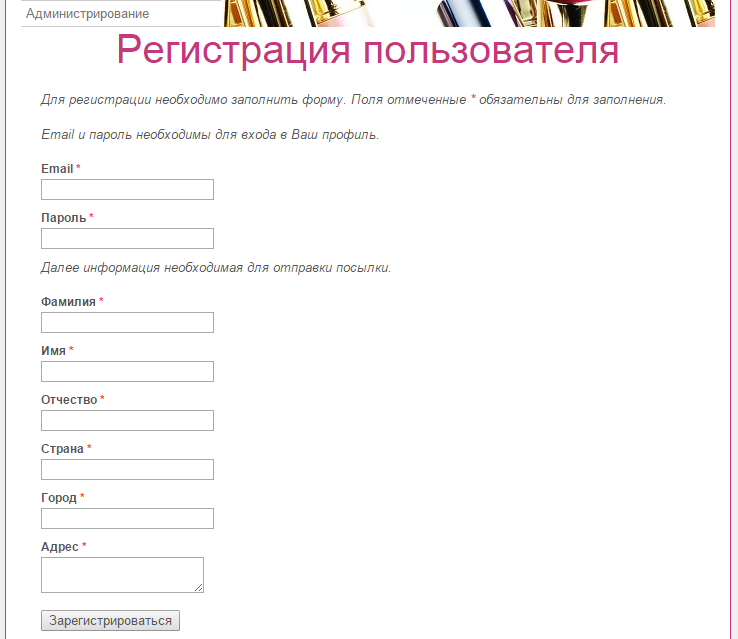


Рисунок 7.10 – Страница «Регистрация»

На странице «Регистрация» выполняется валидация полей ввода. Email должен содержать @. Пароль должен быть введен латинскими буквами, допускаются цифры и специальные символы. Фамилия, имя и отчество должно быть на русском или английском без цифр и без букв. Поля страна и город могут быть написаны на русском или английском с содержанием пробела, но не допускается ввод цифр.

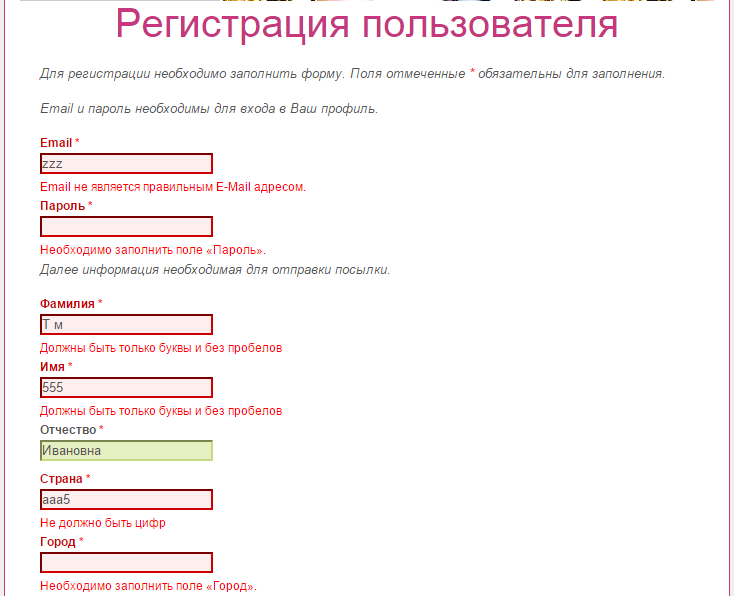


Рисунок 7.11 – Пример неверного заполнения полей

Для того чтобы выполнить вход в систему от имени пользователя необходимо перейти на страницу «Вход» («Enter»). Для авторизации необходимо ввести email и пароль, которые были введены при регистрации.

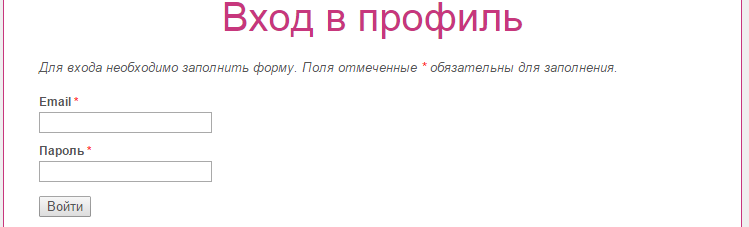


Рисунок 7.12 – Страница «Вход»

Выполнив вход в систему под своим email и паролем можно изменить данные пользователя перейдя на страницу «Изменение данных» («Change data»). При открытии страницы в поля ввода добавляется имеющаяся информация о пользователе. Новые данные проверяются точно также, как и при регистрации.

