## Аннотация

В документе описывается задача, поставленная на курсовую работу, а также веб-сайт, позволяющий просматривать информацию о почтовом отделении, имеющий возможность просмотра данных о складах, посылках, товаров из каталога, а также заполнения формы, необходимой при почтовом отправлении. Приводится структура данных и программные средства, используемые в процессе разработки веб-сайта. Рассмотрены страницы разработанного сайта почтового отделения.

Содержание

[Введение 4](#_Toc29243022)

[1. Анализ предметной области 6](#_Toc29243027)

[2. Постановка задачи 8](#_Toc29243033)

[3. Разработка структуры данных 9](#_Toc29243034)

[3.1. Структура данных сайта 9](#_Toc29243035)

[3.2. Структура базы данных 10](#_Toc29243037)

[4. Обоснование выбора программных средств 13](#_Toc29243038)

[4.1. Notepad++ 13](#_Toc29243040)

[4.2. USBWebserver 13](#_Toc29243041)

[4.3. HTML 14](#_Toc29243046)

[4.4. CSS 15](#_Toc29243047)

[4.5. PHP 15](#_Toc29243048)

[5. Описание логической структуры сайта 16](#_Toc29243049)

[6. Экспериментальная часть 23](#_Toc29243050)

[6.1. Дизайн сайта 23](#_Toc29243051)

[6.2. Инструкция для пользователя 27](#_Toc29243052)

[6.2.1. Вход в систему и регистрация пользователя 27](#_Toc29243053)

[6.2.2. Работа на странице «Абитуриенту» 30](#_Toc29243054)

[6.2.3. Работа на странице «Каталог» 31](#_Toc29243055)

[Заключение 34](#_Toc29243056)

[Список использованных источников 35](#_Toc29243057)

[Лист регистрации изменений 36](#_Toc29243058)

[Приложение А 37](#_Toc29243059)

# Введение

## В настоящее время, в связи с глобальным развитием сети Интернет, в программировании все более резко выделяться отдельная его отрасль - web-программирование. Изначально, оно не могло даже сравниться по своей сложности с другими областями «программистского ремесла», не "дотягиваясь" не только до системного, но даже и до прикладного программирования. Речь идет, конечно, о программировании сценариев для интернет сайтов, или, Web-программировании.

## Во время стремительного прогресса просто красиво оформленный текст и картинки на веб-сайте уже никого не удивят. Требования к сайтам, изменились - теперь для успешного представительства в Интернет необходимо предоставить своим посетителям различные возможности. При помощи обычного html этого не достичь, ведь html - это язык гипертекстовой разметки - инструмент для создания гиперссылок, вставки изображений, таблиц и др. С помощью языка html легко и быстро можно сделать форму для отправки какого-либо запроса. Такая форма будет иметь все необходимые атрибуты: и поле для ввода текста, и кнопку отправки. Однако, при нажатии на такую кнопку в большинстве случаев не произойдет ровным счетом ничего - ведь не был задан сценарий действий, которые следует выполнить, чтобы получить результат. Поэтому создание интерактивных компонентов - это задача для web-программирования.

## Веб-программирование осуществляется при помощи специальных программных средств - скриптов. Основная цель разработки сайта используя PHP программирование - нести в себе информацию, максимально доступно и удобно подавать ее пользователю, обеспечивать актуальность и полезность этой информации. Грамотно использовать технологию веб-программирования PHP для управления информацией - значит сделать еще один шаг к достижению заветной цели - созданию современного сайта.

## World Wide Web (www, web, рус.: веб, Всемирная Паутина) — распределенная информационная система, предоставляющая доступ к гипертекстовым документам по протоколу HTTP. Серверы www (веб-серверы) — это хранилища гипертекстовой (в общем случае) информации, управляемые специальным программным обеспечением. Документы, представленные в виде гипертекста называются веб-страницами. Несколько веб-страниц, объединенных общей тематикой, оформлением, связанных гипертекстовыми ссылками и обычно находящихся на одном и том же веб-сервере, называются веб-сайтом. Для загрузки и просмотра информации с веб-сайтов используются специальные программы — браузеры, способные обрабатывать гипертекстовую разметку и отображать содержимое веб-страниц [1].

# Анализ предметной области

## Почта — термин, используемый для обозначения [[2]](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%B0#cite_note-%D0%91%D0%A1%D0%AD3-1):

## почтовых отправлений;

## учреждений связи, к которым относятся: ответственная [почтовая администрация](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%86%D0%B8%D1%8F) (ОПА) и подчинённые структуры — [почтамты](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%B0%D0%BC%D1%82), [узлы связи](https://ru.wikipedia.org/wiki/%D0%A3%D0%B7%D0%B5%D0%BB_%D1%81%D0%B2%D1%8F%D0%B7%D0%B8), транспортные подразделения и сортировочные пункты, а также [почтовые отделения](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D0%BE%D0%B5_%D0%BE%D1%82%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5) в составе почтамтов и узлов связи;

## предприятия естественной монополии, производящего и реализующего услуги почтовых отправлений местного, внутреннего и международного уровнях обслуживания на основе лицензий.

Главнейшая особенность почты — естественный монополизм на рынке. Почта предлагает потребителям различные услуги почтовых отправлений на условиях всеобщего обслуживания и на всех уровнях обслуживания, Почта подразумевает единые тарифы и единые правила доступа к услугам почтовых отправлений для всех потребителей (вне зависимости от того, частным или юридическим лицом является потребитель; а также от гражданства, возраста, статуса физического лица).

Почтовое отделение — объект [почтовой связи](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B2%D1%8F%D0%B7%D1%8C), осуществляющий приёмку, сортировку, отправку, обработку и доставку [почты](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%B0) [[3]](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D0%BE%D0%B5_%D0%BE%D1%82%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5#cite_note-1).

## Почтовая связь является важнейшим элементом общенациональной инфраструктуры, выполняет ряд важнейших государственных и социальных функций. Состояние национальной почтовой инфраструктуры является неотъемлемым элементом глобальной конкурентоспособности российского государства.

Почтовая связь формирует значительный по размеру сектор экономики, в спектр услуг которого помимо классических, почтовых, входят финансовые, инфокоммуникационные и другие услуги.

Эффективно функционирующая Почта способствует созданию институциональных, финансовых, экономических и социальных основ долгосрочного социально-экономического развития страны.

Для повышения производительности почтового отделения необходимо внедрение информационно-коммуникационных технологий. В случае веб-сайта почтовому отделению будут предоставлены ввод формы, заполняемый при почтовом отправлении, в электронном виде, возможность отслеживания посылок по складам, а также популяризации собственных товаров посредством их размещения в каталоге сайта.

# Постановка задачи

В данной курсовой работе рассматривается создание сайта для почтового отделения. Данная тема актуальна, поскольку повышение производительности почтовых отделений всегда являлось одной из ярко выраженных проблем, существующих в государстве.

Цельюкурсовой работы является разработка информационной системы, представляющей собой сайт-каталог почтового отделения.

Задачи курсовой работы:

* разработка сайта с использованием современных программных средств;
* разработка дизайна сайта для повышения удобства восприятия с точки зрения пользователя;
* создание интерфейса, удобного для пользователя сайта;

# 3. Разработка структуры данных

## 3.1. Структура данных сайта

## Структура 15 файлов, используемых для функционирования веб-сайта, представлена на рисунке 1.

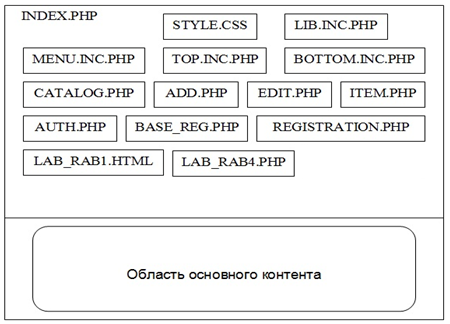


Рисунок 1 – Структура файлов web-сайта

Имеются следующие файлы:

* index.php – главная страница сайта;
* lab\_rab1.html – файл, реализующий форму, заполняемую при почтовом отправлении;
* lab\_rab4.php – файл, реализующий вывод информации о складах и посылках;
* style.css – файл стилей;
* lib.inc.php – файл, содержащий пользовательские функции;
* menu.inc.php – файл, содержащий код бокового меню сайта;
* top.inc.php – фал, содержащий код шапки сайта;
* bottom.inc.php – файл, содержащий код подвала сайта;
* auth.php **-** файл авторизации, блокирующий доступ к страницам неавторизованным пользователям;
* registration.php **-** файл регистрации, позволяющий пользователям регистрироваться и получать доступ к авторизации;
* base\_reg.php – файл, хранящий настройки подключения к базе данных;
* catalog.php - файл каталога, отображающий все записи каталога и позволяющий удалять выбранные записи;
* add.php - файл, реализующий добавление записи;
* edit.php - файл, реализующий редактирование записи ;
* item.php - файл, реализующий просмотр записи;

## 3.2. Структура базы данных

#### Базы данных являются неотъемлемой частью информационных систем, в том числе и веб-сайтов, предоставляющих пользователю различные данные. Если объем данных небольшой, то можно использовать обычные текстовые файлы, однако современные системы оперируют большими массивами информации. Это говорит о том, что целесообразным является использование баз данных.

