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 64

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](#_Toc437209701)

[1. ПОСТАНОВКА ЗАДАЧИ 6](#_Toc437209702)

[1.1 Техническое задание 6](#_Toc437209703)

[2. АНАЛИЗ И ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 9](#_Toc437209704)

[3. ПРОЕКТИРОВАНИЕ WEB-САЙТА 10](#_Toc437209705)

[3.1 Проектирование базы данных сайта 10](#_Toc437209706)

[3.2 Проектирование сайта 11](#_Toc437209707)

[4. ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ 14](#_Toc437209708)

[5. РАЗРАБОТКА РАЗДЕЛА АДМИНИСТРАТОРА 16](#_Toc437209709)

[5.1 Проектирование интерфейса раздела администратора 16](#_Toc437209710)

[5.2 Разработка программных модулей раздела администратора 17](#_Toc437209711)

[6. РАЗРАБОТКА РАЗДЕЛА ПОЛЬЗОВАТЕЛЯ 28](#_Toc437209712)

[6.1 Проектирование интерфейса раздела пользователя 28](#_Toc437209713)

[6.2 Разработка программных модулей раздела пользователя 32](#_Toc437209714)

[7. ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО САЙТА 39](#_Toc437209715)

[7.1 Анализ кроссбраузерности сайта 39](#_Toc437209716)

[7.2 Профилирование разработанного сайта 41](#_Toc437209717)

[7.3 Тестовые примеры работы 42](#_Toc437209718)

[ЗАКЛЮЧЕНИЕ 50](#_Toc437209719)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 51](#_Toc437209720)

[ПРИЛОЖЕНИЕ А 52](#_Toc437209721)

[ПРИЛОЖЕНИЕ Б 56](#_Toc437209722)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

КУРСОВОЙ ПРОЕКТ

Разраб.

Лисянский А. И.

Провер.

*Забаштанский А.К.*

Н. Контр.

Утв.

ПОЯСНИТЕЛЬНАЯ

ЗАПИСКА

Лит.

Листов

64

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 WEB-САЙТА
   1. Проектирование базы данных 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).

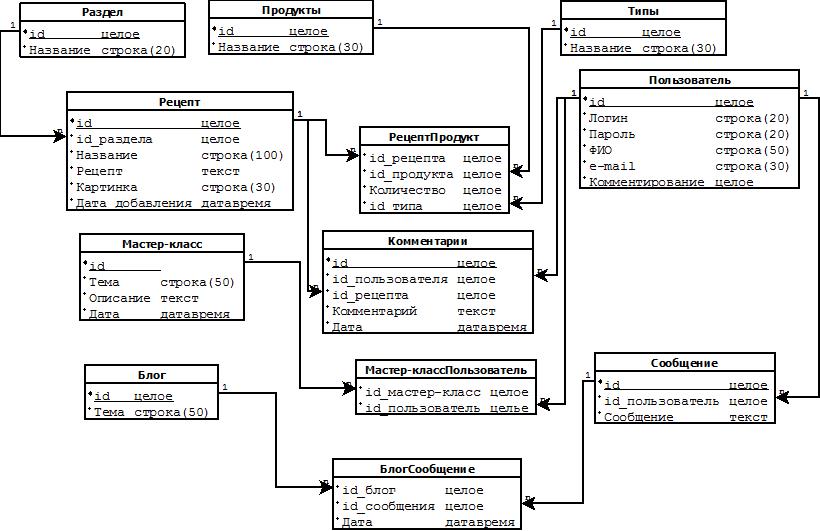


Рисунок 3.1 – Логическая схема базы данных

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

* 1. Проектирование сайта

На основании технического задания была разработана концептуальная модель сайта, содержащая страницы раздела администратора и клиента. Для дальнейшей работы были описаны контроллеры для страниц и функционирования сайта. На рисунке ниже представлены контролеры с их взаимосвязями.

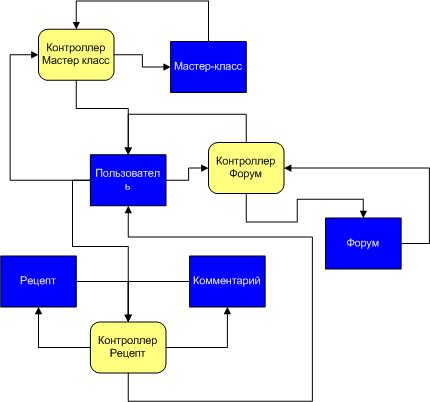


Рисунок 3.2 – Схема взаимодействия контроллеров с моделями

Так как на сайте существуют два независимых раздела со своим функционалом, продемонстрируем переход между разделами.

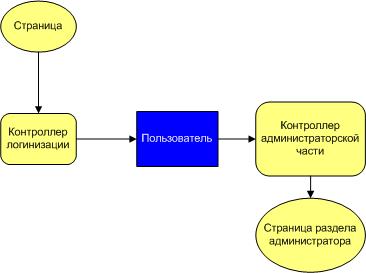


Рисунок 3.3 – Схема перехода между администраторской и клиентской частями

При разработке сайта был тщательно спроектирован взаимосвязи между основными контроллерами, дописаны и отлажены промежуточные контроллеры.

Проектирование сайта заняло половину времени разработки сайта, так как является важнейшей частью разработки любого ПО. Анализируя поставленную задачу, описанную в ТЗ, были получены основные контроллеры разделов администратора и клиента, спроектированы, написаны и отлажены переходные контроллеры.

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.

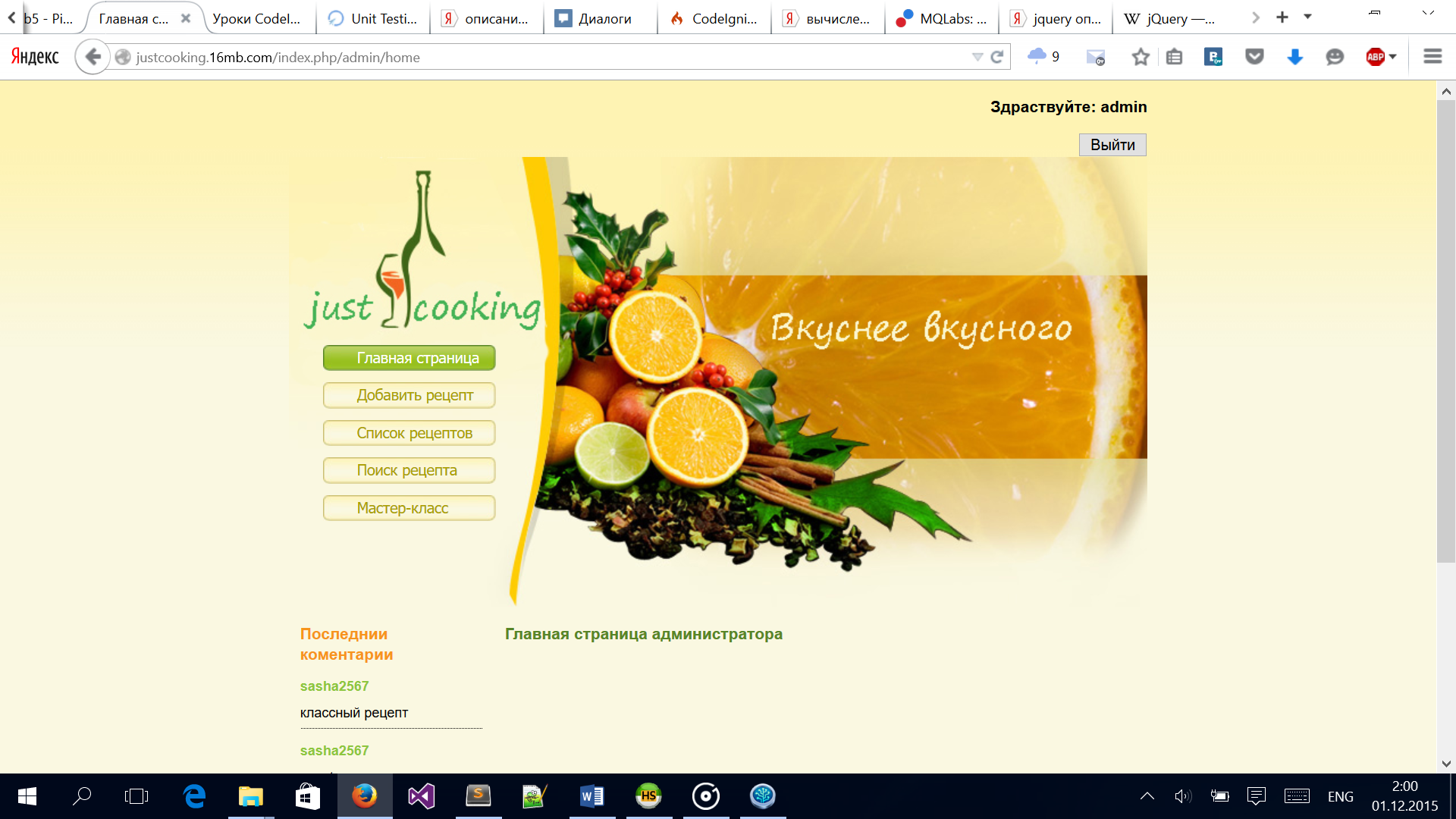


Рисунок 5.1 – Меню раздела администратора

Код отображающий меню раздела администратора представлен ниже.

<div id="templatemo\_menu">

<ul>

<li>

<a href="<?=base\_url();?>admin/admin\_home"

<?php if($pageIndex == 0)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Главная страница'; else echo 'Main page';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>admin/admin\_search/index"

<?php if($pageIndex == 1)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Поиск рецептов'; else echo 'Search recipe';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>admin/admin\_recipe/showadd"

<?php if($pageIndex == 2)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Добавление рецепта'; else echo 'Add recipe';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>admin/admin\_home/listrecipe"

<?php if($pageIndex == 3)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Список рецептов'; else echo 'List of recipes';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>admin/admin\_master"

<?php if($pageIndex == 4)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Мастер-класс'; else echo 'Master-class';?>

</a>

</li>

</ul>

</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 addProduct($data)

{

$this->db->insert($this->product\_table, $data);

$id = null;

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

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

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

$id = $query[0][$this->product\_id];

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 = 0){

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

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

}

function deleteIngredients($id = 1)

{

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

$this->db->delete($this->recipe\_product\_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();

$tmp = $tmp[0];

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

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

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

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

$tmpp = $tmpp[0];

$result[] = array(

'product\_id' => $tmp['product\_id'],

'product' => $tmp['description'],

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

'type' => $tmpp['type'],

'type\_id' => $tmpp['type\_id']

);

}

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();

}

function getnews($id = 1)

{

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

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

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

return $query->result\_array();

}

function getdaily($id = 1)

{

$this->db->where($this->section\_id, 2);

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

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

return $query->result\_array();

}

function dailylist()

{

$this->db->where($this->section\_id, 2);

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

return $query->result\_array();

}

function getfeast($id = 1)

{

$this->db->where($this->section\_id, 1);

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

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

return $query->result\_array();

}

function feastlist()

{

$this->db->where($this->section\_id, 1);

$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 Master 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($id,$userid)

{

$data = array(

'master\_id' => $id,

'user\_id' => $userid

);

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

}

/\*\*

\* add data

\*/

function deleteuser($id, $masterid)

{

$data = array(

$this->key\_id => $masterid,

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

);

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

}

function deleteMaster($id = 1)

{

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

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

}

/\*\*

\* get info

\*/

function getUser($masterid)

{

$this->db->join($this->table\_master, 'user.user\_id = master\_user.user\_id');

$this->db->where('master\_user.master\_id', $masterid);

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

return $query->result\_array();

}

function getmaster($masterid)

{

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

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

return $query->result\_array();

}

/\*\*

\* get list

\*/

function getlist(){

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

$this->db->where('provide', 0);

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

return $query->result\_array();

}

function validateUser($userid)

{

$id = $this->getlist();

$id = $id[0];

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

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

$this->db->where($this->key\_id, $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 внутренних средств фреймворка были написаны тесты моделей. Код теста для всех моделей представлен ниже. В коде продемонстрированы тесты некоторых функций модели, так как функций у всех моделей очень много, а написание тестов занимает длительное время.