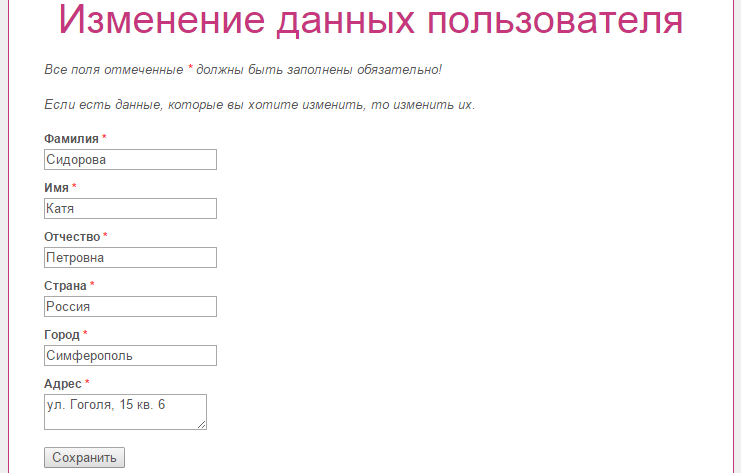


Рисунок 7.13 – Страница «Изменение данных пользователя»

При переходе на страницу «Новости и акции» пользователь может просмотреть список новостей и акций. В случае отсутствия новостей и акций будет выведено соответствующее сообщение.

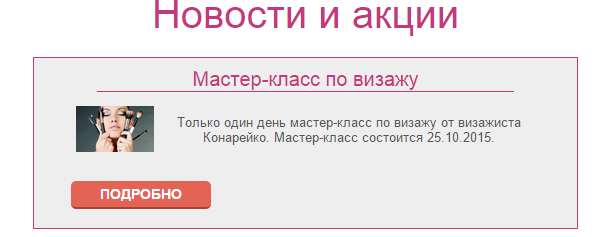


Рисунок 7.14 – Страница «Новости и акции»

При нажатии на кнопку «Подробно», выполнится переход на страницу «Подробная информация о новости», если новость отображает проведение какого либо мероприятия, то место проведения можно указать на карте Google Map.

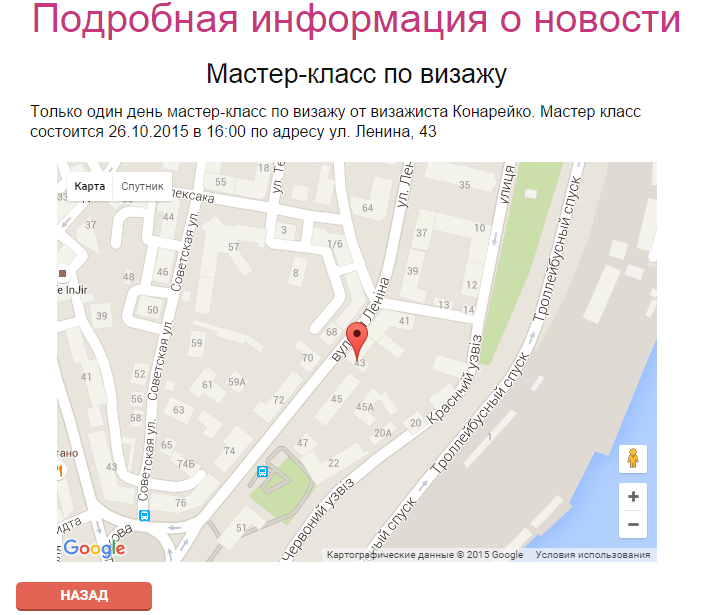


Рисунок 7.15 – Страница «Подробная информация о новости»

При переходе на страницу «Товары» отобразится список товаров, которые представлены в интернет-магазине.



Рисунок 7.16 – Страница «Товары»

Пользователь может заказать товар со страницы «Товары» для этого необходимо указать количество товара и нажать на кнопку «В корзину» («In cart»). Для того чтобы просмотреть подробное описание товара необходимо нажать на кнопку «О товаре» («About product»). Со страницы с подробным описанием товара, также можно выполнить заказ соответствующего товара.