База данных – хранилище информации, представленное одним или несколькими файлами. В настоящее время широко используются реляционные базы данных, в которых информация организована в виде таблиц, состоящих из строк и столбцов. Таблицы имеют между собой связи. Создание баз данных и доступ к ним обеспечивает система управления базами данных (СУБД).

PHP имеет наборы специальных функций для работы с такими СУБД, как Microsoft SQL Server (MSSQL), MySQL, Oracle, Firebird и др. Также PHP соответствует стандарту ODBC (Open Database Connectivity – открытый интерфейс доступа к базам данных), что позволяет обращаться к таким СУБД, как MS Access. Для курсовой работы была выбрана СУБД MySQL.

Для выполнения поставленной задачи разработана следующая структура базы данных (рисунок 2).

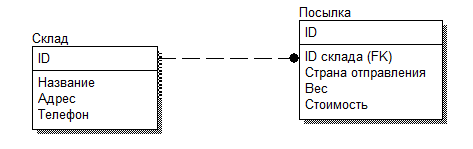


Рисунок 2 – Структура базы данных

Таблица Склад (sklad) – содержит информацию о складах, связанных с почтовым отделением (таблица 1). Поле ID является первичным ключом таблицы (автоинкрементное поле). Поля Название, Адрес и Телефон содержат информацию о названии, адресе и телефоне склада.

Таблица 1 – Атрибуты таблицы Склад (sklad)

|  |  |  |
| --- | --- | --- |
| Поле | Название поля | Тип поля |
| ID | Идентификатор склада | Integer |
| Название | Название склада | Varchar |
| Адрес | Адрес склада | Varchar |
| Телефон | Телефон склада | Varchar |

Таблица Посылка (posilka) – содержит информацию о посылках, отправляемых из почтового отделения (таблица 2). Поле ID содержит информацию об идентификаторе посылки и является первичным ключом (автоинкрементное поле). Поле ID склада является внешним ключом и содержит информацию об идентификаторе склада. Поле Страна отправления содержит информацию о стране отправления посылки. Поле Вес содержит информацию о весе посылки. Поле Стоимость содержит информацию о стоимости посылки.

Таблица 2 – Атрибуты таблицы Посылка (posilka)

|  |  |  |
| --- | --- | --- |
| Поле | Название поля | Тип поля |
| ID | Идентификатор посылки | Integer |
| ID склада | Идентификатор склада | Integer |
| Страна отправления | Страна отправления посылки | Varchar |
| Вес | Вес посылки | Float |
| Стоимость | Стоимость посылки | Float |

# Обоснование выбора программных средств

## Для разработки и реализации сайта были выбраны следующие программные средства:

* Notepad++;
* программное средство USBWebserver;
* язык разметки гипертекста HTML и CSS;
* язык программирования PHP;
  1. **Notepad++**

Notepad++ - это бесплатный редактор исходного кода, являющийся заменой «Блокнота», который поддерживает несколько языков. Работает в среде MS Windows, его использование регулируется лицензией GPL .

Основанный на мощном компоненте редактирования [Scintilla](https://www.scintilla.org/) , Notepad ++ написан на C ++ и использует чистый Win32 API и STL, что обеспечивает более высокую скорость выполнения и меньший размер программы. Notepad ++ оптимизирует как можно больше процедур без потери удобства [4]. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры.

## USBWebserver

**USBWebserver** – это комбинация популярных программ для веб-серверов: Apache, MySQL, PHP и phpMyAdmin. С USBWebserver можно разрабатывать и показывать PHP-сайты везде и в любое время. Основным преимуществом USBWebserver является то, что имеется возможность использовать его с USB-накопителя или обычного CD-диска.

## USBWebserver идеально подходит для:

## демонстрации автономной версии сайта;

## cоздания сайтов с использованием PHP где угодно и когда угодно;

## работы в нескольких местах над проектами [5].

## HTML

HTML (от англ. Hyper Text Markup Language - «язык гипертекстовой разметки») – стандартный язык разметки документов во Всемирной паутине. Большинство web-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

HTML дает авторам средства для:

* публикации онлайн-документов с заголовками, текстом, таблицами, списками, фотографиями и т. д.;
* получения информации онлайн через гипертекстовые ссылки одним нажатием кнопки;
* разработки форм для проведения транзакций с удаленными сервисами, для использования при поиске информации, бронировании, заказе товаров и т. д.;
* включения электронных таблиц, видеоклипов, аудиоклипов непосредственно в HTML-документы [6].

Текстовые документы, содержащие разметку на языке HTML (такие документы традиционно имеют расширение. html или .htm), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет - обозревателями», обычно предоставляют пользователю удобный интерфейс для запроса web - страниц, их просмотра (и вывода на иные

внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер.

## CSS

**CSS (Cascading Style Sheets)** — язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML). Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа, в том числе XML, SVG и XUL. Отделяя стиль представления документов от содержимого документов, CSS упрощает создание веб-страниц и обслуживание сайтов.

Каскадные таблицы стилей описывают правила форматирования элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств, остальные свойства не будут оказывать на него никакого влияния [7].

## PHP

PHP (рекурсивный акроним словосочетания PHP: Hypertext Preprocessor) – это распространенный язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML.

Вместо рутинного вывода HTML-кода командами языка (как это происходит, например, в Perl или C), скрипт PHP содержит HTML с встроенным кодом. Код PHP отделяется специальными [начальным и конечным тегами <?php и ?>](https://www.php.net/manual/ru/language.basic-syntax.phpmode.php), которые позволяют "переключаться" в "PHP-режим" и выходить из него.

PHP крайне прост для освоения, но вместе с тем способен удовлетворить запросы профессиональных программистов [8].

# Описание логической структуры сайта

Дизайн веб-сайта состоит из оформления, навигации по сайту посредством соответствующих элементов, и текстовой информации по тематике сайта.

Дизайн сайта должен соответствовать следующим требованиям:

* внешний вид сайта должен соответствовать выбранной тематике;
* навигация по сайту должна быть удобной для пользователя;

В верхней части страницы необходимо разместить название сайта, отражающее его тему.

Изображение в окне браузера состоит из трех основных элементов:

* название сайта;
* главное информационное меню слева страницы;
* главная страница.

Название сайта находится на каждой странице и не изменяется в зависимости от местонахождения на сайте. За это отвечает файл top.inc.php, который расположен в папке blocks.

Фрагменты кода для шапки сайта:

…

<table class="top">

<tr>

<td colspan="3"><a href="index.php">

<img src="images\post-office-icon.png" width="350" height="200" alt="1"/></a>

<p><strong>Учебный сайт "Почтовое отделение"</strong></p>

</td>

</tr>

…

Меню предназначается для навигации по сайту, в нем отображаются ссылки на все страницы сайта. За это отвечает файл menu.inc.php. Код для меню представлен в виде массива, который хранит название файлов, соответствующих пунктам меню.

…

<?php

$menu = array(

"Главная"=>"index.php",

"Отправителям"=>"index.php?page=lab1",

"Посылки"=>"index.php?page=lab4",

"Каталог"=>"index.php?page=catalog");

?>

…

Основное содержание страницы меняется в зависимости от местонахождения пользователя.

Главная страница отображается при первом входе на сайт.

Код главной страницы представлен в Приложении А (файл index.php). Сайт разбит на блоки, что подтверждается следующим фрагментом кода:

…

<table class="table">

<tr>

<!—Подключаем шапку--!>

<?php include "blocks/top.inc.php" ?>

</tr>

<tr>

<!—Подключаем меню--!>

<?php include "blocks/menu.inc.php" ?>

<td>

…

</td>

</tr>

<tr>

<!—Подключаем подвал--!>

<?php include "blocks\bottom.inc.php" ?>

</tr>

</table>

…

Данная функция «вытаскивает» данные из файлов top.inc.php, menu.inc.php, bottom.inc.php, которые расположены в папке blocks.

За подключение к базе данных «pochta» отвечает файл base\_reg.php. Фрагмент кода:

<?php

$host="localhost";

$user="root";

$password="usbw";

$database="pochta";

?>

В курсовом проекте рассматривается два способа добавления данных в БД.

Первый способ – ручное добавление данных в базу данных. Предполагается добавление соответствующей информации вручную в поля таблиц базы данных. Для этого необходимо запустить локальный сервер USBWebserver и зайти в phpMyAdmin (рисунок 3).