<?php

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

class Tests extends CI\_Controller {

public function index()

{

$this->load->library('unit\_test');

$this->unit->run($this->load->model('coment'), 'is\_object', 'load model "coment"');

$this->unit->run($this->load->model('user'), 'is\_object', 'load model "user"');

$this->unit->run($this->load->model('recipe'), 'is\_object', 'load model "recipe"');

$this->unit->run($this->load->model('masters'), 'is\_object', 'load model "masters"');

$this->unit->run($this->load->model('forum'), 'is\_object', 'load model "forum"');

$m = $this->coment->getlist();

$this->unit->run($m, 'is\_array', 'model "coment" function "getlist"');

$m = $this->coment->getrecipe(1);

$this->unit->run($m, 'is\_array', 'model "coment" function "getrecipe"');

$m = $this->coment->getuser(1);

$this->unit->run($m, 'is\_array', 'model "coment" function "getuser"');

$m = $this->forum->getmessege(1);

$this->unit->run($m, 'is\_array', 'model "forum" function "getmessege"');

$m = $this->forum->gettheme();

$this->unit->run($m, 'is\_array', 'model "forum" function "gettheme"');

$m = $this->user->getUser('admin');

$this->unit->run($m, 'is\_array', 'model "user" function "getUser"');

$m = $this->user->getlist();

$this->unit->run($m, 'is\_array', 'model "user" function "getlist"');

$m = $this->recipe->getproduct(1);

$this->unit->run($m, 'is\_array', 'model "recipe" function "getUser"');

$m = $this->recipe->getrecipeid(array(1,2,3));

$this->unit->run($m, 'is\_array', 'model "recipe" function "getlist"');

$m = $this->masters->getUser(1);

$this->unit->run($m, 'is\_array', 'model "masters" function "getUser"');

$m = $this->masters->getlist();

$this->unit->run($m, 'is\_array', 'model "masters" function "getlist"');

$this->unit->set\_test\_items(array('test\_name', 'result'));

echo $this->unit->report();

}

}

На рисунке 5.2 представлены результаты выполнения тестов.

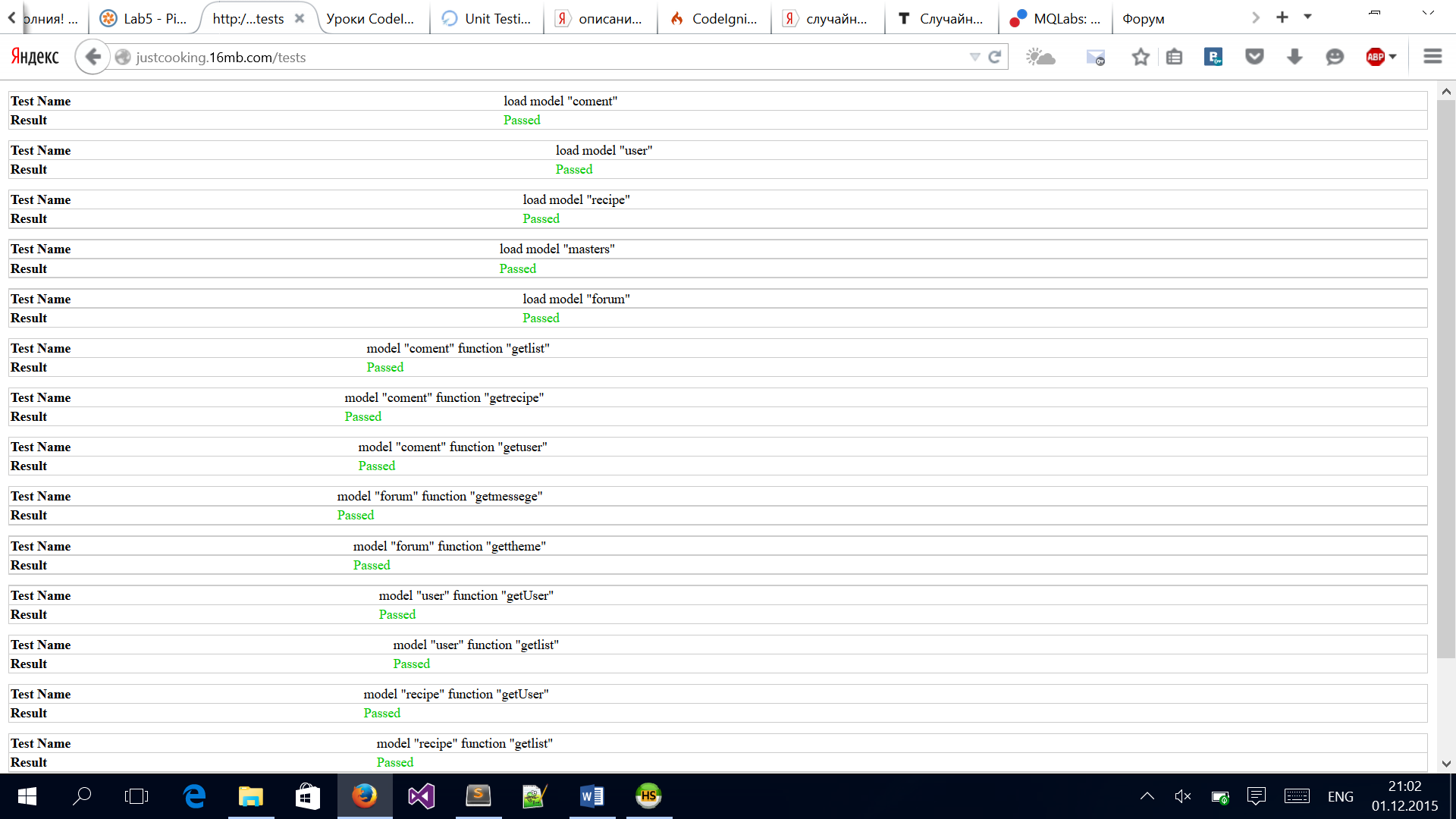


Рисунок 5.2 – Результаты выполнения тестов

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.

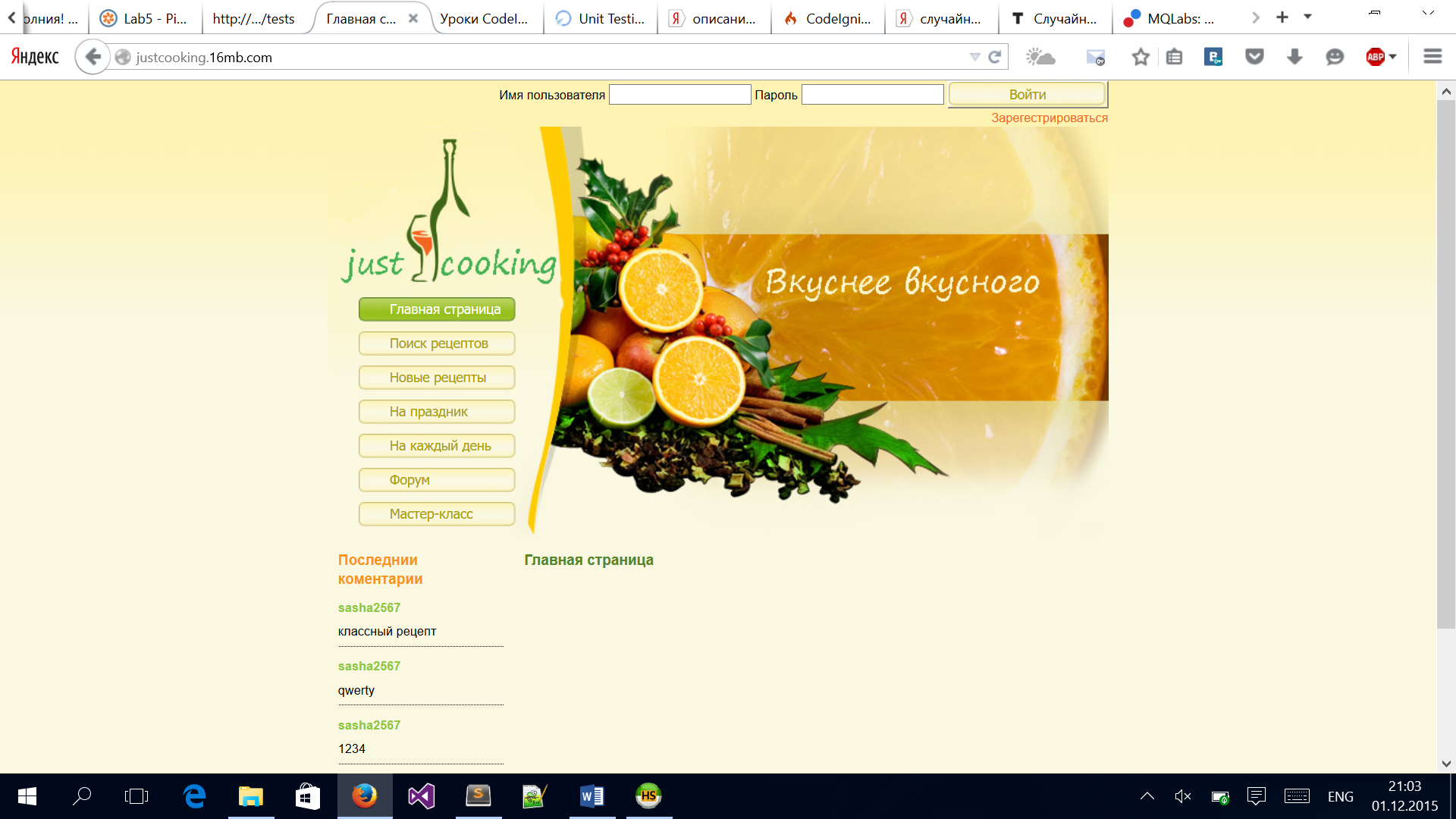


Рисунок 6.1 – Меню для неавторизованного пользователя

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.