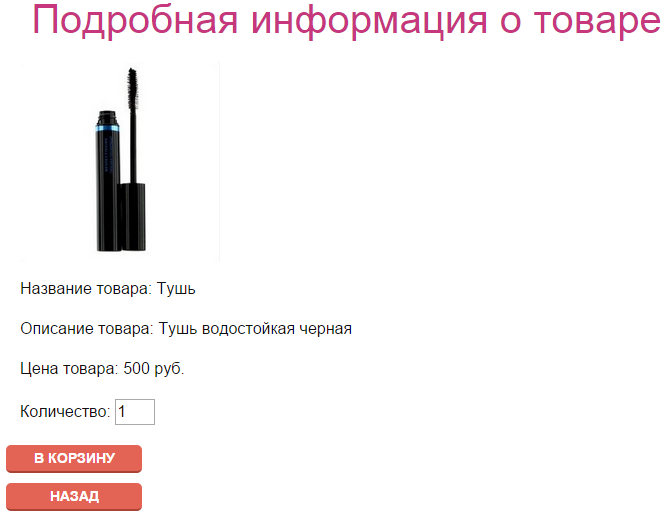


Рисунок 7.17 – Описание товара

После нажатия на кнопку «В корзину» («In cart») товар попадает в корзину. Для того чтобы перейти в корзину необходимо нажать на пункт меню «Корзина» («Cart»). На странице «Корзина» можно выполнять заказ товаров по одному или нажать на кнопку «Заказать все», тогда выполнится заказ всех товаров, которые находятся в корзине.

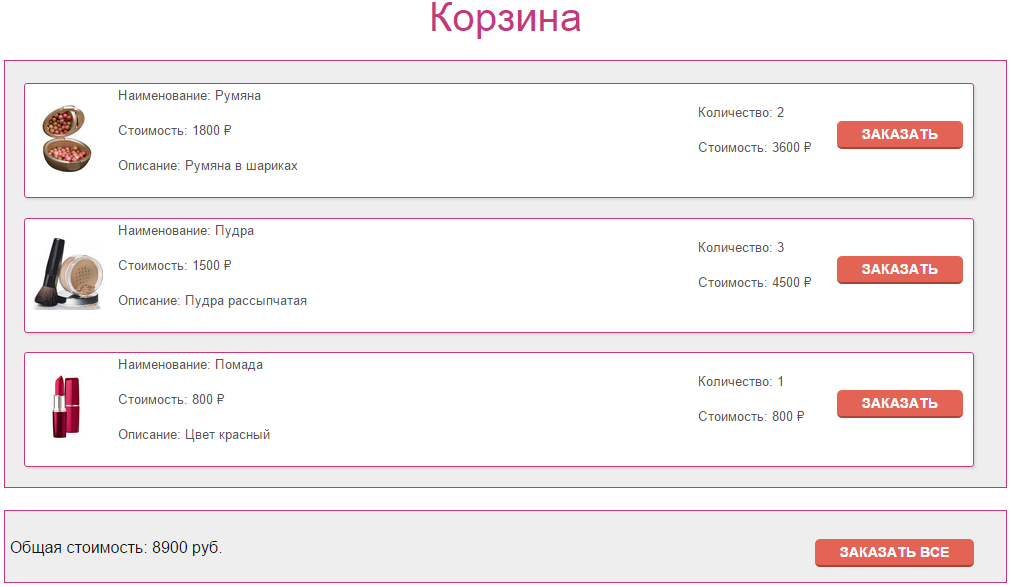


Рисунок 7.18 – Страница «Корзина»

На странице «Контакт» («Contact») можно посмотреть контактную информацию. При переходе на страницу «Информация для пользователя» («Information for user») пользователь может просмотреть интересующую его информацию.

Выполним вход от имени администратора. Для этого перейдем на страницу «Администрирование» («Administration»). Администратор может добавлять, изменять и удалять новости и акции, товары, а также изменять контент страниц «Контакт» и «Информация для пользователя».

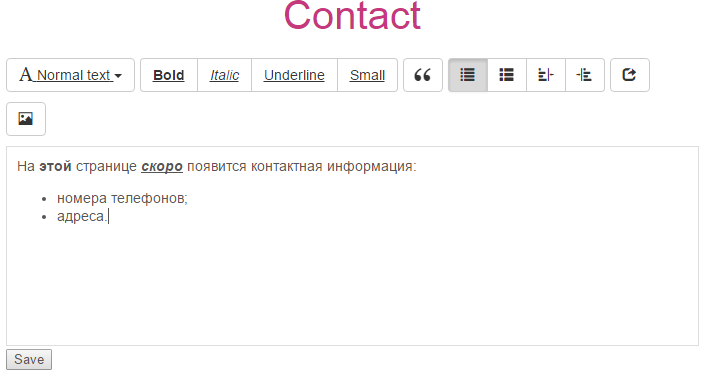


Рисунок 7.19 – Изменение страницы «Контакт»

Интернет-магазин необходимо было адаптировать под мобильные устройства. Результаты тестирования разработанного сайта на смартфоне представлены в приложении А.