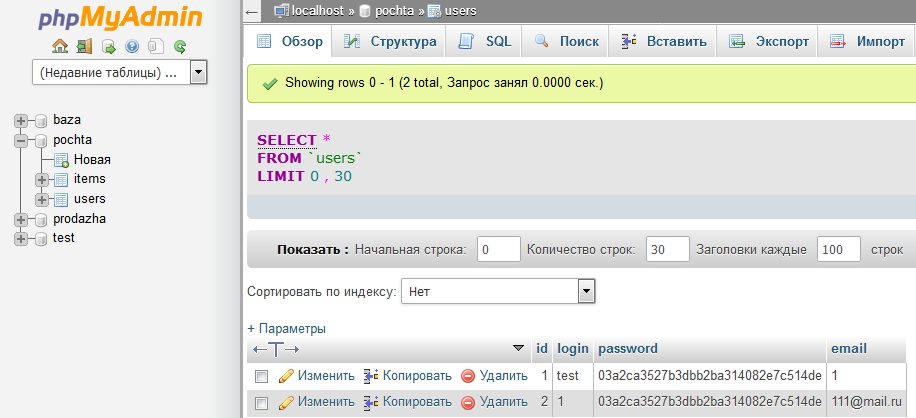


Рисунок 3 – Окно phpMyAdmin

Второй способ – автоматическое добавление данных в базу данных через администраторскую панель в невидимой части сайта. За это отвечает файл registration.php. Фрагмент кода добавления данных в БД:

…

else

{

$query ="INSERT INTO users (login,password,email) VALUES ('$login','$user\_password','$email')";

$result\_set = mysqli\_query($connect,$query) or die("Ошибка добавления пользователя!" . mysqli\_error($connect));

header("Location: index.php?login=".$login);

}

…

При входе в систему проверяется наличие логина пользователя в БД. За это отвечает файл auth.php. Фрагмент кода, отвечающий за проверку логина:

…

$login = clearData($\_POST['login']);

$user\_password = md5(md5(clearData($\_POST['user\_password']).'pechkin'));

// подключаемся к БД

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

$query ="SELECT id FROM users where login = '$login' and password = '$user\_password'";

$result = mysqli\_query($connect,$query) or die("Ошибка проверки логина и email!" . mysqli\_error($connect));

$result = $result->fetch\_assoc();

…

За просмотр записи отвечает файл item.php. Фрагмент кода:

<?php

$id = clearData($\_GET['id']);

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

$query = "SELECT \* FROM items WHERE id = '$id'"; // получаем инфу по данному id

$result = mysqli\_query($connect,$query) or die("Ошибка получения данных!" . mysqli\_error($connect));

$result = $result->fetch\_assoc();

mysqli\_close($connect);

?>

<a href='index.php?page=catalog' style='margin-left:40px' class='catalog-link'>Назад</a>

<a href='index.php?page=edit&id=<?php echo $id; ?>' style='margin-left:20px' class='catalog-link'>Редактировать</a>

…

C помощью функции href осуществляется переход на страницу каталога (файл catalog.php) или на форму редактирования товара (файл edit.php).

Фрагмент кода редактирования товара:

…

<?php

if (isset($\_POST['edit']))

{

if (!empty($\_POST['name']) && !empty($\_POST['article']) && !empty($\_POST['manufacturer']) && !empty($\_POST['price']) && !empty($\_POST['description']))

{

$image = $result['image'];

$name = clearData($\_POST['name']);

$article = clearData($\_POST['article']);

$manufacturer = clearData($\_POST['manufacturer']);

$equipment = clearData($\_POST['equipment']);

$price = clearData($\_POST['price']);

$description = clearData($\_POST['description']);

$query = "SELECT id FROM items WHERE name = '$name' and id<>'$id'";

$result = mysqli\_query($connect,$query) or die("Ошибка проверки названия!" . mysqli\_error($connect));

if (count($result->fetch\_assoc()) == 0) // если таки наименованй нет, то добавляем запись

{

if ($\_FILES['item-img']['tmp\_name']) // если выбрали изображение

$image = loadImage(); // грузим его

if ($image != "error"){

$query = "UPDATE items SET name = '$name', article = '$article', manufacturer = '$manufacturer', equipment = '$equipment', price = '$price', description = '$description', image ='$image' WHERE id = '$id'";

$result = mysqli\_query($connect,$query) or die("Ошибка изменения записи!" . mysqli\_error($connect));

mysqli\_close($connect);

header("Location: index.php?page=catalog");

exit;

}

}

…

?>

…

Редактирование осуществляется с помощью метода $\_POST[‘edit’].

За удаление товара отвечает файл catalog.php. Фрагмент кода удаления товара:

…

<?php

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

if (!empty($\_POST['deleteTov'])){

foreach($\_POST['deleteTov'] as $v)

{

$query = "DELETE FROM items WHERE id = '$v'"; // удаляем записи из таблицы items

$result = mysqli\_query($connect,$query) or die("Ошибка удаления!" . mysqli\_error($connect));

@unlink('images/catalog/'.$v.'.jpg');

}

…

?>

…

Удаление осуществляется с помощью метода $\_POST[‘del’].

За добавление товара в каталог отвечает файл add.php. Фрагмент кода добавления товара:

…

<?php

if (isset($\_POST['add']))

{

if (!empty($\_POST['name']) && !empty($\_POST['article']) && !empty($\_POST['manufacturer']) && !empty($\_POST['price']) && !empty($\_POST['description']))

{

$image = "no-image"; // устанавливаем по умолчанию что у товара нет изображения

$name = clearData($\_POST['name']);

$article = clearData($\_POST['article']);

$manufacturer = clearData($\_POST['manufacturer']);

$equipment = clearData($\_POST['equipment']);

$description = clearData($\_POST['description']);

$price = clearData($\_POST['price']);

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

// проверяем нет ли в таблицы записи с таким названием

$query = "SELECT id FROM items WHERE name = '$name' ";

$result = mysqli\_query($connect,$query) or die("Ошибка проверки названия!" . mysqli\_error($connect));

if (count($result->fetch\_assoc()) == 0) // если таки наименованй нет, то добавляем запись

{

if ($\_FILES['item-img']['tmp\_name']) // если выбрали изображение

$image = loadImage(); // грузим его

if ($image != "error"){ // если нет никаких ошибок добавляем в каталог

$query = "INSERT INTO items (name,article,manufacturer,equipment,price,description,image) VALUES ('$name','$article','$manufacturer','$equipment','$price','$description','$image')";

$result = mysqli\_query($connect,$query) or die("Ошибка добавления записи!" . mysqli\_error($connect));

mysqli\_close($connect);

header("Location: index.php?page=catalog");

exit;

}

…

?>

…

Добавление осуществляется с помощью метода $\_POST[‘add’].

# Экспериментальная часть

## Дизайн сайта

Дизайн сайта **–**это совокупность графических элементов, шрифтов и цветов, реализованных на сайте.

Основная задача дизайна сайта – объединение всех информационных блоков и формирование у посетителя приятного впечатления. Грамотно разработанный дизайн является одним из важнейших факторов, определяющих посещаемость вашей веб-страницы.  
   Хороший дизайн сайта – это незаметный, ненавязчивый дизайн, который не отвлекает посетителя от основного – от предоставленной целевой информации [9].

Главная страница - это страница, на которую попадает пользователь при входе на сайт (рисунок 4). На ней предоставляется информация о почтовой связи, в которой основными элементами как раз-таки являются почтовые отделения.

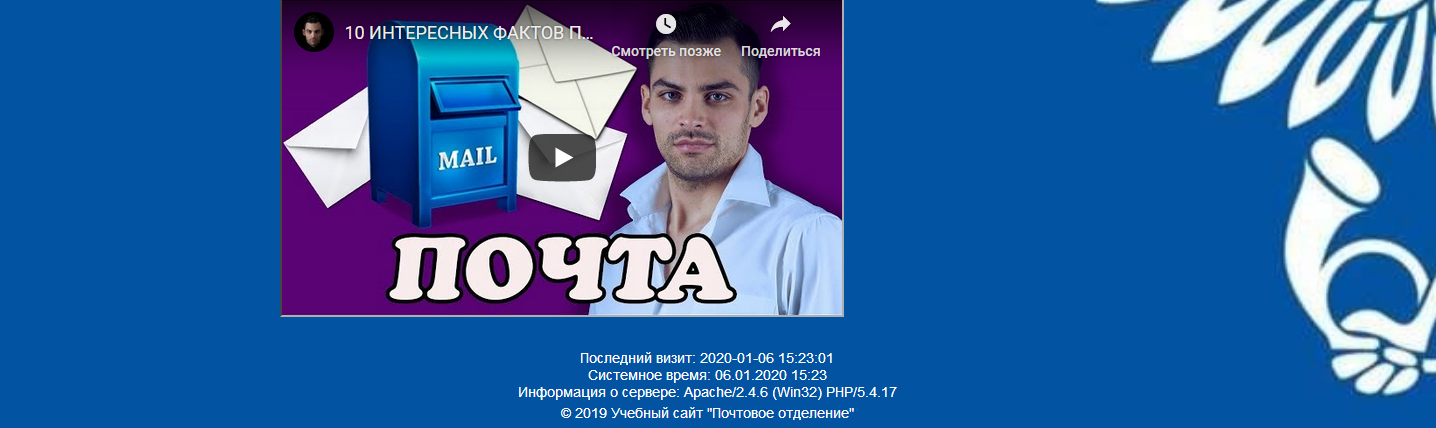


Рисунок 4 – Главная страница сайта

Следующая страница сайта – страница «Отправителям» (рисунок 5). На этой странице располагается форма, заполняемая при почтовом отправлении. Цель данной страницы – возможность заполнения формы в электронном виде для повышения производительности работы отделения.