<?php defined('BASEPATH') OR exit('No direct script access allowed');?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>

<?=$title;?>

</title>

<meta name="keywords" content="free design template, download web templates, Fruit And Juice Website, XHTML, CSS" />

<meta name="description" content="Fruit And Juice - Free CSS Template, Free XHTML CSS Design Layout" />

<link href="<?=base\_url();?>templatemo\_style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="<?=base\_url();?>highslide/highslide.css" />

<script type="text/javascript" src="<?=base\_url();?>highslide/highslide-with-gallery.js"></script>

<script type="text/javascript" src="<?=base\_url();?>highslide/jquery.js"></script>

<script type="text/javascript">

hs.graphicsDir = 'highslide/graphics/';

hs.align = 'center';

hs.transitions = ['expand', 'crossfade'];

hs.wrapperClassName = 'dark borderless floating-caption';

hs.fadeInOut = true;

hs.dimmingOpacity = .75;

if (hs.addSlideshow) hs.addSlideshow({

interval: 5000,

repeat: false,

useControls: true,

fixedControls: 'fit',

overlayOptions: {

opacity: .6,

position: 'bottom center',

hideOnMouseOut: true

}

});

</script>

</head>

<body>

<div id="templatemo\_container">

<div id="lang\_div">

<a href="<?=base\_url();?>language/index/1">rus</a>

<a href="<?=base\_url();?>language/index/2">eng</a>

</div>

<div id="autorise">

<?php

if(!isset($username) || $username === FALSE){

$attributes = array(

'class' => 'user\_login',

'id' => 'form\_user\_login'

);

echo form\_open('login/logined', $attributes);

?>

<label><?php if($lang == 'rus') echo "Имя пользователя"; else echo "User name";?></label>

<input type="text" name="username" />

<label><?php if($lang == 'rus') echo "Пароль"; else echo "Password";?></label>

<input type="password" name="user\_password" />

<input type="submit" name="user\_login\_btn" id="btn\_log" value="<?php if($lang == 'rus') echo 'Войти'; else echo 'Login';?>" />

<?php

echo form\_close();

?>

<div id="registr">

<a href="<?=base\_url();?>registration"><?php if($lang == 'rus') echo "Зарегестрируются"; else echo "Registrated";?></a>

</div>

<?php

}

else

{

?>

<h3><?php if($lang == 'rus') echo 'Здраствуйте'; else echo 'Hello';?>: <?=$username;?></h3>

<?php

$attributes = array(

'class' => 'user\_login',

'id' => 'form\_user\_login'

);

echo form\_open('login/exitlog', $attributes);

?>

<input type="submit" name="user\_login\_btn" id="btn\_log" value="<?php if($lang == 'rus') echo 'Выйти'; else echo 'Logout';?>" />

<?php

echo form\_close();

}

?>

</div>

<div id="templatemo\_header">

<div id="templatemo\_menu">

<ul>

<li>

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

<?php if($pageIndex == 0)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Главная страница'; else echo 'Main page';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>user/user\_search/index"

<?php if($pageIndex == 1)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Поиск рецептов'; else echo 'Search recipe';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>user/user\_home/news"

<?php if($pageIndex == 2)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Новые рецепты'; else echo 'New recipe';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>user/user\_home/feast"

<?php if($pageIndex == 3)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Праздничные блюда'; else echo 'Holiday recipes';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>user/user\_home/daily"

<?php if($pageIndex == 4)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Ежедневные блюда'; else echo 'Everyday recipes';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>forum/home"

<?php if($pageIndex == 5)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Форум'; else echo 'Forum';?>

</a>

</li>

<li>

<a href="<?=base\_url();?>user/user\_master/show"

<?php if($pageIndex == 6)

echo 'class="current"';

?>

>

<?php if($lang == 'rus') echo 'Мастер-класс'; else echo 'Master-class';?>

</a>

</li>

</ul>

</div>

</div>

<div id="templatemo\_content\_area">

<div id="templatemo\_left\_col">

<div class="templatemo\_section">

<h1><?php if($lang == 'rus') echo 'Последнии коментарии'; else echo 'Last coments';?></h1>

<?php

if(isset($titlecoment)){

foreach ($titlecoment as $value) :?>

<h2><?=$value['user']['username'];?></h2>

<p><?=$value['coment']['coment'];?></p>

<?php

endforeach;

}

?>

</div>

<div class="templatemo\_section">

<h1>W3C Validators</h1>

<p>

<a href="http://validator.w3.org/check?uri=referer"><img src="http://www.w3.org/Icons/valid-xhtml10" alt="Valid XHTML 1.0 Transitional" width="88" height="31" border="0" /></a>

<a href="http://jigsaw.w3.org/css-validator/check/referer"><img style="border:0;width:88px;height:31px" alt="Valid CSS" src="http://jigsaw.w3.org/css-validator/images/vcss-blue" /></a>

</p>

</div>

</div><!-- End Of Left -->

* 1. Разработка программных модулей раздела пользователя

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

Для реализации возможности просмотра списка товаров на соответствующей странице было реализовано представление Home.php. Код представления представлен ниже.

<?php defined('BASEPATH') OR exit('No direct script access allowed');?>

<div id="templatemo\_right\_col">

<div id="container">

<h1 id="headerform"><?php if($lang == 'rus') echo 'Рецепты'; else echo 'Recipes';?></h1>

<div id="body">

<ul id="recipes">

<?php foreach ($recipes as $item):?>

<li>

<a href="<?=base\_url();?>user/user\_recipe/show/<?=$item['recipe\_id']?>">

<img src="<?=base\_url();?>images/<?=$item['image']?>" height="150" width="150"/>

</a>

&nbsp;

<a href="<?=base\_url();?>user/user\_recipe/show/<?=$item['recipe\_id']?>">

<?=$item['description']?>

</a>

</li>

<?php endforeach; ?>

</ul>

<div id="pagination">

<?=$pagination;?>

</div>

</div>

</div>

Для просмотра рецепта на странице «Рецепт» было реализовано представление Recipe.php. Код представления представлен ниже.

<?php defined('BASEPATH') OR exit('No direct script access allowed');?>

<div id="templatemo\_right\_col">

<div id="container">

<h1 id="headerform"><?=$item['description']?></h1>

<div id="body">

<div clacc="image">

<img src="<?=base\_url();?>images/<?=$item['image']?>" height="150" width="150"/>

</div>

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Продукты:'; else echo 'Products:';?></h3>

<div id="products">

<?php

foreach ($products as $value) {

?>

<h3><?=$value['product']." : ".$value['count']." ".$value['type']?></h3>

<?php

}

?>

</div>

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Рецепт:'; else echo 'Recipe:';?></h3>

<p class="recipe">

<?=$item['recipe']?>

</p>

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Коментарии:'; else echo 'Coments:';?></h3>

<div id="coments">

<?php

foreach ($coments as $value) {

?>

<div id="coment">

<h3>Пользователь : <?=$value->user['username'];?></h3>

<h4><?=$value->coment['datetime']?></h4>

<p><?=$value->coment['coment']?></p>

</div>

<?php

}

?>

</div>

<?php

if(isset($username) && $username !== FALSE && $comented == 1){

?>

<div id="coment\_form">

<?php

$attributes = array(

'class' => 'user\_login',

'id' => 'form\_add\_coment'

);

echo form\_open('user/user\_recipe/addcoment/'.$item['recipe\_id'], $attributes);

?>

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Добавьте свой коментарий:'; else echo 'Add your comment:';?></h3>

<textarea rows="5" cols="63" name="coment\_user"></textarea>

<div id="submit">

<input type="submit" name="add\_coment\_user" value="<?php if($lang == 'rus') echo 'Добавить коментарий'; else echo 'Add coment';?>" />

</div>

<?php

echo form\_close();

echo validation\_errors();

}

?>

</div>

</div>

Для просмотра страниц «Форум» и «Мастер-класс» реализованы представления Master.php и Forum.php. Код представления представлен ниже.

<?php defined('BASEPATH') OR exit('No direct script access allowed');?>

<div id="templatemo\_right\_col">

<?php

if(isset($masters)){

foreach ($masters as $value) {

?>

<div id="container">

<div id="body">

<div id="master">

<h1 id="headerform"><?php if($lang == 'rus') echo 'Тема'; else echo 'Subject';?>: <?=$value['subject']?></h1>

<h5><?php if($lang == 'rus') echo 'Дата проведения'; else echo 'Provide date';?>: <?=$value['dateprovide']?></h5>

<p>

<?php if($lang == 'rus') echo 'Описание'; else echo 'Description';?>: <?=$value['description']?>

</p>

</div>

<?php

if(isset($username) && $username !== FALSE){

$attributes = array(

'class' => 'user\_master',

'id' => 'form\_master\_login'

);

echo form\_open('user/user\_master/write/'.$value["master\_id"], $attributes);

?>

<div id="submit">

<input type="submit" name="user\_master\_btn" value="<?php if($lang == 'rus') echo 'Записаться'; else echo 'Follow';?>" />

</div>

<?php

echo form\_close();

}

?>

</div>

</div>

<?php

}

//die();

?>

<?php

}

else{

?>

<div id="container">

<div id="body">

<?php if($lang == 'rus') echo 'Сейчас нет ни одного мастер-класса'; else echo 'Nothind master-classes';?>

</div>

</div>

<?php

}

?>

<?php defined('BASEPATH') OR exit('No direct script access allowed');?>

<div id="templatemo\_right\_col">

<div id="container">

<h1 id="headerform"><?=$item['description']?></h1>

<div id="body">

<div clacc="image">

<img src="<?=base\_url();?>images/<?=$item['image']?>" height="150" width="150"/>

</div>

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Продукты:'; else echo 'Products:';?></h3>

<div id="products">

<?php

foreach ($products as $value) {

?>

<h3><?=$value['product']." : ".$value['count']." ".$value['type']?></h3>

<?php

}

?>

</div>

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Рецепт:'; else echo 'Recipe:';?></h3>

<p class="recipe">

<?=$item['recipe']?>

</p>

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Коментарии:'; else echo 'Coments:';?></h3>

<div id="coments">

<?php

foreach ($coments as $value) {

?>

<div id="coment">

<h3>Пользователь : <?=$value->user['username'];?></h3>

<h4><?=$value->coment['datetime']?></h4>

<p><?=$value->coment['coment']?></p>

</div>

<?php

}

?>

</div>

<?php

if(isset($username) && $username !== FALSE && $comented == 1){

?>

<div id="coment\_form">

<?php

$attributes = array(

'class' => 'user\_login',

'id' => 'form\_add\_coment'

);

echo form\_open('user/user\_recipe/addcoment/'.$item['recipe\_id'], $attributes);

?>

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Добавьте свой коментарий:'; else echo 'Add your comment:';?></h3>