# ЗАКЛЮЧЕНИЕ

В данном курсовом проекте был реализован интернет-магазин женской косметики. Для достижения поставленной цели был выполнен анализ предметной области и поэтапное проектирование системы, которое включает в себя создание HTML-макета сайта, разработка раздела администратора и раздела пользователя. Для выбранной предметной области была разработана база данных, которая содержим следующие таблицы: товары, новости и акции, пользователи сайта, корзина.

Интернет-магазин был разработан с использованием MVC фраймворка Yii. Помимо MVC фраймворка использовался CSS фраймворк Bootstrap. Разработанный интернет-магазин имеет разграничение прав доступа. Интернет-магазин размещен на бесплатном Web-хостинге hostinger.ru. Также написанный интернет-магазин был проверен на кроссбраузерность. Интернет-магазин отображается и работает во всех популярных браузерах (Google Chrome, Opera, Firefox, а также Internet Explorer) идентично – без изменения внешнего вида и функциональности.

Все цели курсового проектирования были достигнуты. В результате выполнения курсового проектирования были закреплены навыки создания web-сайтов с использованием шаблона проектирования модель-представление-контроллер и разработан интернет-магазин женской косметики.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Полное руководство по Yii [Электронный ресурс]. — Режим доступа: <http://yiiframework.ru/doc/guide/ru/index> . — Последний доступ: 03.10.2015. — Название с экрана.
2. Методические указания к выполнению курсовой работы по дисциплине «Web-технологии» для студентов направления 09.03.02 «Информационные системы и технологии» / А. Л. Овчинников. — Севастополь: Изд-во СевГУ, 2015. — 31 с.
3. Makarov A. Yii 1.1 Application Development Cookbook [Электронный ресурс] / A. Makarov. — Birmingham: Packt publishing, 2011. — 371. — Режим доступа: <http://indev.org.ua/download/yii_application_develop.pdf> . — Последний доступ: 25.10.2015. — Название с экрана.
4. Spurlock J. Bootstrap / J. Spurlock. — Sebastopol: O’Reilly Media, 2013. — 110 с.

# ПРИЛОЖЕНИЕ А

В приложении А представлены изображения, демонстрирующие отображение сайта на смартфоне.

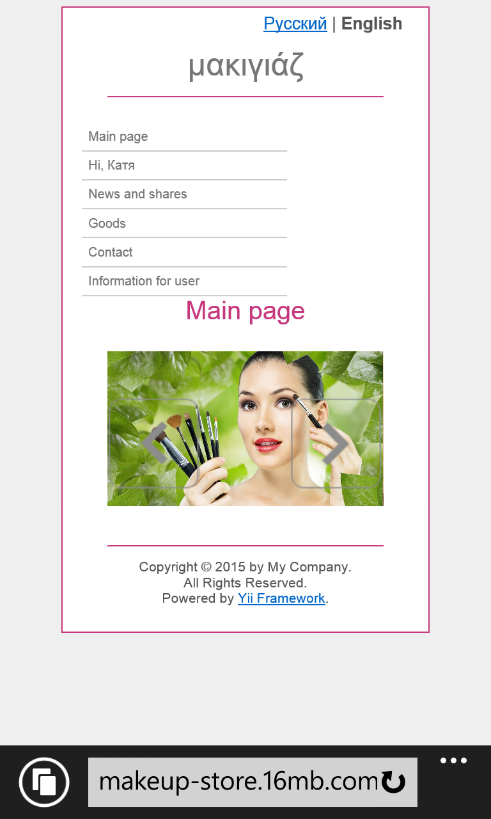


Рисунок А.1 – Главная страница

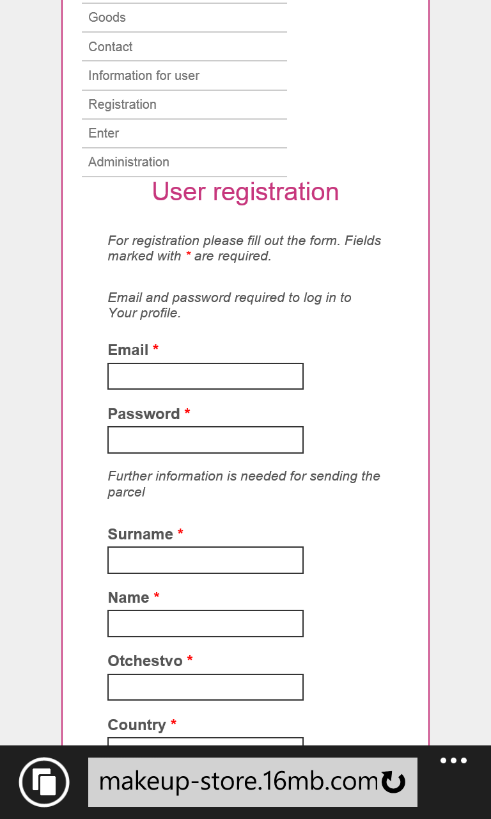


Рисунок А.2 – Страница «Регистрации»

