

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГОРОДА МОСКВЫ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ ЦЕНТР ПРОФЕССИОНАЛЬНЫХ КВАЛИФИКАЦИЙ И
СОДЕЙСТВИЯ ТРУДОУСТРОЙСТВУ «ПРОФЕССИОНАЛ»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к итоговой аттестационной работе на тему

«Разработка web-ресурса с использованием технологий HTML, CSS, JavaScript,

на примере web-ресурса: <http://herzberg.p-host.in/>

репозиторий проекта: <https://github.com/LagodaAN/shop2>

слушателя Лагода Андрея Николаевича группы №: 116

программы профессиональной переподготовки

«Frontend разработка»

Москва, 2024

Оглавление

ПОСТАНОВКА ЗАДАЧИ И ПЛАН РАБОТЫ.....	3
ОСНОВНАЯ ЧАСТЬ.....	4
НАЗНАЧЕНИЕ ВЕБ-РЕСУРСА.....	4
ОПИСАНИЕ ЭТАПОВ РАЗРАБОТКИ, ОПИСАНИЕ ФУНКЦИОНАЛА С ПРИЛОЖЕНИЕМ ЛИСТИНГОВ ИСХОДНОГО ПРОГРАММНОГО КОДА ОСНОВНОЙ ФУНКЦИИ, СТРУКТУРНЫХ МОДУЛЕЙ, ПРИМЕР КОДА-РАЗМЕТКИ.....	4
СПИСОК ЛИТЕРАТУРЫ	9

ПОСТАНОВКА ЗАДАЧИ И ПЛАН РАБОТЫ.

Задачей является создание веб ресурса – интернет-магазин. При начале работы над проектом, необходимо определиться, как будет работать серверная часть, как будет работать проект со стороны клиента. После чего необходимо определиться какой будет внешний вид сайта, какой будет внешний вид личного кабинета пользователя.

Задачи к наполнению сайта: наличие информации о компании, «часто задаваемые вопросы», контакты, схема проезда к офису, каталог цветов, личный кабинет пользователя, форма связи.

План работ:

1. Создание базы данных на хостинге.
2. Подключение CMS Opencart
3. Пошаговая вёрстка интернет-магазина.
4. Проверка работоспособности.

ОСНОВНАЯ ЧАСТЬ.

Назначение веб-ресурса.

Веб-ресурс предназначен для торговли цветами.

Описание этапов разработки, описание функционала с приложением листингов исходного программного кода основной функции, структурных модулей, пример кода-разметки

Этап разработки начинался с создания главной страницы, на которой попадает пользователь введя адрес главной страницы <http://herzberg.p-host.in/>

Веб ресурс создавался с использованием философии структуры MVC (model-view-controller) на базе ООП (объектно-ориентированное программирование). При вёрстке сайта использовались HTML5, CSS3, JS, JQuery, Bootstrap, FontAwesome. Для серверной части использовался PHP8 и MySQL.

В проект веб ресурса входит пять основных страниц: главная страница, страница категорий, страница продукта, страница логинизации, авторизации, страница оформления заказа, страница корзины, страница карта сайта. При переходе на сайт <http://herzberg.p-host.in/> пользователь попадает на страницу, в шапке которой расположено логотип, кнопка корзины, главное меню. Далее под шапкой сайта располагается большой баннер, приглашающий пользователю зарегистрироваться. Далее идут блоки категорий цветов при клике на которые пользователь попадает в категорию товара. Далее идет блок акции при клике на которую пользователь попадает на страницу продукта.

Загрузка сайта начинается с файла index.php находящегося в корне сайта. В файле index.php идет предварительная проверка, если всё хорошо, загружается папка catalog в которой находятся все исполняемые файлы. Верстка находится в папке \catalog\view\theme\default\template\common\. Здесь находятся файлы home.twig (это

файл главной станицы), а также файл `header.twig`, отвечающий за шапку сайта, и файл `footer.twig`, отвечающий за подвал сайта.