Рисунок 5 – Страница «Отправителям»

На странице «Каталог» находится информация о почтовых ящиках, имеющихся в продаже у почтового отделения (рисунки 6 – 7).



Рисунок 6 – Страница «Каталог»

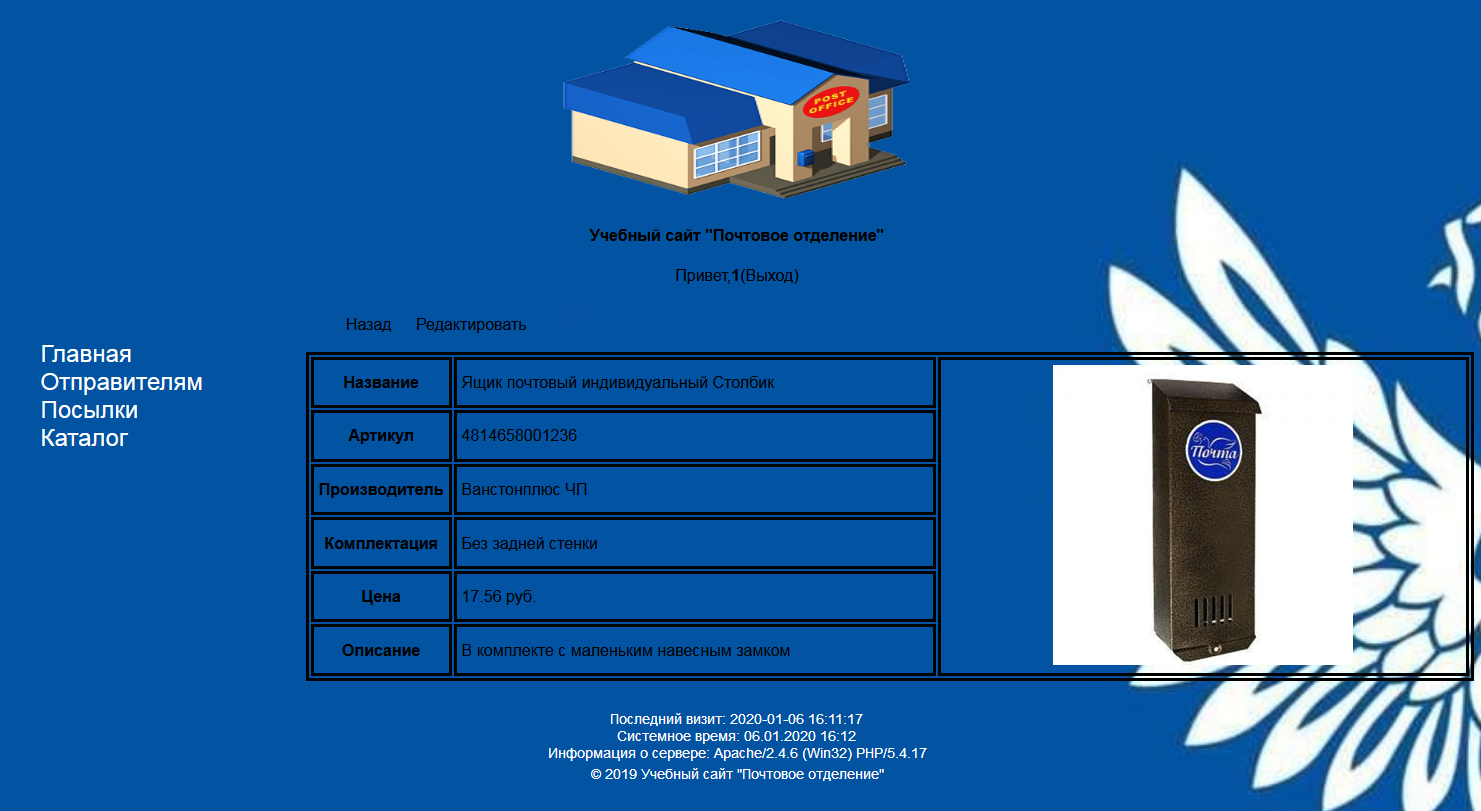


Рисунок 7 – Информация о товаре из каталога

Страница «Посылки» содержит данные о складах и посылках, находящихся на них (рисунок 8). Цель данной страницы – возможность вести статистику по посылкам, отправляемым из почтового отделения.



Рисунок 8 – Страница «Посылки»

Коды, реализующие описанные страницы, представлены в приложении А.

## 

## Инструкция для пользователя

### Вход в систему и регистрация пользователя

Для получения доступа к веб-сайту необходимо ввести данные учетной записи (логин и пароль), если пользователь ранее был зарегистрирован на сайте. Вход в систему представлен на рисунке 9.

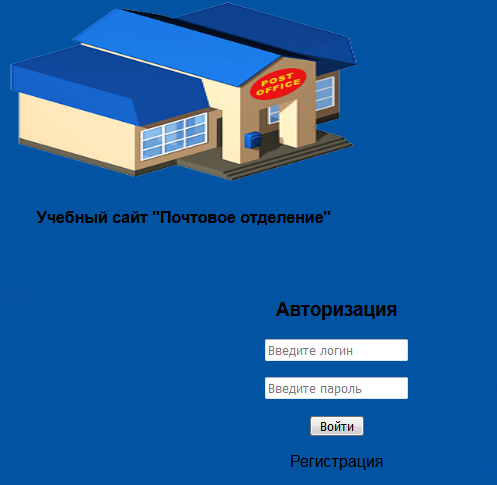


Рисунок 9 – Вход в систему

Если же учетная запись не создана, то пользователю предоставляется возможность зарегистрироваться на сайте почтового отделения. Регистрация пользователя представлена на рисунке 10.

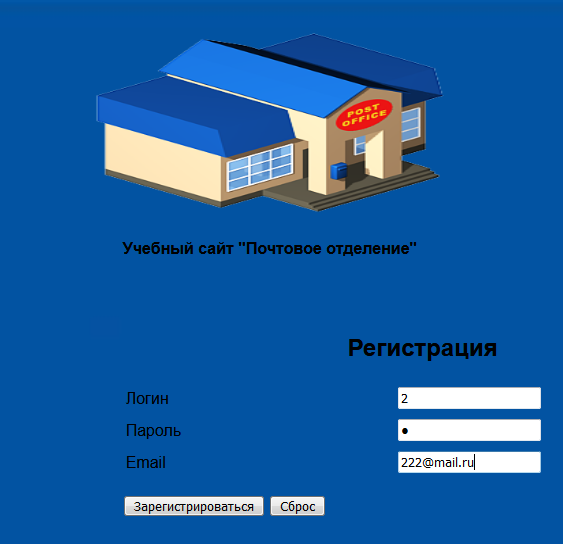


Рисунок 10 – Регистрация пользователя

После успешной регистрации пользователю предоставляется возможность войти на сайт с помощью логина и пароля, которые были указаны при регистрации. Вход в систему представлен на рисунке 11.

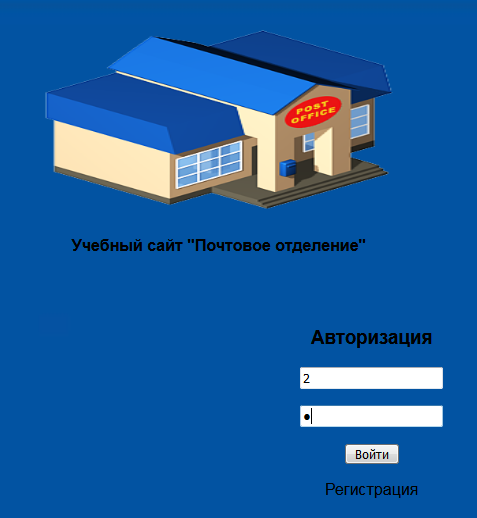


Рисунок 11 – Авторизация на сайте

После успешного авторизации в правом верхнем углу сайта появится приветствие, логин пользователя и гиперссылка для выхода с сайта (рисунок 12).