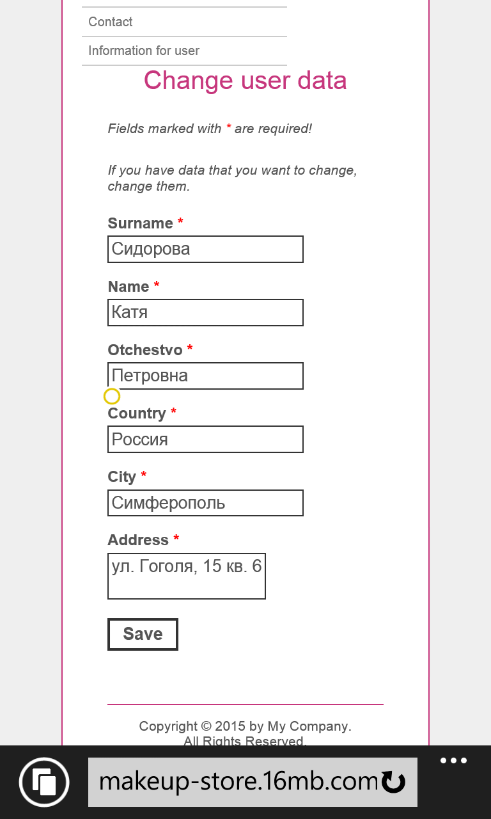


Рисунок А.3 – Страница «Изменение информации пользователя»

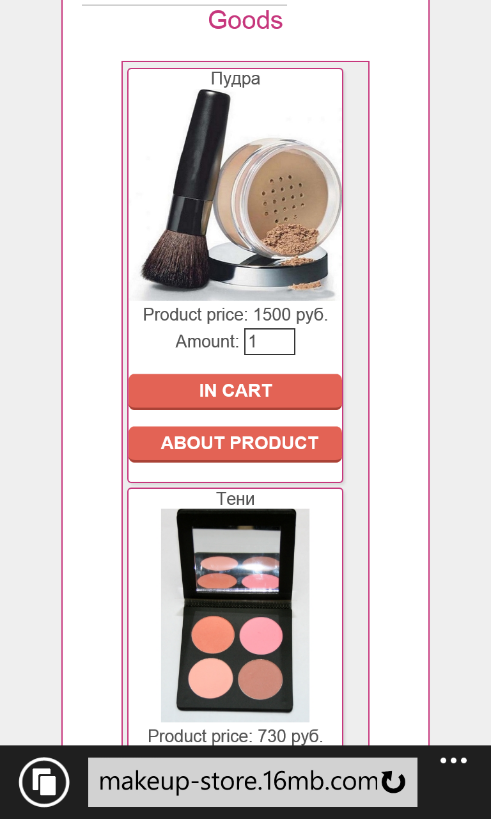


Рисунок А.4 – Страница «Товары»

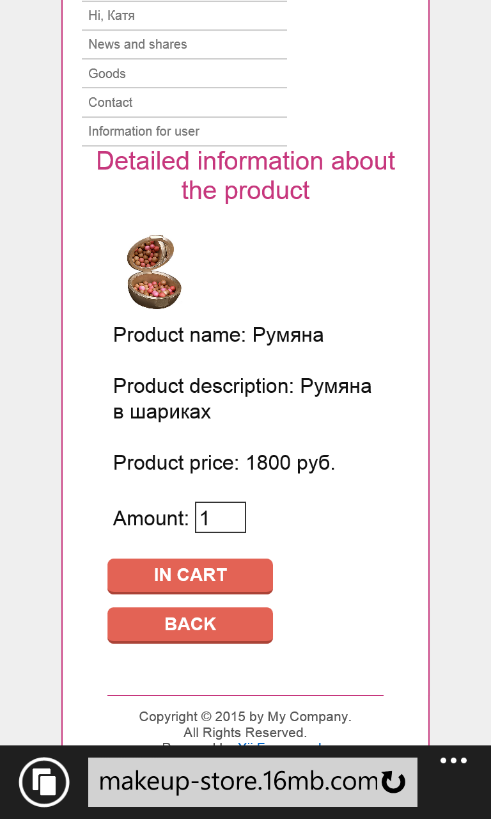


Рисунок А.5 – Страница «Подробная информация о товаре»