Реализована возможность добавления в корзину и избранное при помощи JS/Jquery, не заходя непосредственно в карточку товара.

```
<div class="card-product_img">
  
  <ul class="card-product_imgOverlay">
    <li><button onclick="cart.add('59', '1');"><i class="ti-shopping-cart"></i></button></li>
    <li><button onclick="wishlist.add('59');"><i class="ti-heart"></i></button></li>
  </ul>
</div>
<div class="card-body">
```

Функция `cart.add` принимает два параметра, первый это `product_id` товара, а второй это количество. Исполняемый файл находится: `catalog/view/javascript/common.js`

```
{% endfor %}
<script src="catalog/view/javascript/common.js" type="text/javascript"></script>
{% for link in links %}
```

Далее функция в файле `common.js` принимает эти данные и передает при помощи метода библиотеки Jquery и далее отправляет на сервер при помощи метода POST.

```
// Cart add remove functions
var cart = {
  'add': function(product_id, quantity) {
    $.ajax({
      url: 'index.php?route=checkout/cart/add',
      type: 'post',
      data: 'product_id=' + product_id + '&quantity=' + (typeof(quantity) != 'undefined' ? quantity : 1),
      dataType: 'json',
      beforeSend: function() {
        $('#cart > button').button('loading');
      },
      complete: function() {
        $('#cart > button').button('reset');
      },
      success: function(json) {
        $('.alert-dismissible, .text-danger').remove();

        if (json['redirect']) {
          location = json['redirect'];
        }

        if (json['success']) {
          $('header').before('<div class="alert alert-success alert-dismissible"><i class="fa fa-check-circle"></i> ' + json['success'] + ' <button type="button" class="close" data-dismiss="alert">&times;</button></div>');
        }

        // Need to set timeout otherwise it wont update the total
        setTimeout(function () {
          $('#cart-total').html(json['total']);
        }, 100);

        $('html, body').animate({ scrollTop: 0 }, 'slow');
      }
    });
  }
};
```

Далее по адресу: `index.php?route=checkout/cart/add` контроллер обрабатывает запрос и посылает его дальше в модели.

```
public function add() {
  $this->load->language('checkout/cart');

  $json = array();

  if (isset($this->request->post['product_id'])) {
    $product_id = (int)$this->request->post['product_id'];
  } else {
    $product_id = 0;
  }

  $this->load->model('catalog/product');

  $product_info = $this->model_catalog_product->getProduct($product_id);

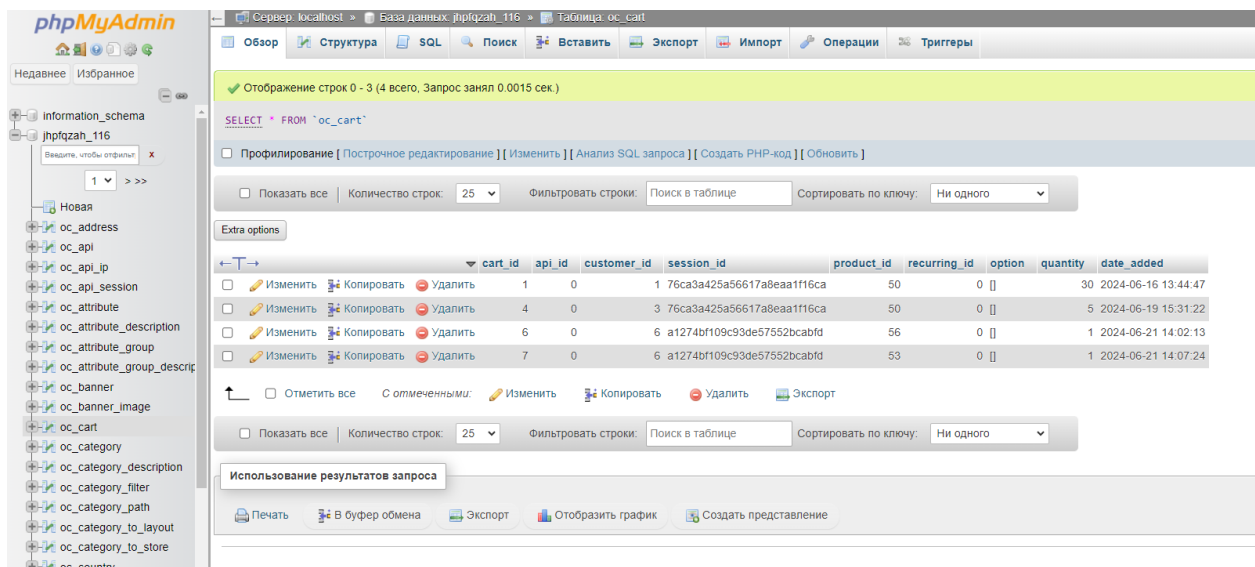
  if ($product_info) {
    if (isset($this->request->post['quantity'])) {
      $quantity = (int)$this->request->post['quantity'];
    } else {
      $quantity = 1;
    }

    if (isset($this->request->post['option'])) {
      $option = array_filter($this->request->post['option']);
    } else {
      $option = array();
    }
  }
}
```