<textarea rows="5" cols="63" name="coment\_user"></textarea>

<div id="submit">

<input type="submit" name="add\_coment\_user" value="<?php if($lang == 'rus') echo 'Добавить коментарий'; else echo 'Add coment';?>" />

</div>

<?php

echo form\_close();

echo validation\_errors();

}

?>

</div>

</div>

<div id="templatemo\_right\_col">

<div id="container">

<h1 id="headerform"><?php if($lang == 'rus') echo 'Тема: '; else echo 'Theme: ';?><?=$themeName?></h1>

<div id="body">

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Сообщения:'; else echo 'Messages';?></h3>

<ul id="themes">

<?php foreach ($messeges as $item):?>

<li>

<div id="message">

<p>

<?=$item['username']?>

</p>

<h3><?=$item['messege']?></h3>

</div>

</li>

<?php endforeach; ?>

</ul>

</div>

</div>

<?php

if($username != false){?>

<div id="container">

<h3 id="headerformh3"><?php if($lang == 'rus') echo 'Добавить сообщение:'; else echo 'Add message';?></h3>

<div id="body">

<?php

$attributes = array(

'class' => 'user\_messege',

'id' => 'form\_forum\_messege'

);

echo form\_open('forum/home/addmessege/'.$themeId, $attributes);

?>

<input type="text" name="user\_messege">

<div id="submit">

<input type="submit" name="user\_messege\_btn" value="<?php if($lang == 'rus') echo 'Написать сообщение'; else echo 'Send message';?>" />

</div>

<?php

echo form\_close();

echo validation\_errors();?>

</div>

</div>

<?php

}

?>

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

<div id="templatemo\_right\_col">

<div class="templatemo\_post\_area">

<h1>Регистрация</h1>

<?php

if(!isset($username) || $username === FALSE){

$attributes = array(

'class' => 'user\_registr',

'id' => 'form\_user\_registr'

);

echo form\_open('registration/registrated', $attributes);

?>

<label>Имя пользователя/Логин</label>

<input type="text" name="user\_name" />

<br />

<label>Пароль</label>

<input type="password" name="user\_password1" />

<br />

<label>Повторите пароль</label>

<input type="password" name="user\_password2" />

<br />

<label>ФИО</label>

<input type="text" name="user\_fio" />

<br />

<label>E-mail</label>

<input type="text" name="user\_mail" />

<br />

<input type="submit" name="user\_registr\_btn" id="btn\_reg" value="Зарегестрироваться" />

<?php

echo form\_close();

}

else{

?>

<p>Вы уже зарегестрированны</p>

<?php

}

?>

</div>

1. ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО САЙТА
   1. Анализ кроссбраузерности сайта

Проверка кроссбраузерности сайта в браузерах Google Chrome, Mozilla Firefox, Opera выполнялась в встроенном сервисе в Micrsoft Edge. Далее будут представлены результаты тестирования кроссбраузерности.

Отображение главной страницы в Google Chrome 11 показано на рисунке 7.1.

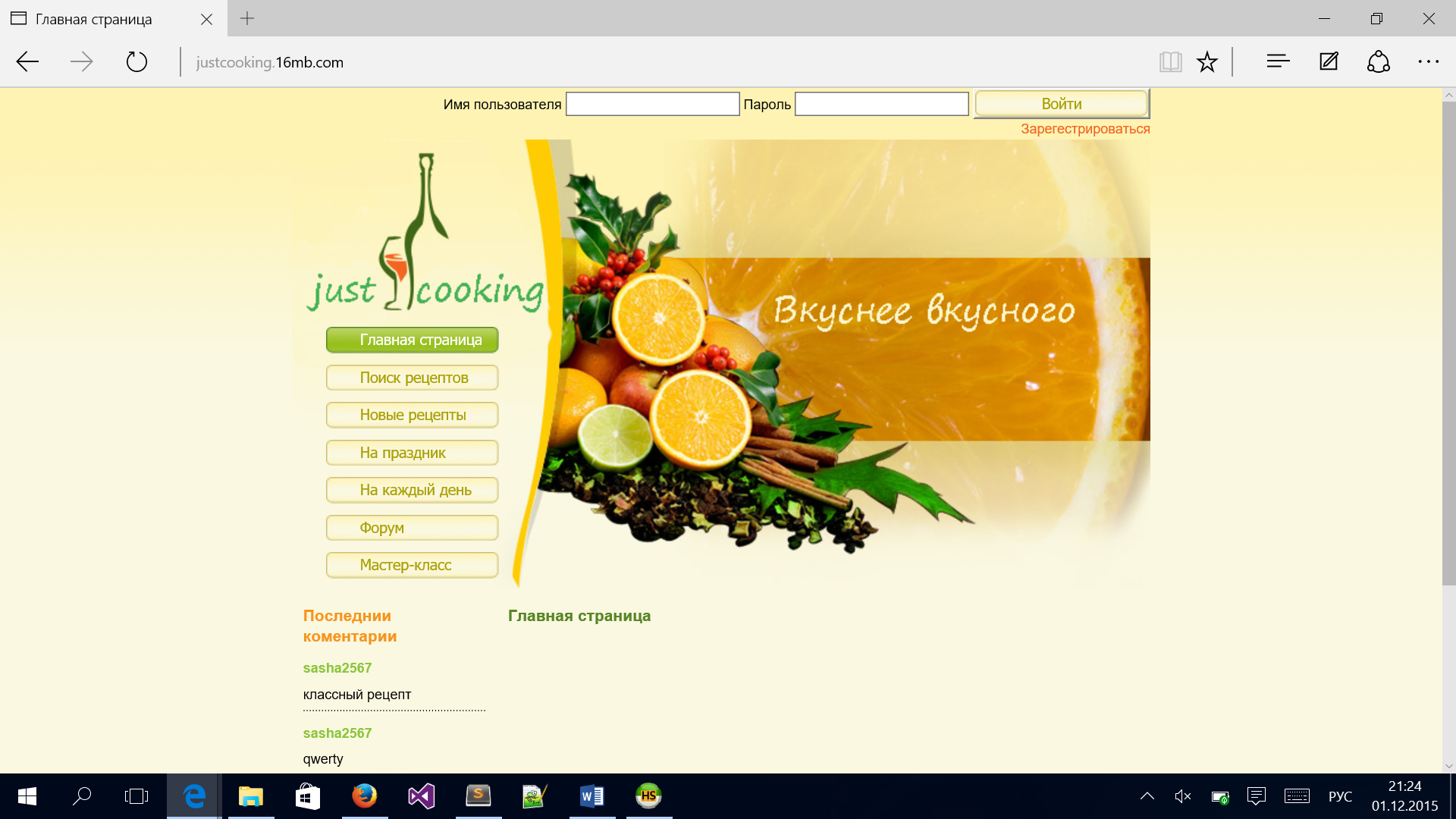


Рисунок 7.1 – Отображение страниц в Google Chrome 11

На рисунке 7.2 показано отображение страниц в браузере Firefox 38.0.