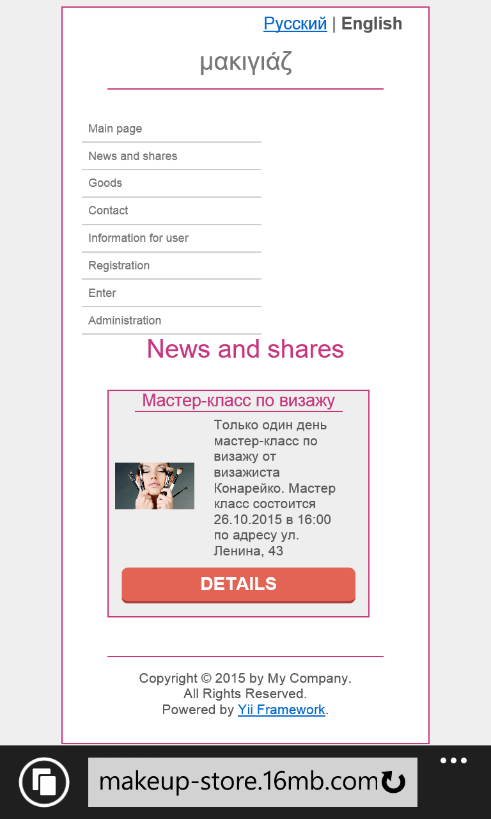


Рисунок А.6 – Страница «Новости и акции»

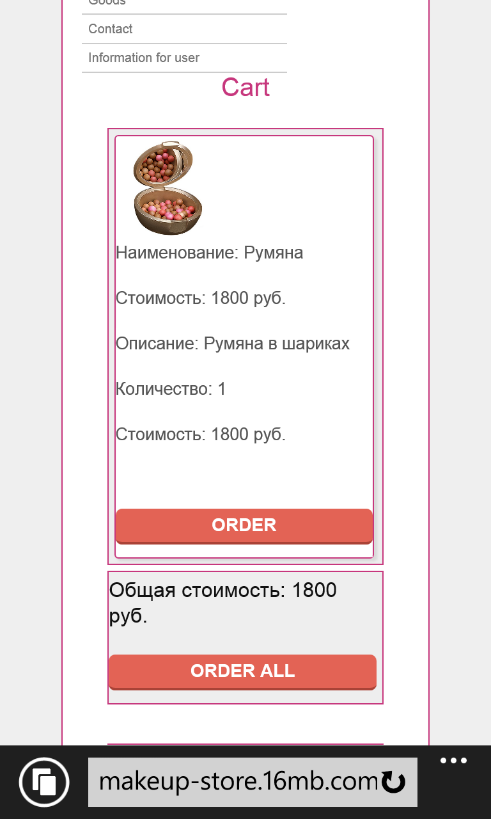


Рисунок А.7 – Страница «Корзина»

# ПРИЛОЖЕНИЕ Б

SQL для создания база данных сайта.

CREATE TABLE IF NOT EXISTS `basket` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`userid` varchar(6) NOT NULL,

`goodsid` int(11) NOT NULL,

`amount` int(11) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=10 ;

CREATE TABLE IF NOT EXISTS `content` (

`pagename` varchar(100) NOT NULL,

`pagecontent` longtext NOT NULL,

PRIMARY KEY (`pagename`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `goods` (

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`goodsname` varchar(50) NOT NULL,

`price` varchar(20) NOT NULL,

`imagefile` varchar(100) NOT NULL,

`description` varchar(400) NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`id`) REFERENCES `basket` (`goodsid`) ON DELETE CASCADE ON UPDATE CASCADE,

UNIQUE KEY `id` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=64 ;

CREATE TABLE IF NOT EXISTS `news` (

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`title` varchar(100) NOT NULL,

`imagefile` varchar(100) NOT NULL,

`text` varchar(400) NOT NULL,

`location` varchar(100) DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `id` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=16 ;

CREATE TABLE IF NOT EXISTS `siteuser` (

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`name` varchar(30) NOT NULL,

`surname` varchar(30) NOT NULL,

`otchestvo` varchar(30) NOT NULL,

`country` varchar(50) NOT NULL,

`city` varchar(50) NOT NULL,

`address` varchar(150) NOT NULL,

`email` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`id`) REFERENCES `basket` (`userid`) ON DELETE CASCADE ON UPDATE CASCADE,

UNIQUE KEY `id` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=10 ;

# ПРИЛОЖЕНИЕ В

Исходный код разработанных модулей

<?php

class SiteController extends Controller {

//Определение базовой функции actions. Вызывает отображение страницы сайта

public function actions() {

return array(

'captcha' => array(

'class' => 'CCaptchaAction',

'backColor' => 0xFFFFFF,

),

'page' => array(

'class' => 'CViewAction',

),

);

}

//Вызывает отображение главной страницы

public function actionIndex() {

$this->render('index');