Далее модель отправляет эти данные в базу данных:

```
$this->cart->add($this->request->post['product_id'], $quantity, $option, $recurring_id);

$json['success'] = sprintf($this->language->get('text_success'), $this->url->link('product/product',
'product_id=' . $this->request->post['product_id']), $product_info['name'], $this->url->link(
'checkout/cart'));
```



The screenshot shows the phpMyAdmin interface for a database named 'jhpqzahn_116'. The 'oc_cart' table is selected, and its structure is displayed. The table has 10 columns: cart_id, api_id, customer_id, session_id, product_id, recurring_id, option, quantity, and date_added. The data is shown in a table with 4 rows. The first row has cart_id 1, api_id 0, customer_id 1, session_id 76ca3a425a56617a8eaa1f16ca, product_id 50, recurring_id 0, option [], quantity 30, and date_added 2024-06-16 13:44:47. The second row has cart_id 4, api_id 0, customer_id 3, session_id 76ca3a425a56617a8eaa1f16ca, product_id 50, recurring_id 0, option [], quantity 5, and date_added 2024-06-19 15:31:22. The third row has cart_id 6, api_id 0, customer_id 6, session_id a1274bf109c93de57552bcabfd, product_id 56, recurring_id 0, option [], quantity 1, and date_added 2024-06-21 14:02:13. The fourth row has cart_id 7, api_id 0, customer_id 6, session_id a1274bf109c93de57552bcabfd, product_id 53, recurring_id 0, option [], quantity 1, and date_added 2024-06-21 14:07:24.

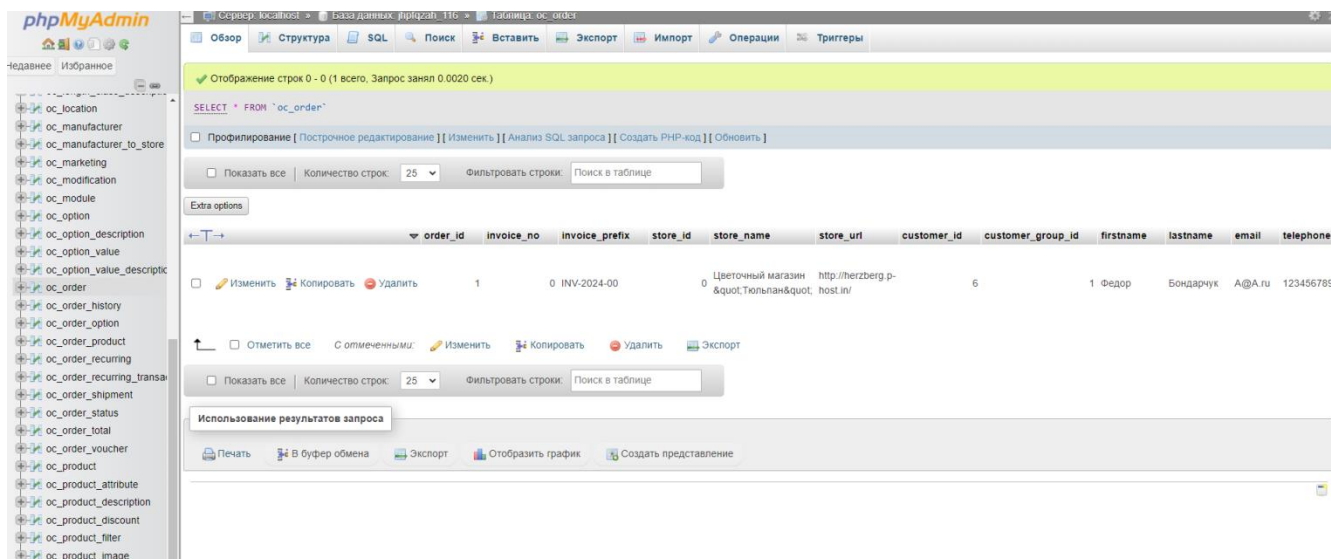
cart_id	api_id	customer_id	session_id	product_id	recurring_id	option	quantity	date_added
1	0	1	76ca3a425a56617a8eaa1f16ca	50	0	[]	30	2024-06-16 13:44:47
4	0	3	76ca3a425a56617a8eaa1f16ca	50	0	[]	5	2024-06-19 15:31:22
6	0	6	a1274bf109c93de57552bcabfd	56	0	[]	1	2024-06-21 14:02:13
7	0	6	a1274bf109c93de57552bcabfd	53	0	[]	1	2024-06-21 14:07:24