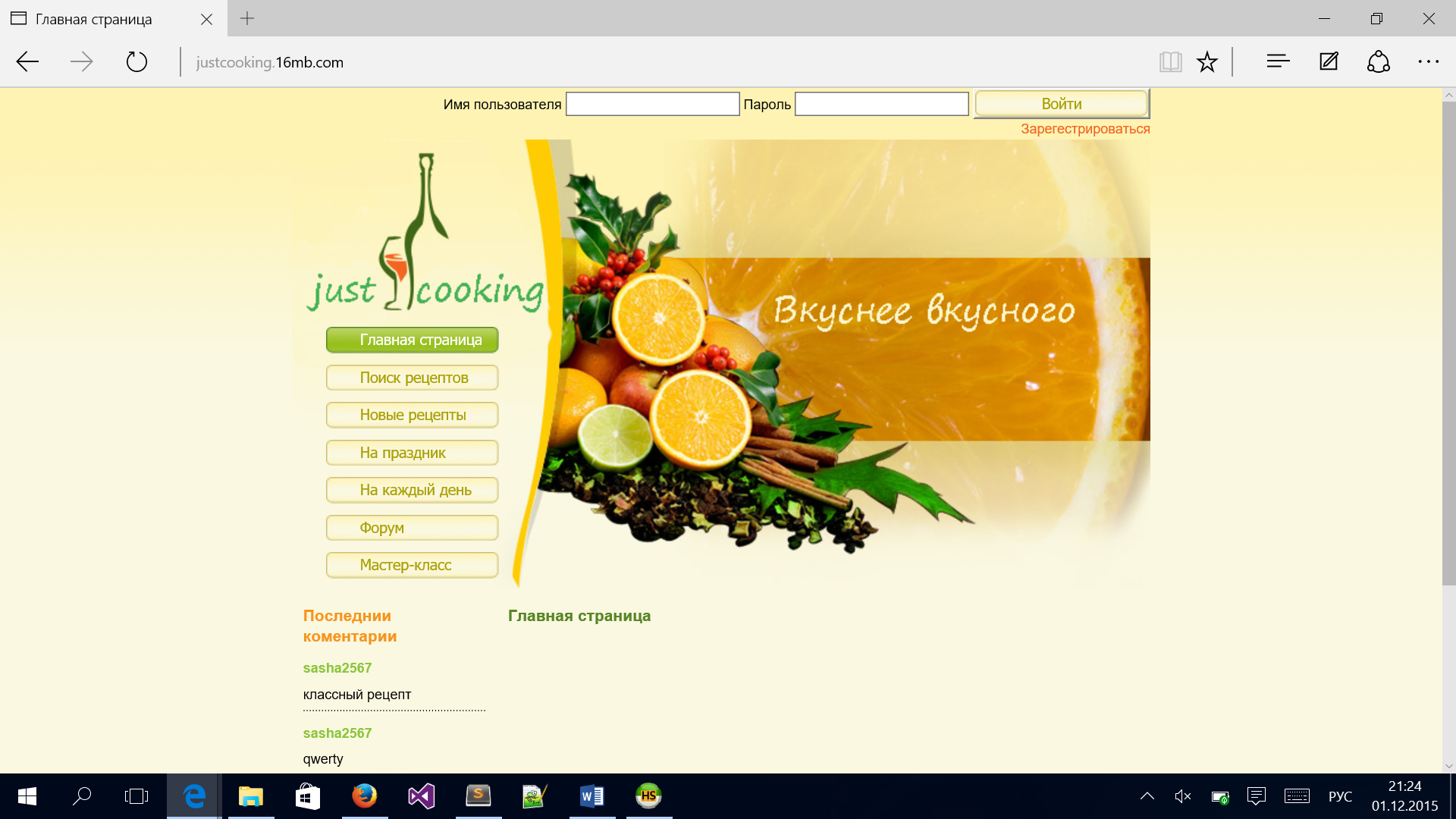


Рисунок 7.2 – Отображение страниц в Firefox 38.0

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

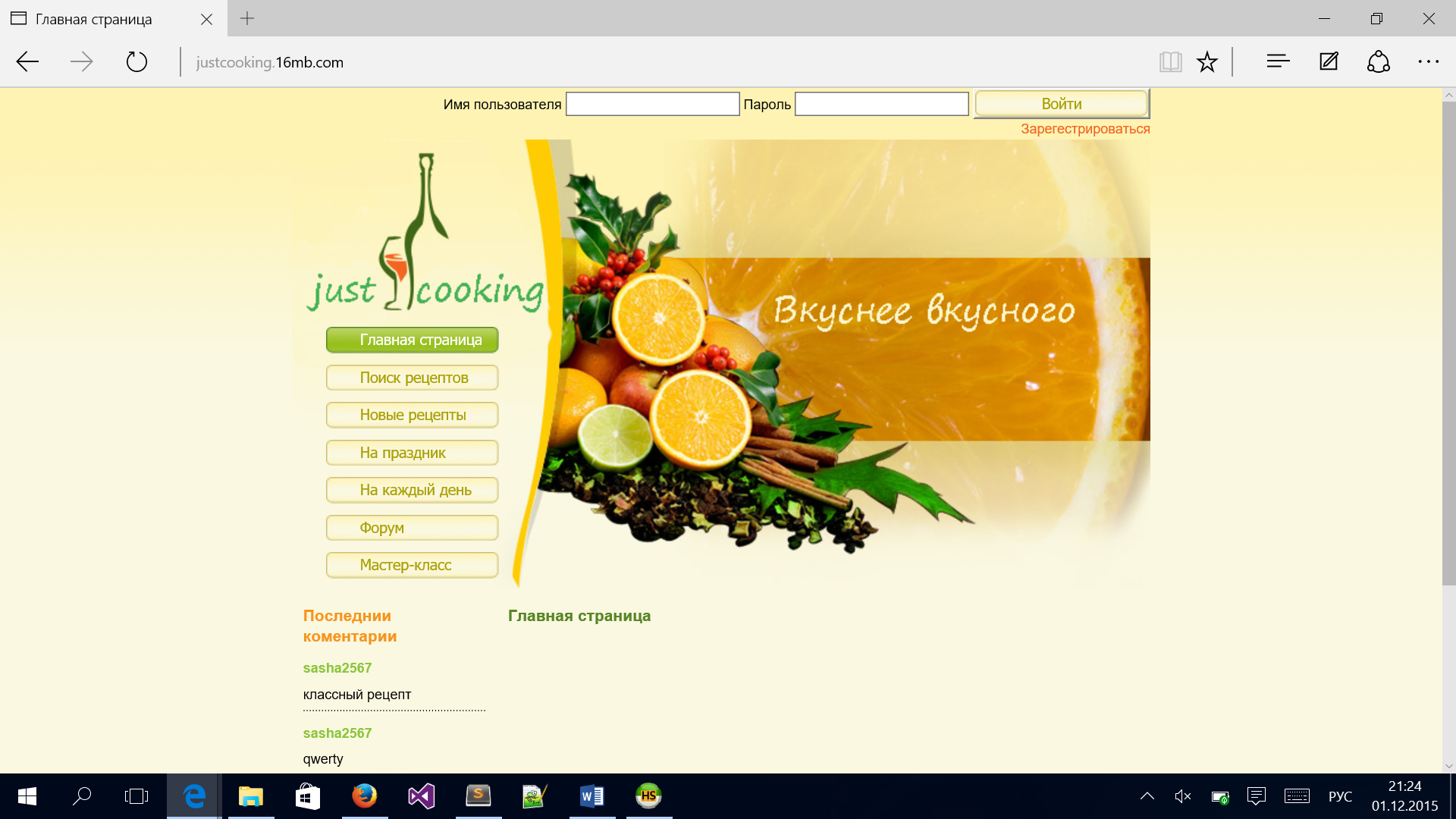


Рисунок 7.3 – Отображение главной страницы в Opera 12.2

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

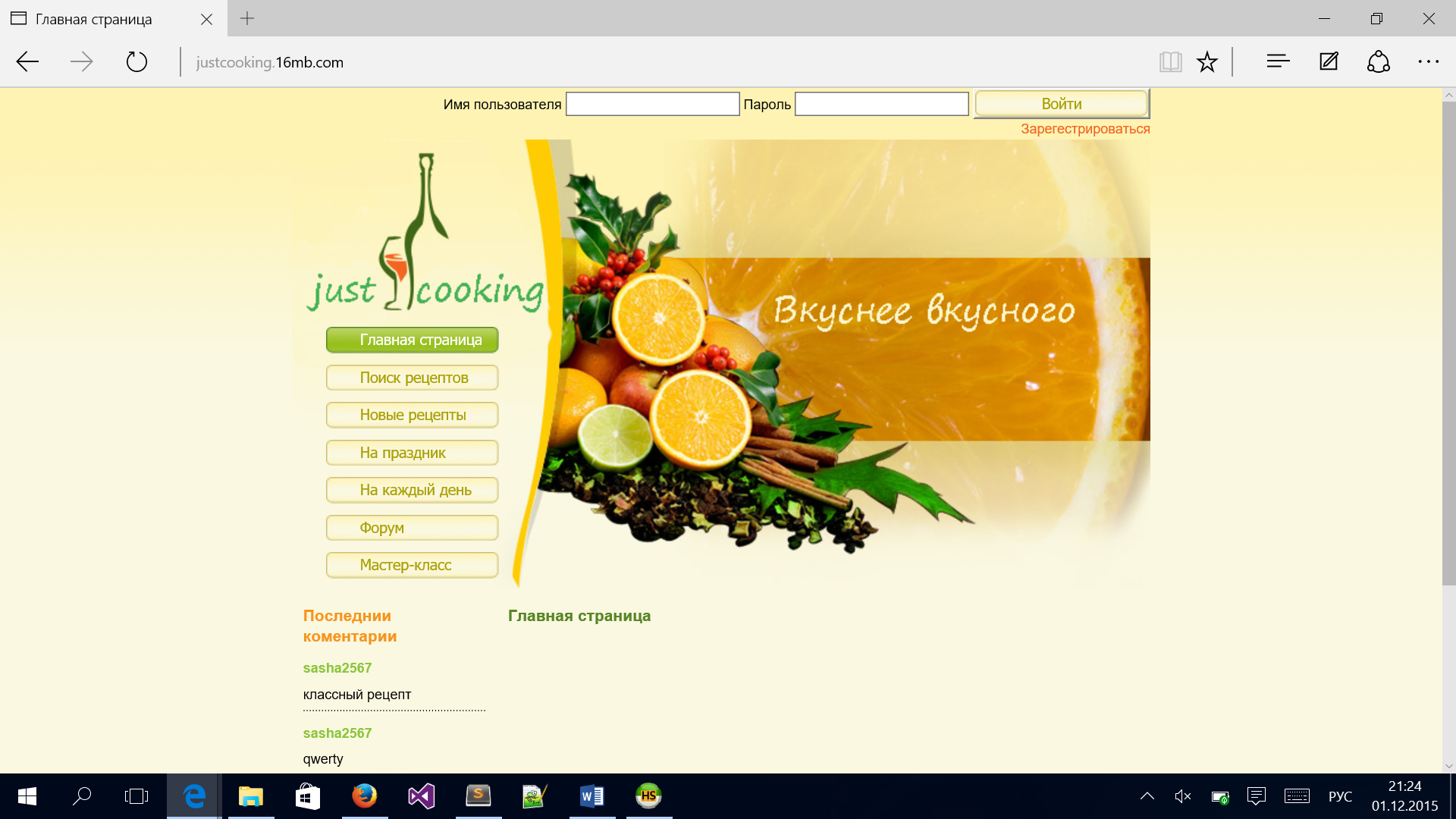


Рисунок 7.4 – Отображение страниц в Internet Explorer 8

Интерфейс разработанного сайта отображается в различных браузерах идентично.

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

* 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 | 1180 | 5,6 |
| Поиск рецепта(1) | 1140 | 2,9 |
| Поиск рецепта(2) | 1190 | 3,5 |
| Рецепт | 1210 | 3,6 |
| Форум | 1200 | 2,9 |
| Тема форума | 1280 | 4,9 |
| Мастер-класс | 1200 | 3,0 |

* 1. Тестовые примеры работы

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

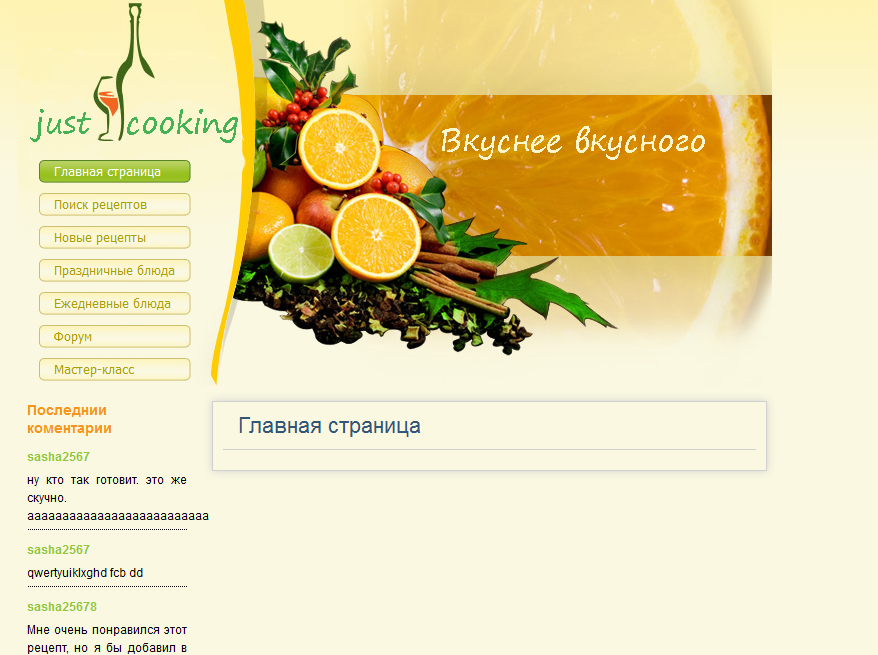


Рисунок 7.9 – Главная страница сайта

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

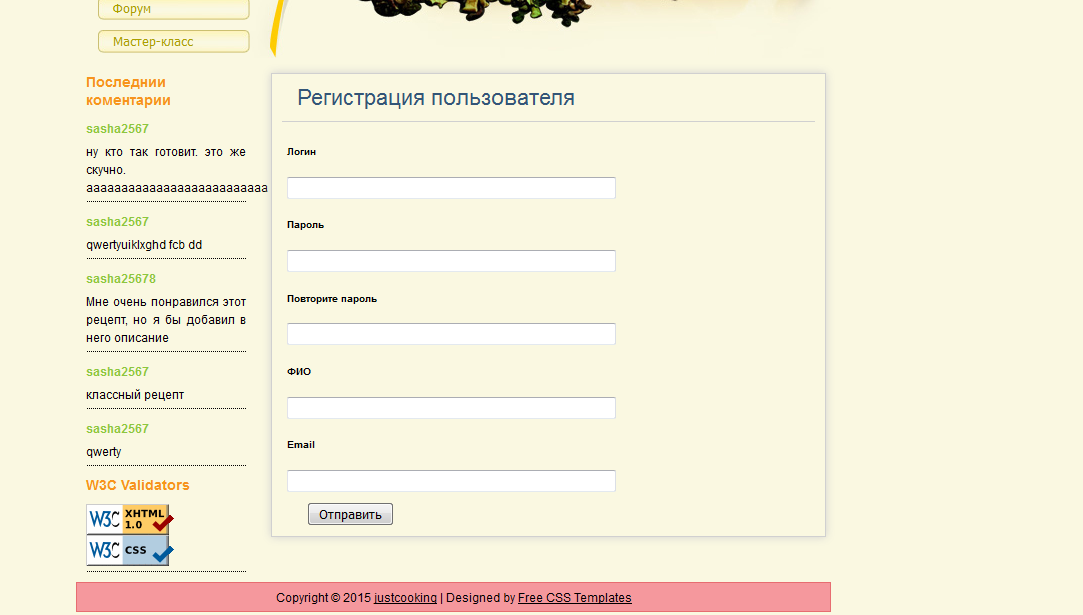


Рисунок 7.10 – Страница «Регистрация»

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.