}

//Вызывает отображение страницы новости и акции

public function actionNews() {

$model = new NewsForm;

$lang = $\_GET['language'];

if (isset($\_POST['ajax']) && $\_POST['ajax'] === 'news-form') {

echo CActiveForm::validate($model);

Yii::app()->end();

}

if (isset($\_POST['NewsForm']) && isset($\_POST['save-add'])) {

$model->attributes = $\_POST['NewsForm'];

// если введенные данные прошли валидацию

if ($model->validate()) {

$newNews = new News();

$newNews->title = $model->title;

$newNews->imagefile = $model->imagefile;

$newNews->text = $model->text;

$newNews->location = $model->location;

$newNews->save();

}

}

if (isset($\_POST['details'])) {

$newsid = $\_POST['newsId'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/detailsNews&language=' . $lang . '&id=' . $newsid);

}

if (isset($\_POST['delete'])) {

$newsId = $\_POST['newsId'];

$delNews = new News();

$delNews->deleteByPk($newsId);

$this->redirect(Yii::app()->baseUrl . '/index.php?r=site/news&language=' . $lang);

}

if (isset($\_POST['change'])) {

$newsid = $\_POST['newsId'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/changeNews&language=' . $lang . '&id=' . $newsid);

}

$this->render('../site/pages/news', array('model' => $model));

}

//Вызывает отображение страницы товары

public function actionGoods() {

$model = new GoodsForm;

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

$lang = $\_GET['language'];

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

if (isset($\_POST['ajax']) && $\_POST['ajax'] === 'goods-form') {

echo CActiveForm::validate($model);

Yii::app()->end();

}

if (isset($\_POST['GoodsForm']) && isset($\_POST['save-add'])) {

$model->attributes = $\_POST['GoodsForm'];

if ($model->validate()) {

$newGoods = new Goods();

$newGoods->goodsname = $model->goodsname;

$newGoods->price = $model->price;

$newGoods->imagefile = $model->imagefile;

$newGoods->description = $model->description;

$newGoods->save();

}

}

if (isset($\_POST['details'])) {

$goodsid = $\_POST['goodsId'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/details&language=' . $lang . '&page=' . $page . '&id=' . $goodsid);

}

if (isset($\_POST['basket'])) {

if (!is\_numeric(Yii::app()->session['UserId'])) {?>

<script>

lang = "<?= $\_GET['language'] ?>"

page = "<?= $\_GET['page'] ?>";

//Проверка авторизации пользователя

if (confirm('Для того чтобы добавить товар в корзину, необходимо выполнить вход в систему. Перейти на страницу авторизации?'))

document.location = 'http://localhost/web/?r=site/userlogin&language=' + lang;

else

document.location = 'http://localhost/web/?r=site/goods&language=' + lang + '&page=' + page;

</script>

<?php

} else {

$basket = new Basket();

$basket->userid = Yii::app()->session['UserId'];

$basket->goodsid = $\_POST['goodsId'];

$basket->amount = $\_POST['amount'];

$basket->save();

$goods = Goods::model()->findByPk($\_POST['goodsId']);

Yii::app()->session['sum\_basket'] += $\_POST['amount'] \* $goods->price;

Yii::app()->session['amount\_basket'] += $\_POST['amount'];

}

}

if (isset($\_POST['delete'])) {

$goodsid = $\_POST['goodsId'];

$delGoods = new Goods();

$delGoods->deleteByPk($goodsid);

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/goods&language=' . $lang . '&page=' . $page);

}

if (isset($\_POST['change'])) {

$goodsid = $\_POST['goodsId'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/changegoods&language=' . $lang . '&page=' . $page . '&id=' . $goodsid);

}

$this->render('../site/pages/goods', array('model' => $model));

}

//Вызывает отображение страницы изменения данных товара

public function actionChangeGoods() {

$this->render('../site/pages/changeGoods');

}

//Вызывает отображение страницы изменения новостей

public function actionChangeNews() {

$this->render('../site/pages/changeNews');

}

//Вызывает отображение страницы контакт

public function actionContact() {

$this->render('../site/pages/contact');

}

//Вызывает отображение страницы подробной информации о новости

public function actionDetailsNews() {

$this->render('../site/pages/detailsNews');

}

//Вызывает отображение страницы информации для пользователя

public function actionInfo() {

$this->render('../site/pages/info');

}

//Вызывает отображение страницы регистрации

public function actionRegistration() {

$modelReg = new RegistrationForm;

if (isset($\_POST['ajax']) && $\_POST['ajax'] === 'registration-form') {

echo CActiveForm::validate($modelReg);

Yii::app()->end();