Рисунок 12 – Результат авторизации на сайте

### Работа на странице «Абитуриенту»

На этой странице пользователь может заполнить все необходимые поля с данными на форме, заполняемой при почтовом отправлении.

Полями, необходимыми для заполнения, являются:

* от кого (указывается фамилия, имя, отчество);
* откуда (указывается адрес);
* индекс отправителя;
* кому (указывается фамилия, имя, отчество);
* куда (указывается адрес);
* индекс получателя;
* тип посылки (простая, заказная, с уведомлением, с описью);
* есть ли наложенный платёж;
* вес;
* стоимость;
* описание к посылке.

Данные поля представлены на рисунке 13.

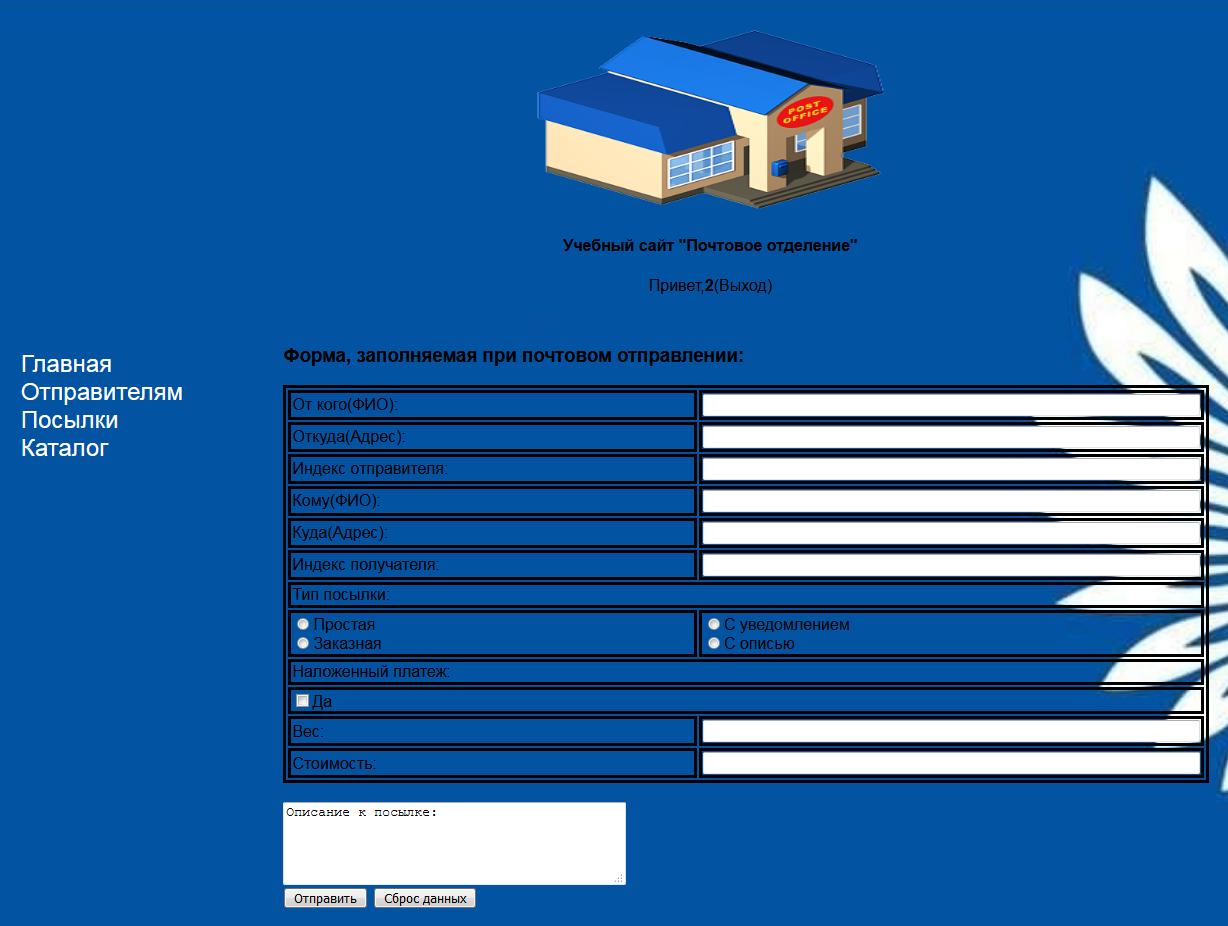


Рисунок 13 – Информация, заполняемая при почтовом отправлении

### Работа на странице «Каталог»

На данной странице хранится каталог почтовых ящиков, предлагаемых почтовым отделением. Страница с каталогом представлена на рисунке 14.

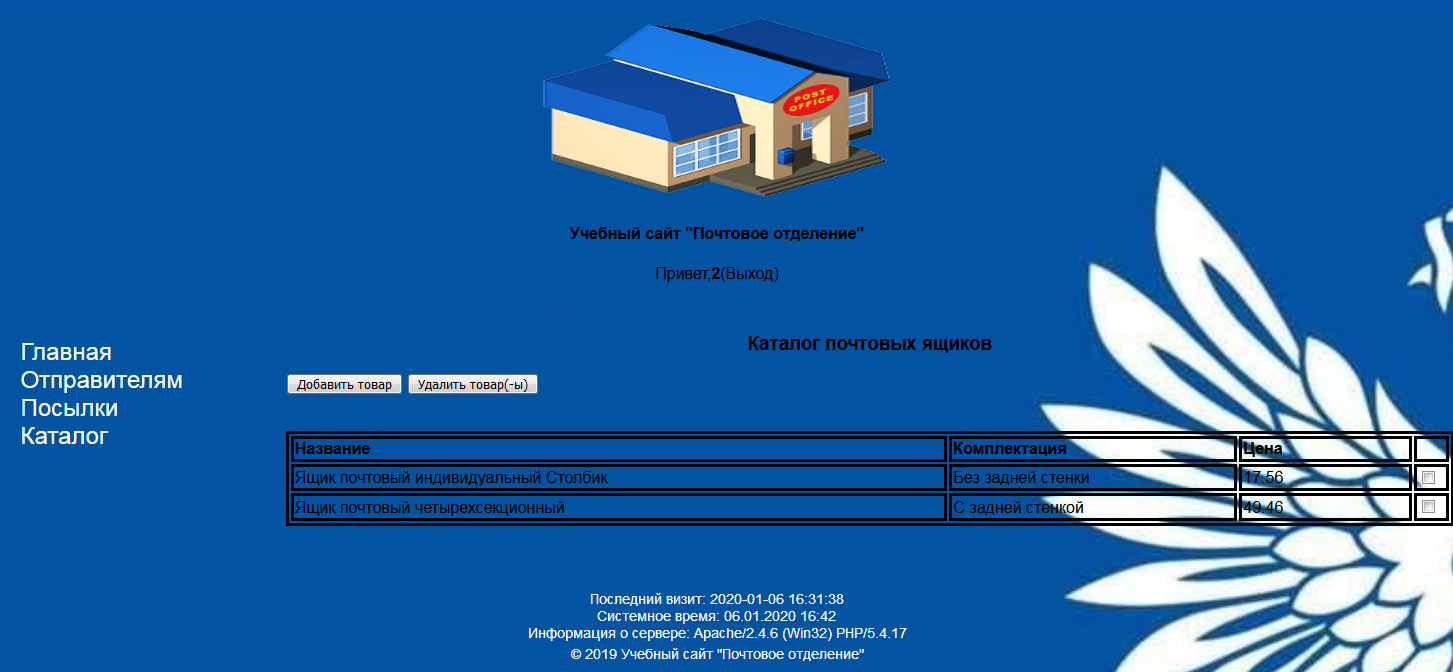


Рисунок 14 – Страница «Каталог»

Товар из каталога можно добавлять, удалять, редактировать и просматривать подробно, нажимая на название товара. Нажав на название товара, произойдет переход на форму просмотра товара. Форма изображена на рисунке 15.



Рисунок 15 – Описание товара

Чтобы отредактировать данные о товаре, необходимо нажать на «Редактировать». После нажатия произойдет переход на форму редактирования товара. Форма изображена на рисунке 16.