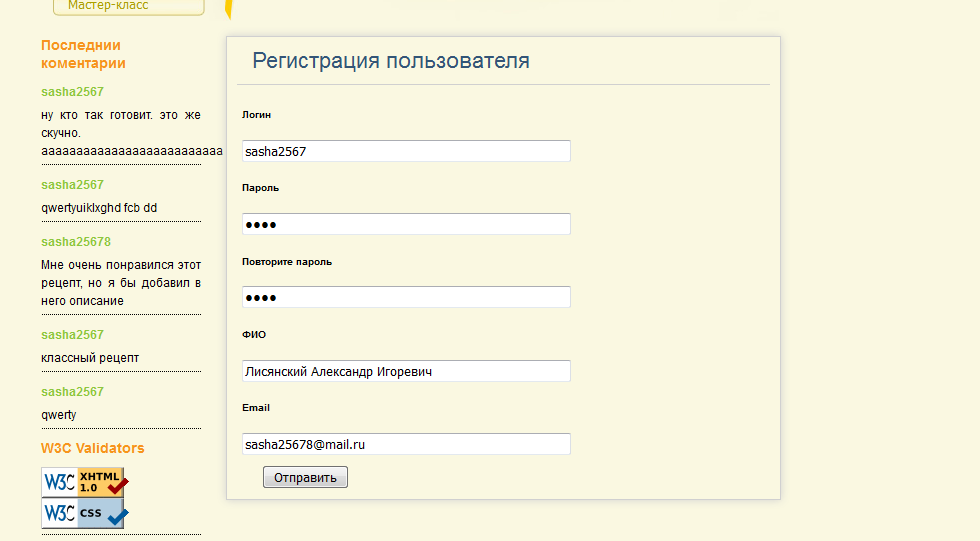


Рисунок 7.11 – Пример верного заполнения полей

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

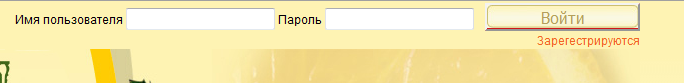
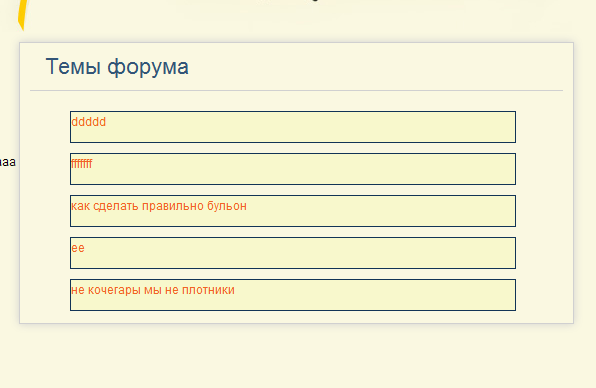


Рисунок 7.12 – Страница «Вход»

Выполнив вход в систему под своим login и паролем можно добавлять комментарии на странице конкретного рецепта, добавлять темы в разделе форум и добавлять сообщения к теме на форуме. При открытии страницы в поля ввода добавляется имеющаяся информация о пользователе. Новые данные проверяются точно также, как и при регистрации.



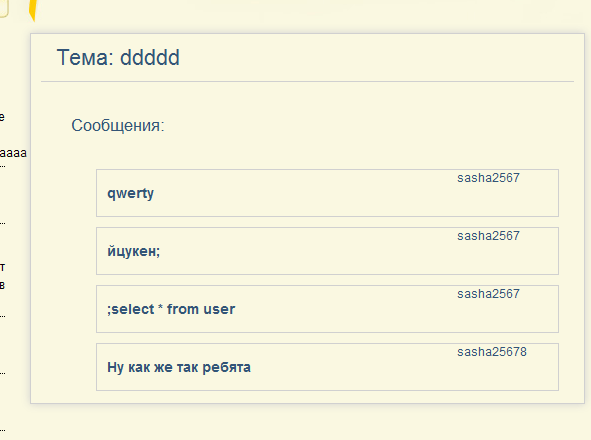


Рисунок 7.13 – Раздел «Форум»

При переходе на страницу «Мастер-класс» пользователь может просмотреть описание и тему предстоящего мастер-класса и записаться на него. В случае отсутствия мастер-классов будет выведено соответствующее сообщение.

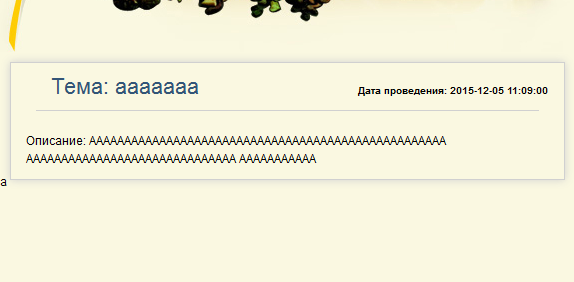


Рисунок 7.14 – Страница «Мастер-класс»

При нажатии на кнопку «Записаться», выполнится запись авторизированного пользователя на последний мастер-класс.

На сайте предусмотрено 3 типа поиска рецептов:

* по разделам;
* по названию;
* по ингредиентам.

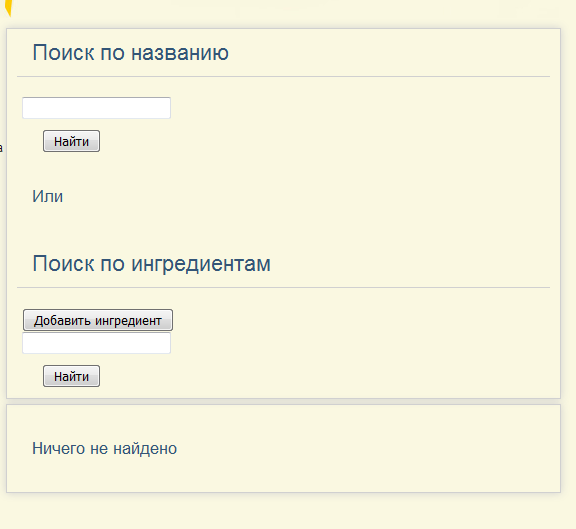


Рисунок 7.15 – Страница поиска по ингредиентам

При переходе на страницу «Новинки» отобразится список рецептов, которые представлены на сайте.

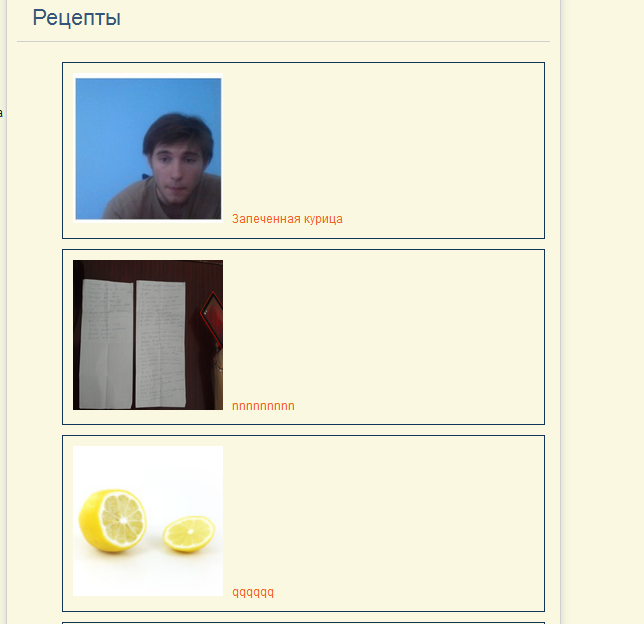


Рисунок 7.16 – Страница «Товары»

Для того чтобы просмотреть подробное описание рецепта необходимо нажать на картинку рецепта или на название. На странице с подробным описанием рецепта, также можно выполнить добавление авторизированным пользователем комментария.

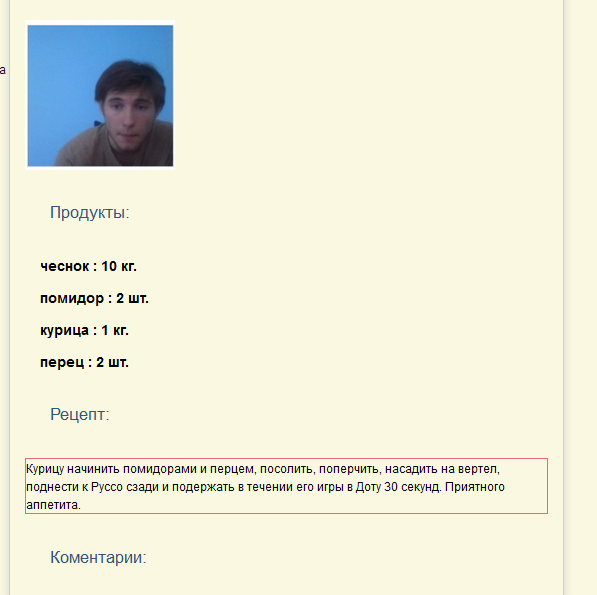


Рисунок 7.17 – Описание товара

После нажатия на кнопку «Добавить комментарий» комментарий пользователя добавляется в базу, и страница перезагружается уже с новым комментарием.

Выполним вход от имени администратора. Для этого выполним выход из сессии текущего пользователя и войдем как администратор. Администратор может добавлять, изменять мастер-классы, рецепты, запрещать или разрешать комментировать пользователю.