В базе данных находятся таблица oc_cart в соответствующих полях.

Реализован функционал оформления заказа с отправкой на eMail.

Исполняемые файлы находятся в папках controller, model, view.

Для этого в файле checkout.php создается новый экземпляр ControllerCheckoutCheckout, который расширяется от класса Controller, тем самым принимая все его свойства. В этом классе есть несколько функций. Одна из публичных функций index отвечает за вывод формы оформления заказа, сбор данных покупателя, их обработку и отправку в папку model. В папке model есть также файл checkout.php, которая принимает все эти данные и отправляет их в базу данных в таблицу oc_order



Также реализована функция отправки на email в файле `\controller\mail\order.php`. В этом файле создан экземпляр класса `ControllerMailOrder`. Он тоже расширяется от класса `Controller`.

```
$mail = new Mail($this->config->get('config_mail_engine'));
$mail->parameter = $this->config->get('config_mail_parameter');
$mail->smtp_hostname = $this->config->get('config_mail_smtp_hostname');
$mail->smtp_username = $this->config->get('config_mail_smtp_username');
$mail->smtp_password = html_entity_decode($this->config->get('config_mail_smtp_password'), ENT_QUOTES, 'UTF-8');
$mail->smtp_port = $this->config->get('config_mail_smtp_port');
$mail->smtp_timeout = $this->config->get('config_mail_smtp_timeout');

$mail->setTo($order_info['email']);
$mail->setFrom($from);
$mail->setSender(html_entity_decode($order_info['store_name'], ENT_QUOTES, 'UTF-8'));
$mail->setSubject(html_entity_decode(sprintf($language->get('text_subject'), $order_info['store_name'],
$order_info['order_id']), ENT_QUOTES, 'UTF-8'));
$mail->setHtml($this->load->view('mail/order_add', $data));
$mail->send();
```


СПИСОК ЛИТЕРАТУРЫ

1. Электронные ресурсы.

1. Документация по Opencart: электронный учебник <https://docs.opencart.com/en-gb/introduction/>
2. Электронный учебник по HTML и CSS [Электронный ресурс]: офиц. сайт. <https://htmlbook.ru/html5>
3. Официальная документация для CSS3 Mozilla [Электронный ресурс]: офиц. сайт. URL: <https://developer.mozilla.org/ru/docs/Glossary/CSS>
4. Электронный учебник по JavaScript [Электронный ресурс]: офиц. сайт. URL: <https://learn.javascript.ru/>
5. Электронный учебник по Bootstrap: <https://getbootstrap.com/>
6. Электронный учебник по jQuery: <https://jquery.com/>