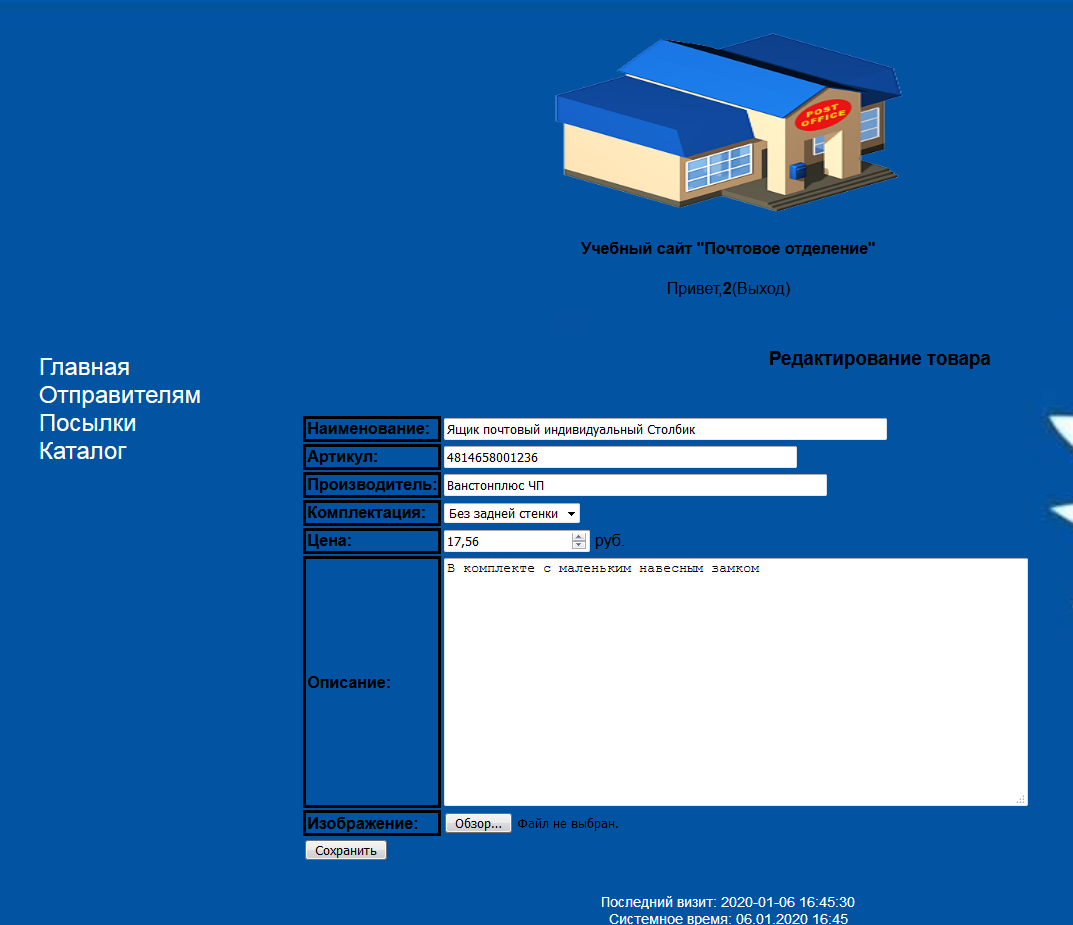


Рисунок 16 – Редактирование товара

После того, как информация о товаре будет отредактирована, необходимо нажать на кнопку «Сохранить».

# Заключение

Результатом курсовой работы стал готовый к работе веб-сайт, в котором размещена информация о почтовом отделении. В ходе проведения работы поставленные задачи были решены. По результатам работы можно сделать следующие выводы:

* сайт предоставляет пользователям необходимую информацию о почтовом отделении;
* дизайн сайта соответствует предпочтениям целевой аудитории сайта;
* реализована навигация с помощью меню по страницам сайта.

# Список использованных источников

1. А. Г. Анатольев. Сервис WWW // 4stud.info. 2017 [Электронный ресурс]. http://www.4stud.info/web-programming/lecture1.html (дата обращения: 06.01.2020).

2. Почта // Плата — Проб. — М. : Советская энциклопедия, 1975. — (Большая советская энциклопедия : [в 30 т.] / гл. ред. А. М. Прохоров ; 1969—1978, т. 20).

3. Post office (англ.). Glossary. Postal Service Information. Canada Post[en] Corporation (дата обращения 06.01.2020).

4. What is Notepad++? (англ.) // Notepad++ [Электронный ресурс]. <https://notepad-plus-plus.org/> (дата обращения: 06.01.2020).

5. What is USBWebserver? (англ.) // USBWebserver [Электронный ресурс]. <https://www.usbwebserver.net/webserver/> (дата обращения: 06.01.2020).

6. HTML & CSS (англ.) // W3C [Электронный ресурс]. <https://www.w3.org/standards/webdesign/htmlcss> (дата обращения: 06.01.2020).

7. Основы CSS // HTML5Book. 2015 [Электронный ресурс]. <https://html5book.ru/osnovy-css> (дата обращения: 06.01.2020).

8. Что такое PHP? // PHP [Электронный ресурс]. <https://www.php.net/manual/ru/intro-whatis.php> (дата обращения: 06.01.2020).

9. Что такое дизайн сайта? // Web-Siter [Электронный ресурс]. <http://www.web-siter.com/web-design.html> (дата обращения: 06.01.2020).

# Лист регистрации изменений

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц) в документе | № доку-мента | Входящие номер сопроводи-тельного документа | Под-пись | Дата |
| Изм | Изме-нен-ных | Заме-нен-ных | Нов-ых | Анну-лиро-ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

## 

# Приложение А

Текст модулей сайта

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Рязанский государственный радиотехнический университет**

**имени В.Ф. Уткина»**

Кафедра Автоматизированных систем управления

**«Разработка сайта-каталога для почтового отделения»**

Текст программы

090402-10-19

Листов 21

Рязань 2019

###### **index.php**

<?php

session\_start();

ob\_start();

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

date\_default\_timezone\_set('Europe/Minsk');

header("Content-Type: text/html; charset=utf-8");

header("Cache-control: no-store");

include "lib.inc.php";

include "base\_reg.php";

if (isset($\_COOKIE['lastVisit']))

$lastVisit = $\_COOKIE['lastVisit'];

setcookie('lastVisit',date('Y-m-d H:i:s'),time()+0xFFFFFFF);

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

if (isset($\_GET['exit']))

{

session\_destroy();

header("Location: index.php");

exit;

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Учебный сайт "Почтовое отделение"</title>

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

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

</head>

<body>

<table class="table">

<tr>

<?php include "blocks/top.inc.php" ?>

</tr>

<tr>

<?php include "blocks/menu.inc.php" ?>

<td>

<table class="content">

<tr>

<td class="content\_td">

<?php

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

{

switch($page)

{

case 'lab1': include 'lab\_rab1.html'; break;

case 'lab4': include 'lab\_rab4.php'; break;

case 'catalog': include 'catalog.php';break;

case 'add': include 'add.php'; break;

case 'item': include 'item.php'; break;

case 'edit': include 'edit.php'; break;

default : echo '<p><b>Почтовая связь является важнейшим элементом общенациональной инфраструктуры, выполняет ряд важнейших государственных

и социальных функций.Состояние национальной почтовой инфраструктуры является неотъемлемым элементом глобальной конкурентоспособности

российского государства.</b></p>

<p><b>Почтовая связь формирует значительный по размеру сектор экономики, в спектр услуг которого помимо классических, почтовых, входят

финансовые, инфокоммуникационные и другие услуги.Эффективно функционирующая Почта способствует созданию институциональных,

финансовых, экономических и социальных основ долгосрочного социально-экономического развития страны.</b></p>

<img src="images\twvwswm4g9zfvtni.jpg" height="200" width="400" alt="2"/>

<img src="images\cennoe\_pismo.jpg" height="200" width="400" alt="3"/>

<iframe width="560" height="315" src="https://www.youtube.com/embed/P4BqM6W1Qes" allow="accelerometer; autoplay;

encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>';break;

}

}

else{

if ($page=="registration")

include 'registration.php';

else

include 'auth.php';

}

?>

</td>

</tr>

</table>

</td>

</tr>

<tr>

<?php include "blocks\bottom.inc.php" ?>

</tr>

</table>

</body>

</html>

###### **lab\_rab1.html**

<h3>Форма, заполняемая при почтовом отправлении:</h3>

<form action="POST">

<table class="table2">

<tr>

<td>От кого(ФИО): </td>

<td> <input type="text" name="fromfio" size="79" maxlength="70" /></td>

</tr>

<tr>

<td>Откуда(Адрес):</td>

<td> <input type="text" name="fromaddress" size="79" maxlength="200" /></td>

</tr>

<tr>

<td>Индекс отправителя: </td>

<td> <input type="text" name="fromindex" size="79" maxlength="20" /></td>

</tr>

<tr>

<td>Кому(ФИО): </td>

<td> <input type="text" name="forfio" size="79" maxlength="70" /></td>

</tr>

<tr>

<td>Куда(Адрес): </td>

<td> <input type="text" name="foraddress" size="79" maxlength="200" /></td>

</tr>

<tr>

<td>Индекс получателя: </td>

<td> <input type="text" name="forindex" size="79" maxlength="20" /></td>

</tr>

<tr>

<td colspan="2">Тип посылки:</td>

</tr>

<tr>

<td>

<input type="radio" name="package" value="yes" />Простая<br/>

<input type="radio" name="package" value="yes" />Заказная<br/>

</td>

<td>

<input type="radio" name="package" value="yes" />С уведомлением<br/>

<input type="radio" name="package" value="yes" />С описью<br/>

</td>

</tr>

<tr>

<td colspan="2">Наложенный платеж: </td>

</tr>

<tr>

<td colspan="2">

<input type="checkbox" name="pay" value="yes" />Да<br/>

</td>

</tr>

<tr>

<td>Вес: </td>

<td> <input type="text" name="weight" size="79" maxlength="20" /></td>

</tr>

<tr>

<td >Стоимость: </td>

<td > <input type="text" name="ammount" size="79" maxlength="20" /></td>

</tr>

</table>

<br/>

<div class="mytext">

<textarea name="comments" cols="40" rows="4">

Описание к посылке:

</textarea>

<br/>

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

<input type="reset" value="Сброс данных" />

</div>

</form>

###### **lab\_rab4.php**

<?php

function get\_column\_names\_with\_meta ($conn\_id)

{

$query = "SELECT \* FROM sklad,posilka WHERE 1 = 0";

if (!($result\_id = mysqli\_query ($conn\_id,$query)))

return (FALSE);

echo "<table border='1' align='center'>";

echo "<tr><th>Таблица</th><th>Поле</th><th>Тип</th><th>Длинна</th></tr>";

for ($i = 0; $i < mysqli\_num\_fields ($result\_id); $i++)

{

if ($field = mysqli\_fetch\_field ($result\_id))

echo "<tr>";

echo "<td>".$field->table."</td>";

echo "<td>".$field->name."</td>";

echo "<td>".$field->type."</td>";

echo "<td>".$field->length."</td>";

echo "</tr>";

}

echo "</table>";

mysqli\_free\_result ($result\_id);

}

function resultSetArray($result\_set){ // функция преобразования полученных данных из БД в ассоциативный массив

$array =array();

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

$array[] = $row;

return $array;

}

function addData($connect)

{

mysqli\_query($connect,"INSERT INTO sklad (name,address,phone) values ('Склад 71','ш. Московское, д.11, 390000, г. Рязань', '(8-1111) 61-40-77')");

mysqli\_query($connect,"INSERT INTO sklad (name,address,phone) values ('Склад 72','ул. Космонавтов, д.6, г. Скопин', '(8-2222) 61-64-22')");

mysqli\_query($connect,"INSERT INTO sklad (name,address,phone) values ('Склад 73','ул. Октябрьская, д.38, 390001, г.Рязань', '(8-3333) 46-44-64')");

mysqli\_query($connect,"INSERT INTO sklad (name,address,phone) values ('Склад 74','ул. Гоголя, 29, 390008, г.Скопин', '(8-4444) 46-40-80')");

$rows = resultSetArray(mysqli\_query($connect,"SELECT \* FROM sklad ORDER BY id ASC"));

echo "<br>Таблица sklad:<br>";

echo "<table border='1' align='center' width='1000'>";

echo "<tr><th>ID</th><th>Название</th><th>Адрес</th><th>Телефон</th></tr>";

foreach ($rows as $item)

{

echo "<tr>";

echo "<td>".$item['id']."</td>";

echo "<td>".$item['name']."</td>";

echo "<td>".$item['address']."</td>";

echo "<td>".$item['phone']."</td>";

echo "</tr>";

}

echo "</table><br>";

mysqli\_query($connect,"INSERT INTO posilka (id\_sklad,country,ves,stoim) values ('1','Китай','10','50')");

mysqli\_query($connect,"INSERT INTO posilka (id\_sklad,country,ves,stoim) values ('1','Китай','22.10','150.50')");

mysqli\_query($connect,"INSERT INTO posilka (id\_sklad,country,ves,stoim) values ('3','Беларусь','30','65')");

mysqli\_query($connect,"INSERT INTO posilka (id\_sklad,country,ves,stoim) values ('1','Беларусь','19','45')");

mysqli\_query($connect,"INSERT INTO posilka (id\_sklad,country,ves,stoim) values ('4','Россия','5','25.50')");

mysqli\_query($connect,"INSERT INTO posilka (id\_sklad,country,ves,stoim) values ('4','Россия','3.48','10')");

$rows = resultSetArray(mysqli\_query($connect,"SELECT \* FROM posilka ORDER BY id ASC"));

echo "<br>Таблица posilka:<br>";

echo "<table border='1' align='center' width='1000'>";

echo "<tr><th>ID</th><th>ID склада</th><th>Страна отправления</th><th>Вес (кг.)</th><th>Стоимость</th></tr>";

foreach ($rows as $item)

{

echo "<tr>";

echo "<td>".$item['id']."</td>";

echo "<td>".$item['id\_sklad']."</td>";

echo "<td>".$item['country']."</td>";

echo "<td>".$item['ves']."</td>";

echo "<td>".$item['stoim']."</td>";

echo "</tr>";

}

echo "</table>";

}

function query2($connect)

{

echo "<br>Общая стоимость посылок по стране отправления:<br>";

$rows = resultSetArray(mysqli\_query($connect,"SELECT country,round(sum(stoim),2) as 'summa' FROM posilka GROUP BY(country)

ORDER BY country ASC"));

echo "<table border='1' align='center' width='1000'>";

echo "<tr><th>Страна отправления</th><th>Сумма</th></tr>";

foreach ($rows as $item)

{

echo "<tr>";

echo "<td>".$item['country']."</td>";

echo "<td>".$item['summa']."</td>";

echo "</tr>";

}

echo "</table>";

}

function delDB($connect){

$query = 'DROP DATABASE lab4';

mysqli\_query($connect,$query) or die("Ошибка при удалении базы данных: " . mysqli\_error($connect));

mysqli\_close($connect);

}

$host = "localhost";

$user = "root";

$password = "usbw";

$database = "lab4";

$connect = mysqli\_connect($host, $user, $password) or die("Ошибка соединения : " . mysqli\_error($connect));

$query = "CREATE DATABASE IF NOT EXISTS lab4 ";

mysqli\_query($connect,$query) or die("Ошибка создания БД : " . mysqli\_error($connect));

mysqli\_select\_db($connect,$database);

$query = "CREATE TABLE IF NOT EXISTS sklad (

id integer not null auto\_increment primary key,

name varchar(65) not null,

address varchar(75) not null)";

mysqli\_query($connect,$query) or die("Ошибка при создании таблицы sklad : " . mysqli\_error($connect));

$query = "CREATE TABLE IF NOT EXISTS posilka (

id integer not null auto\_increment primary key,

id\_sklad varchar(10) not null,

country varchar(10) not null,

ves float not null,

stoim varchar(20) not null)";

mysqli\_query($connect,$query) or die("Ошибка при создании таблицы posilka: " . mysqli\_error($connect));

$query = "ALTER TABLE sklad ADD phone varchar(50)";

mysqli\_query($connect,$query) or die("Ошибка при изменении таблицы sklad: " . mysqli\_error($connect));

echo "<br><br>";

$query = "ALTER TABLE posilka MODIFY id\_sklad integer not null,MODIFY country varchar(40) not null,MODIFY stoim float not null";

mysqli\_query($connect,$query) or die("Ошибка при изменении таблицы posilka : " . mysqli\_error($connect));

addData($connect); // заполняем таблицы и выводим

query2($connect); // запрос №2

delDB($connect); //удаляем БД

?>

###### **style.css**

/\* свойства всего документа \*/

html, body

{

font:16px Arial, Helvetica, sans-serif;

color:#000000;

background:url("images/1.jpg") no-repeat;

text-align:center;

border:0;

background-size: auto;

height: 0;

-moz-background-size: 100%; /\* Firefox 3.6+ \*/

-webkit-background-size: 100%; /\* Safari 3.1+ и Chrome 4.0+ \*/

background-size: 100%; /\* Современные браузеры \*/

}

a {

text-decoration: none; /\* Отменяем подчеркивание у ссылки \*/

}

.top td

{

height:15%;

text-align:center;

}

/\* свойства основной таблицы \*/

.table

{

margin:0 auto;

width:80%;

height:100%;

border-collapse: separate;

}

/\* общие свойства верхней, нижней части сайта, области контента и меню \*/

.top, .content, .menu, .bottom

{

width:100%;

height:100%;

}

/\* дополнительные свойства верхней части сайта \*/

.top

{

border-spacing:0;

}

.menu

{

font:24px Arial, Helvetica, sans-serif;

}

/\* общие свойства левой и правой области верхней части сайта \*/

.top\_left, .top\_right

{

padding:5px 20px;

}

/\* дополнительные свойства левой области верхней части сайта \*/

.top\_left

{

width:60%;

border-right-width:0;

text-align:left;

}

/\* дополнительные свойства правой области верхней части сайта \*/

.top\_right

{

width:40%;

border-left-width:0;

text-align:right;