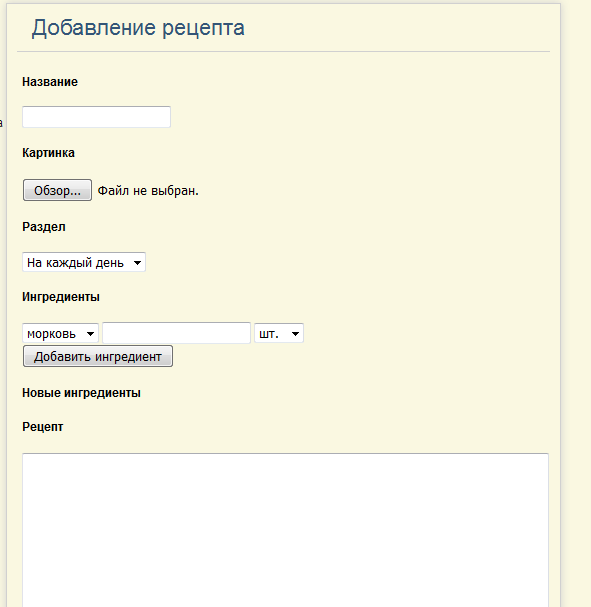


Рисунок 7.19 – Добавление администратором рецепта

Администратор также может удалять рецепты, отмечать проведенные мастер-классы, запрещать или разрешать комментировать пользователю, редактировать рецепты.

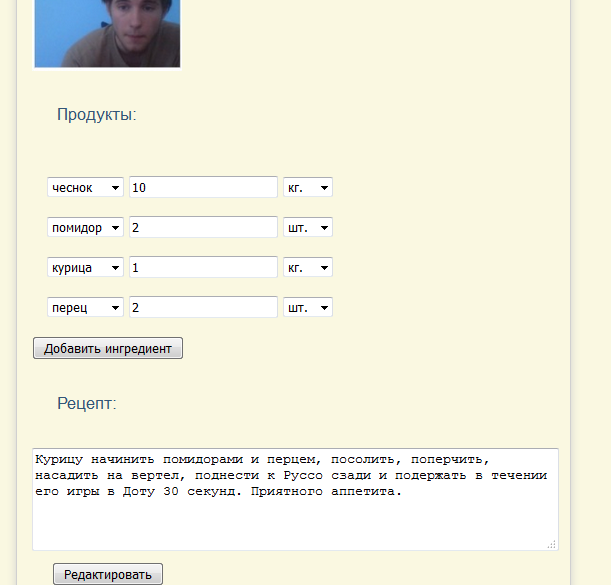


Рисунок 7.20 – Редактирование рецепта

# ЗАКЛЮЧЕНИЕ

* В данном курсовом проекте был реализован личный кулинарный сайт. 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: раздел; продукт; тип; рецепт; рецепт\_продукт; пользователь; комментарий; мастер-класс; мастер-класс\_пользователь; форум; сообщение; форум\_сообщение.

Интернет-магазин был разработан с использованием MVC фреймворка CodeIgniter. Помимо MVC фреймворка использовался CSS фреймворк Bootstrap. Разработанный сайт имеет разграничение прав доступа. Сайт размещен на бесплатном Web-хостинге hostinger.com.ua. Также написанный сайт был проверен на кроссбраузерность. Сайт 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 личный кулинарный сайт.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Полное руководство по CodeIgniter [Электронный ресурс]. — Режим доступа: http://cidocs.bericode.ru/. — Последний доступ: 21.10.2015. — 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. Краткое руководство по созданию сайта на CodeIgniter — Режим доступа: http://promo-creative.com/uroki-codeigniter/ . — Последний доступ: 25.10.2015. — Название с экрана.
4. Spurlock J. Bootstrap / J. Spurlock. — Sebastopol: O’Reilly Media, 2013. — 110 с.

# ПРИЛОЖЕНИЕ А

SQL для создания база данных сайта.

-- --------------------------------------------------------

-- Хост: 127.0.0.1

-- Версия сервера: 5.5.25 - MySQL Community Server (GPL)

-- ОС Сервера: Win32

-- HeidiSQL Версия: 9.3.0.4984

-- --------------------------------------------------------

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

-- Дамп структуры для таблица kursach\_web.web\_coment

CREATE TABLE IF NOT EXISTS `web\_coment` (

`coment\_id` int(11) NOT NULL AUTO\_INCREMENT,

`user\_id` int(11) NOT NULL,

`recipe\_id` int(11) NOT NULL,

`coment` mediumtext NOT NULL,

`datetime` datetime NOT NULL,

PRIMARY KEY (`coment\_id`),

KEY `FK\_web\_coment\_web\_user` (`user\_id`),

KEY `FK\_web\_coment\_web\_recipe` (`recipe\_id`),

CONSTRAINT `FK\_web\_coment\_web\_recipe` FOREIGN KEY (`recipe\_id`) REFERENCES `web\_recipe` (`recipe\_id`),

CONSTRAINT `FK\_web\_coment\_web\_user` FOREIGN KEY (`user\_id`) REFERENCES `web\_user` (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_master

CREATE TABLE IF NOT EXISTS `web\_master` (

`master\_id` int(11) NOT NULL AUTO\_INCREMENT,

`subject` varchar(100) NOT NULL,

`description` text NOT NULL,

`date` datetime NOT NULL,

PRIMARY KEY (`master\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_master\_user

CREATE TABLE IF NOT EXISTS `web\_master\_user` (

`master\_id` int(11) NOT NULL,

`user\_id` int(11) NOT NULL,

KEY `FK\_\_web\_master` (`master\_id`),

KEY `FK\_\_web\_user` (`user\_id`),

CONSTRAINT `FK\_\_web\_master` FOREIGN KEY (`master\_id`) REFERENCES `web\_master` (`master\_id`),

CONSTRAINT `FK\_\_web\_user` FOREIGN KEY (`user\_id`) REFERENCES `web\_user` (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_messege

CREATE TABLE IF NOT EXISTS `web\_messege` (

`messege\_id` int(11) NOT NULL,

`user\_id` int(11) NOT NULL,

`messege` text NOT NULL,

PRIMARY KEY (`messege\_id`),

KEY `FK\_web\_messeges\_web\_user` (`user\_id`),

CONSTRAINT `FK\_web\_messeges\_web\_user` FOREIGN KEY (`user\_id`) REFERENCES `web\_user` (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_products

CREATE TABLE IF NOT EXISTS `web\_products` (

`product\_id` int(11) NOT NULL AUTO\_INCREMENT,

`description` varchar(50) NOT NULL,

PRIMARY KEY (`product\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_recipe

CREATE TABLE IF NOT EXISTS `web\_recipe` (

`recipe\_id` int(11) NOT NULL AUTO\_INCREMENT,

`section\_id` int(11) NOT NULL,

`description` varchar(100) NOT NULL,

`recipe` text NOT NULL,

`image` varchar(50) NOT NULL,

`date` datetime NOT NULL,

PRIMARY KEY (`recipe\_id`),

KEY `FK\_web\_recipe\_recipe\_web\_section` (`section\_id`),

CONSTRAINT `FK\_web\_recipe\_recipe\_web\_section` FOREIGN KEY (`section\_id`) REFERENCES `web\_section` (`section\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_recipe\_product

CREATE TABLE IF NOT EXISTS `web\_recipe\_product` (

`recipe\_id` int(11) NOT NULL AUTO\_INCREMENT,

`product\_id` int(11) NOT NULL,

`count` int(11) NOT NULL,

`type\_id` int(11) NOT NULL,

KEY `FK\_web\_recipe\_web\_products` (`product\_id`),

KEY `FK\_web\_recipe\_web\_recipe\_recipe` (`recipe\_id`),

KEY `FK\_web\_recipe\_product\_web\_types` (`type\_id`),

CONSTRAINT `FK\_web\_recipe\_product\_web\_types` FOREIGN KEY (`type\_id`) REFERENCES `web\_types` (`type\_id`),

CONSTRAINT `FK\_web\_recipe\_web\_products` FOREIGN KEY (`product\_id`) REFERENCES `web\_products` (`product\_id`),

CONSTRAINT `FK\_web\_recipe\_web\_recipe\_recipe` FOREIGN KEY (`recipe\_id`) REFERENCES `web\_recipe` (`recipe\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_section

CREATE TABLE IF NOT EXISTS `web\_section` (

`section\_id` int(11) NOT NULL AUTO\_INCREMENT,

`title` varchar(50) NOT NULL,

PRIMARY KEY (`section\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_theme

CREATE TABLE IF NOT EXISTS `web\_theme` (

`theme\_id` int(11) NOT NULL AUTO\_INCREMENT,

`theme` varchar(50) NOT NULL,

PRIMARY KEY (`theme\_id`),

UNIQUE KEY `theme` (`theme`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_theme\_messege

CREATE TABLE IF NOT EXISTS `web\_theme\_messege` (

`theme\_id` int(11) NOT NULL,

`messege\_id` int(11) NOT NULL,

KEY `FK\_\_web\_theme` (`theme\_id`),

KEY `FK\_\_web\_messeges` (`messege\_id`),

CONSTRAINT `FK\_\_web\_messeges` FOREIGN KEY (`messege\_id`) REFERENCES `web\_messege` (`messege\_id`),

CONSTRAINT `FK\_\_web\_theme` FOREIGN KEY (`theme\_id`) REFERENCES `web\_theme` (`theme\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_types

CREATE TABLE IF NOT EXISTS `web\_types` (

`type\_id` int(11) NOT NULL AUTO\_INCREMENT,

`type` varchar(50) NOT NULL,

PRIMARY KEY (`type\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

-- Дамп структуры для таблица kursach\_web.web\_user

CREATE TABLE IF NOT EXISTS `web\_user` (

`user\_id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

`fio` varchar(50) NOT NULL,

`email` varchar(50) NOT NULL,

`is\_comented` int(11) NOT NULL,

PRIMARY KEY (`user\_id`),

UNIQUE KEY `username` (`username`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Экспортируемые данные не выделены.