}

// сохранение информации введенной пользователем

if (isset($\_POST['RegistrationForm'])) {

$modelReg->attributes = $\_POST['RegistrationForm'];

if ($modelReg->validate()) {

$user = new SiteUser();

$user->email = $modelReg->email;

$user->password = $modelReg->password;

$user->surname = $modelReg->surname;

$user->name = $modelReg->name;

$user->otchestvo = $modelReg->otchestvo;

$user->country = $modelReg->country;

$user->city = $modelReg->city;

$user->address = $modelReg->address;

$user->save();

$lang = $\_GET['language'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/index&language=' . $lang);

}

}

$this->render('pages/registration', array('modelReg' => $modelReg));

}

//Вызывает отображение страницы авторизации пользователя

public function actionUserlogin() {

$modelUser = new UserLoginForm;

if (isset($\_POST['ajax']) && $\_POST['ajax'] === 'user-login-form') {

echo CActiveForm::validate($modelUser);

Yii::app()->end();

}

if (isset($\_POST['UserLoginForm'])) {

$modelUser->attributes = $\_POST['UserLoginForm'];

if ($modelUser->validate()) {

$loginUser = new SiteUser();

$loginUser = SiteUser::model()->findAll();

foreach ($loginUser as $checkUser) {

if ($checkUser->email === $modelUser->email && $checkUser->password === $modelUser->password) {

Yii::app()->session['UserId'] = $checkUser->id;

$findBasket = new Basket();

$findBasket = Basket::model()->findAll();

$find\_flag = 0;

foreach ($findBasket as $findBasket) {

if ($findBasket->userid === Yii::app()->session['UserId']) {

$goods = Goods::model()->findByPk($findBasket->goodsid);

Yii::app()->session['amount\_basket'] += $findBasket->amount;

Yii::app()->session['sum\_basket'] += $findBasket->amount \* $goods->price;

$find\_flag = 1;

}

}

if ($find\_flag === 0) {

Yii::app()->session['amount\_basket'] = 0;

Yii::app()->session['sum\_basket'] = 0;

}

$lang = $\_GET['language'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/index&language=' . $lang);

}

}

}

}

$this->render('pages/userLogin', array('modelUser' => $modelUser));

}

//Вызывает отображение страницы описания товара

public function actionDetails() {

$this->render('../site/pages/details');

}

//Вызывает отображение страницы корзина

public function actionBasket() {

$this->render('../site/pages/basket');

}

//Выход пользователя из системы

public function actionUserlogout() {

Yii::app()->user->logout();

unset(Yii::app()->session['UserId']);

unset(Yii::app()->session['amount\_basket']);

unset(Yii::app()->session['sum\_basket']);

$lang = $\_GET['language'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/index&language=' . $lang);

}

public function actionChange() {

$modelChange = new ChangeInfoForm();

if (isset($\_POST['ajax']) && $\_POST['ajax'] === 'change-info-form') {

echo CActiveForm::validate($modelChange);

Yii::app()->end();

}

if (isset($\_POST['ChangeInfoForm'])) {

$modelChange->attributes = $\_POST['ChangeInfoForm'];

if ($modelChange->validate()) {

$user = SiteUser::model()->findByPk(Yii::app()->session['UserId']);

$user->surname = $modelChange->surname;

$user->name = $modelChange->name;

$user->otchestvo = $modelChange->otchestvo;

$user->country = $modelChange->country;

$user->city = $modelChange->city;

$user->address = $modelChange->address;

$user->update();

$lang = $\_GET['language'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/index&language=' . $lang);

}

}

$this->render('pages/changeInfo', array('modelChange' => $modelChange));

}

//action вызывается при возникновении существенных ошибок

public function actionError() {

if ($error = Yii::app()->errorHandler->error) {

if (Yii::app()->request->isAjaxRequest)

echo $error['message'];

else

$this->render('error', $error);

}

}

//Вызывает отображение страницы авторизации администратора

public function actionLogin() {

$model = new LoginForm;

if (isset($\_POST['ajax']) && $\_POST['ajax'] === 'login-form') {

echo CActiveForm::validate($model);

Yii::app()->end();

}

if (isset($\_POST['LoginForm'])) {

$model->attributes = $\_POST['LoginForm'];

// validate user input and redirect to the previous page if valid

if ($model->validate() && $model->username === 'admin' && $model->password === '') {

Yii::app()->session['UserId'] = 'admin';

$lang = $\_GET['language'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/index&language=' . $lang);

}

}

$this->render('login', array('model' => $model));

}

//Выход администратора из системы

public function actionLogout() {

unset(Yii::app()->session['UserId']);

$lang = $\_GET['language'];

$this->redirect(Yii::app()->homeUrl . 'index.php?r=site/index&language=' . $lang);

}

}