/\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=IF(@OLD\_FOREIGN\_KEY\_CHECKS IS NULL, 1, @OLD\_FOREIGN\_KEY\_CHECKS) \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

# ПРИЛОЖЕНИЕ Б

Исходный код разработанных модулей

<?php

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

class Coments {

public $coment = null;

public $user = null;

function \_\_construct($c,$u){

$this->coment = $c;

$this->user = $u;

}

}

class Home extends CI\_Controller {

public function getTitleComent()

{

$this->load->model('coment');

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

$this->db->limit (5, 0);

$coment = $this->coment->getlist();

$this->load->model('user');

$users = array();

$coments = array();

foreach ($coment as $value) {

$temp = $this->user->get($value['user\_id']);

foreach ($temp as $var) {

$temp = $var;

break;

}

$coments[] = new Coments($value,$temp);

}

return $coments;

}

public function index($id = 1)

{

$titlecoment = $this->getTitleComent();

$data = array(

'titlecoment' => $titlecoment,

'pageIndex' => 0,

'username' => $this->session->userdata('username'),

'title' => 'Главная страница'

);

$this->load->helper('url');

$this->load->view('menu',$data);

$this->load->view('home');

$this->load->view('footer');

}

}

<?php

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

class Login extends CI\_Controller {

public function logined()

{

if(isset($\_POST) && isset($\_POST['user\_login\_btn']))

{

$username = htmlspecialchars($\_POST['username'], NULL, 'ISO-8859-1');

$user\_password = htmlspecialchars($\_POST['user\_password'], NULL, 'ISO-8859-1');

$this->load->model('user');

$user = $this->user->getUser($username);

foreach ($user as $item)

{

$user = $item;

break;

}

if($user['password'] == $user\_password && $username == 'admin'){

{

$newdata = array(

'username' => $username,

'email' => $user['email'],

'user\_id' => $user['user\_id']

);

$this->session->set\_userdata($newdata);

redirect('admin/home');

}

}

if($user['password'] == $user\_password && $username == $user['username'])

{

if($username == "" || $user\_password == "")

redirect('home');

$newdata = array(

'username' => $username,

'email' => $user['email'],

'user\_id' => $user['user\_id']

);

$this->session->set\_userdata($newdata);

redirect('home');

}

}

else{

redirect('home');

}

}

public function exitlog()

{

$this->session->unset\_userdata('username');

$this->session->unset\_userdata('email');

$this->session->unset\_userdata('user\_id');

redirect('home');

}

}

<?php

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

class Coments {

public $coment = null;

public $user = null;

function \_\_construct($c,$u){

$this->coment = $c;

$this->user = $u;

}

}

class Registration extends CI\_Controller {

public function getTitleComent()

{

$this->load->model('coment');

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

$this->db->limit (5, 0);

$coment = $this->coment->getlist();

$this->load->model('user');

$users = array();

$coments = array();

foreach ($coment as $value) {

$temp = $this->user->get($value['user\_id']);

foreach ($temp as $var) {

$temp = $var;

break;

}

$coments[] = new Coments($value,$temp);

}

return $coments;

}

public function index($id = 1)

{

if(false === $this->session->userdata('username'))

redirect('home');

$titlecoment = $this->getTitleComent();

$data = array(

'titlecoment' => $titlecoment,

'pageIndex' => 7,

'username' => $this->session->userdata('username'),

'title' => 'Регистрация пользователя'

);

$this->load->helper('url');

$this->load->view('menu',$data);

$this->load->view('registration');

$this->load->view('footer');

}

public function registrated()

{

if(isset($\_POST) && isset($\_POST['user\_registr\_btn'])){

$username = htmlspecialchars($\_POST['user\_name'], NULL, 'ISO-8859-1');

$userpass1 = htmlspecialchars($\_POST['user\_password1'], NULL, 'ISO-8859-1');

$userpass2 = htmlspecialchars($\_POST['user\_password2'], NULL, 'ISO-8859-1');

$userfio = htmlspecialchars($\_POST['user\_fio'], NULL, 'ISO-8859-1');

$usermail = htmlspecialchars($\_POST['user\_mail'], NULL, 'ISO-8859-1');

if($userpass1 == $userpass2){

$this->load->model('user');

$user = $this->user->getUser($username);

foreach ($user as $item)

{

$user = $item;

break;

}

if($user == false){

echo "4<br />";

$data = array(

'username' => $username,

'password' => $userpass1,

'fio' => $userfio,

'email' => $usermail,

'is\_comented' => 1

);

$this->user->add($data);

$userid = $this->user->getUser($username);

foreach ($userid as $item)

{

$userid = $item;

break;

}

$userid = $userid['user\_id'];

$newdata = array(

'username' => $username,

'email' => $usermail,

'user\_id' => $userid

);

$this->session->set\_userdata($newdata);

redirect('home');

}

}

}

}

}

<?php

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

class Coments {

public $coment = null;

public $user = null;

function \_\_construct($c,$u){

$this->coment = $c;

$this->user = $u;

}

}

class Searchs extends CI\_Controller {

private $searchString = "";

private $ingredients = null;

public function getTitleComent()

{

$this->load->model('coment');

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

$this->db->limit (5, 0);

$coment = $this->coment->getlist();

$this->load->model('user');

$users = array();

$coments = array();

foreach ($coment as $value) {

$temp = $this->user->get($value['user\_id']);

foreach ($temp as $var) {

$temp = $var;

break;

}

$coments[] = new Coments($value,$temp);

}

return $coments;

}

public function index()

{

$titlecoment = $this->getTitleComent();

$searchString = "";

$ingredients = null;

$data = array(

'titlecoment' => $titlecoment,

'pageIndex' => 4,

'username' => $this->session->userdata('username'),

'title' => 'Поиск рецепта'

);

$this->load->helper('url');

$this->load->view('menu',$data);

$this->load->view('searchs',$data);

$this->load->view('footer');

}

public function search($id = 1)

{

if(isset($\_POST['user\_search\_btn'])){

$datain = htmlspecialchars($\_POST['search\_text'], NULL, 'ISO-8859-1');

$searchString = $datain;

}

if($searchString != ""){

$titlecoment = $this->getTitleComent();

$this->load->model('recipe');

$recipes = $this->recipe->getsearch($datain, $id);

$recordCount = count($this->recipe->getsearch($datain, $id)) / 10;

$data = array(

'recipes' => $recipes,

'titlecoment' => $titlecoment,

'pageIndex' => 4,

'currentPage' => $id,

'searchflag' => 1,

'recordCount' => $recordCount,

'username' => $this->session->userdata('username'),

'title' => 'Поиск рецепта'

);

$this->load->helper('url');

$this->load->view('menu',$data);

$this->load->view('searchs',$data);

$this->load->view('footer');

}

}

public function recipesearch($id = 1)

{

if(isset($\_POST['user\_search\_recipe\_btn'])){

$titlecoment = $this->getTitleComent();

$ingredients = array();

$this->load->model('recipe');

foreach ($\_POST['ingredients'] as $ingredient) {

$temp = $this->recipe->getproductid($ingredient);

$temp = $temp[0];

$ingredients[] = $temp['product\_id'];

}

}

if($ingredients != null){

$recipeId = $this->recipe->getrecipeid($ingredients);

$recipes = array();

foreach ($recipeId as $value) {

$temp = $this->recipe->get($value['recipe\_id']);

$temp = $temp[0];

$recipes[] = $temp;

}

$recordCount = count($recipes) / 10;

$data = array(

'recipes' => $recipes,

'titlecoment' => $titlecoment,

'pageIndex' => 4,

'currentPage' => $id,

'searchflag' => 1,

'recordCount' => $recordCount,

'username' => $this->session->userdata('username'),

'title' => 'Поиск рецепта'

);

$this->load->helper('url');

$this->load->view('menu',$data);

$this->load->view('searchs',$data);

$this->load->view('footer');

}

}

}

<?php

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

class Coments {

public $coment = null;

public $user = null;

function \_\_construct($c,$u){

$this->coment = $c;

$this->user = $u;

}

}

class Master extends CI\_Controller {

public function getTitleComent()

{

$this->load->model('coment');

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

$this->db->limit (5, 0);

$coment = $this->coment->getlist();

$this->load->model('user');

$users = array();

$coments = array();

foreach ($coment as $value) {

$temp = $this->user->get($value['user\_id']);

foreach ($temp as $var) {

$temp = $var;

break;

}

$coments[] = new Coments($value,$temp);

}

return $coments;

}

public function index()

{

$titlecoment = $this->getTitleComent();

$this->load->model('masters');

$master = $this->masters->get();

if(count($master) > 0)

$master = $master[0];

else{

$master = null;

}

$data = array(

'master' => $master,

'titlecoment' => $titlecoment,

'pageIndex' => 6,

'username' => $this->session->userdata('username'),

'title' => 'Мастер-класс'

);

$this->load->helper('url');

$this->load->view('menu',$data);

$this->load->view('users/master',$data);

$this->load->view('footer');

}

public function write()

{

if (isset($\_POST['user\_master\_btn'])) {

$this->load->model('masters');

$this->masters->adduser($this->session->userdata('user\_id'));

redirect('users/master/index');

}

}

}

<?php

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

class Coments {

public $coment = null;

public $user = null;

function \_\_construct($c,$u){

$this->coment = $c;

$this->user = $u;

}

}

function handle\_error($user\_error\_message, $system\_error\_message) {

die ($user\_error\_message ." " . $system\_error\_message);

};

class Addrecipe extends CI\_Controller {

public function getTitleComent()

{

$this->load->model('coment');

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

$this->db->limit (5, 0);

$coment = $this->coment->getlist();

$this->load->model('user');

$users = array();

$coments = array();

foreach ($coment as $value) {

$temp = $this->user->get($value['user\_id']);

foreach ($temp as $var) {

$temp = $var;

break;

}

$coments[] = new Coments($value,$temp);

}

return $coments;

}

public function index()

{

if ($this->session->userdata('username') === false || $this->session->userdata('username') != 'admin') {

redirect('home');

}

$titlecoment = $this->getTitleComent();

$this->load->model('recipe');

$products = $this->recipe->getproductlist();

$types = $this->recipe->gettypeslist();

$data = array(

'titlecoment' => $titlecoment,

'username' => $this->session->userdata('username'),

'title' => 'Раздел администратора - Добавление нового рецепта'

);

$this->load->helper('url');

$this->load->view('admin/menu',$data);

$data = array(

'products' => $products,

'types' => $types

);

$this->load->view('admin/add\_recipe',$data);

$this->load->view('admin/footer');

}

public function add()

{

if ($this->session->userdata('username') === false || $this->session->userdata('username') != 'admin') {

redirect('home');

}

if(isset($\_POST) && isset($\_POST['recipe\_add\_btn']))

{

define('DIRSEP', DIRECTORY\_SEPARATOR);

$upload\_dir = \_\_DIR\_\_.DIRSEP."..".DIRSEP."..".DIRSEP."..".DIRSEP."images".DIRSEP;

$image\_fildname = "recipe\_image";

$upload\_filename = $upload\_dir . $\_FILES[$image\_fildname]['name'];

move\_uploaded\_file($\_FILES[$image\_fildname]['tmp\_name'], $upload\_filename);

$description = htmlspecialchars($\_POST['recipe\_description'], NULL, 'ISO-8859-1');

$recipe = htmlspecialchars($\_POST['recipe\_recipe'], NULL, 'ISO-8859-1');

$section = htmlspecialchars($\_POST['select\_section'], NULL, 'ISO-8859-1');

$imagename = $\_FILES[$image\_fildname]['name'];

$data = array(

'description' => $description,

'recipe' => $recipe,

'section\_id' => $section,

'image' => $imagename,

'date' => date("Y-m-d H:i:s")

);

$this->load->model('recipe');

$id = $this->recipe->add($data);

$ingredients = array();

foreach ($\_POST['ingredients'] as $ingredient) {

$ingredients[] = array(

'recipe\_id' => $id,

'product\_id' => $ingredient['product\_id'],

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

'type\_id' => $ingredient['type\_id'],

);

}

$this->recipe->addIngredients($ingredients);

redirect('admin/home');

}

}

}