}

/\* свойства нижней части сайта \*/

.bottom

{

text-align:center;

font-size:14px;

color:#fff;

}

/\* общие свойства областей контента и меню \*/

.menu td, .content\_td

{

vertical-align:top;

text-align:left;

padding:20px 20px;

position:relative;

width:25%;

height:75%;

}

.table p

{

align: center;

}

.content td

{

width:600px;

height:100%;

}

.bottom td

{

height:5%;

}

.table1

{

width:500px;

border:3px solid #000000;

}

.table1 td

{

border:3px solid #000000;

width: 50px;

height: 45px;

}

.table1 tr { height: 45px; }

.mytext

{

text-align:left;

}

.table2

{

border:3px solid #000000;

width:80%;

}

.table2 td

{

padding:5;

border:3px solid #000000;

border-collapse:collapse;

width:50%;

text-align:left;

}

.table2 tr

{

width:100%;

}

.catalog

{

border:3px solid #000000;

}

.catalog td,th

{

border:3px solid #000000;

}

.item {

border:3px solid #000000;

}

.item th {

text-align:center;

padding: 5px 5px;

}

.item td {

border:3px solid #000000;

text-align:left;

padding: 5px 5px;

}

.catalog-link:link {

color: #000000;

border-bottom: 1px;

}

.catalog-link:visited {

color: #000000;

}

.catalog-link:active {

color: #000000;

border-bottom: 1px;

}

###### **lib.inc.php**

<?php

function getMenu($menu, $vertical=true)

{

$style = "";

if(!$vertical)

{

$style = "display:inline";

}

echo '<ul style="list-style-type:none">';

foreach ($menu as $link=>$href)

{

echo "<li style='$style'><a href=\"$href\" style='color: #FFFFFF'>", $link, '</a></li>';

}

echo '</ul>';

}

function clearData($data)

{

return trim(strip\_tags($data));

}

function loadImage(){

if ($\_FILES['item-img']['type'] != 'image/jpeg') {

echo '<center><font color="red"><b>Не верный тип изображения!</b></font></center>';

return "error";

}

else

{

if ($\_FILES['item-img']['size'] > 100000) {

echo '<center><font color="red"><b>Превышен максимальный размер файла! (макс.=100кб.)</b></font></center>';

return "error";

}

else

{

$Image = imagecreatefromjpeg($\_FILES['item-img']['tmp\_name']);

$Size = getimagesize($\_FILES['item-img']['tmp\_name']);

$Tmp = imagecreatetruecolor(300, 300);

imagecopyresampled($Tmp, $Image, 0, 0, 0, 0, 300, 300, $Size[0], $Size[1]);

include "base\_reg.php";

$link = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($link,"utf8");

if (isset($\_POST["add"])){

$query = "SELECT id FROM items order by id desc";

$result = mysqli\_query($link,$query) or die("Ошибка!" . mysqli\_error($link));

$result = $result->fetch\_assoc();

mysqli\_close($link);

$img = $result['id']+1;

}

else {

$query = "SELECT image FROM items where id =".$\_GET['id']." ";

$result = mysqli\_query($link,$query) or die("Ошибка!" . mysqli\_error($link));

$result = $result->fetch\_assoc();

$img = $result['image'];

}

if ($img=="no-image") $img = $\_GET['id'];

imagejpeg($Tmp, 'images/catalog/'.$img.'.jpg'); // сохраняем

imagedestroy($Image);

imagedestroy($Tmp);

}

}

return $img;

}

?>

###### **menu.inc.php**

<?php

$menu = array(

"Главная"=>"index.php",

"Отправителям"=>"index.php?page=lab1",

"Посылки"=>"index.php?page=lab4",

"Каталог"=>"index.php?page=catalog");

?>

<td style="width:20%; height:100%">

<table class="menu">

<tr>

<td>

<?php

getMenu($menu);

?>

</td>

</tr>

</table>

</td>

###### **top.inc.php**

<td colspan="3">

<!-- Верхняя часть сайта -->

<table class="top">

<tr>

<td colspan="3"><a href="index.php">

<img src="images\post-office-icon.png" width="350" height="200" alt="1"/></a>

<p><strong>Учебный сайт "Почтовое отделение"</strong></p>

</td>

</tr>

<tr>

<td colspan="2" class="top\_left">

<?php

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

echo "Привет,<b>".$\_SESSION['login']."</b><a href='index.php?exit=true'class='catalog-link'>(Выход)</a>";

?>

</td>

</tr>

</table>

</td>

###### **bottom.inc.php**

<td colspan="3">

<!-- Нижняя часть сайта -->

<table class="bottom">

<tr>

<td>

Последний визит: <?=$lastVisit?>

<br/> Системное время: <?=date("d.m.Y H:i")?>

<br/> Информация о сервере: <?=$\_SERVER['SERVER\_SOFTWARE']?>

</td>

</tr>

<tr>

<td>&copy; 2019 Учебный сайт "Почтовое отделение"

</td>

</tr>

</table>

</td>

###### **auth.php**

<?php

if (isset($\_POST['login']) && isset($\_POST['user\_password']))

{

if (!empty($\_POST['login']) && !empty($\_POST['user\_password']))

{

$login = clearData($\_POST['login']);

$user\_password = md5(md5(clearData($\_POST['user\_password']).'pechkin'));

// подключаемся к БД

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

$query ="SELECT id FROM users where login = '$login' and password = '$user\_password'";

$result = mysqli\_query($connect,$query) or die("Ошибка проверки логина и email!" . mysqli\_error($connect));

$result = $result->fetch\_assoc();

if(!empty($result))

{

$\_SESSION['ip'] = $\_SERVER['REMOTE\_ADDR'];

$\_SESSION['login'] = $login;

header("Location: http://".$\_SERVER['HTTP\_HOST'].$\_SERVER['REQUEST\_URI']);

exit;

}

else{

echo "<br><font color=red>Неверный логин или пароль!</font>";

}

}

else echo "<center><font color=red><b>Заполните все поля!</b></font></center>";

}

?>

<div style="text-align:center;">

<h3>Авторизация</h3>

<form method="POST">

<p><input type="text" name="login" placeholder="Введите логин"></p>

<p><input type="password" name="user\_password" placeholder="Введите пароль"></p>

<p><input type="submit" value="Войти" id="submit"></p>

<p><a href="index.php?page=registration" class="catalog-link">Регистрация</a></p>

</form>

</div>

###### **registration.php**

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST')

{

if (!empty($\_POST['user\_password']) && !empty($\_POST['login']) && !empty($\_POST['email']))

{

$user\_password = md5(md5(clearData($\_POST['user\_password']).'pechkin')); // хэшируем пароль

$login = clearData($\_POST['login']);

$email = clearData($\_POST['email']);

// подключаемся к БД

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

$query ="SELECT id FROM users where login = '$login' or email = '$email'"; // проверяем нет ли пользователя с такии логином или емайл

$result = mysqli\_query($connect,$query) or die("Ошибка проверки логина и email!" . mysqli\_error($connect));

$result = $result->fetch\_assoc();

if(!empty($result))

{

echo "<br><font color=red>Пользователь с таким логином или email уже существует!</font>";

}

else

{

$query ="INSERT INTO users (login,password,email) VALUES ('$login','$user\_password','$email')";

$result\_set = mysqli\_query($connect,$query) or die("Ошибка добавления пользователя!" . mysqli\_error($connect));

header("Location: index.php?login=".$login);

}

mysqli\_close($connect); // закрываем соединение с БД

}

else echo '<br><font color=red>Заполните все поля!</font>';

}

?>

<center>

<table>

<tr>

<td>

<h2 align="center">Регистрация</h2>

<form method="POST">

<table>

<tr style="height:30px;">

<td>Логин</td>

<td><input type="text" name="login" required></td>

</tr>

<tr style="height:30px;">

<td>Пароль</td>

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

</tr>

</tr>

<tr style="height:30px;">

<td>Email</td>

<td><input type="text" name="email" required></td>

</tr>

</table>

<p>

<input type="submit" value="Зарегистрироваться">

<input type="reset" value="Сброс">

</p>

</form>

</td>

</tr>

</table>

</center>

###### **base\_reg.php**

<?php

$host="localhost";

$user="root";

$password="usbw";

$database="pochta";

?>

###### **catalog.php**

<?php

if (isset($\_POST['add'])) header("Location: index.php?page=add");

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

?>

<h3><center>Каталог почтовых ящиков</center></h3>

<div class="article">

<form action="index.php?page=catalog" method="POST">

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

<input type="submit" name="delete" value="Удалить товар(-ы)">

<br><br>

<?php

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

if (!empty($\_POST['deleteTov'])){

foreach($\_POST['deleteTov'] as $v)

{

$query = "DELETE FROM items WHERE id = '$v'"; // удаляем записи из таблицы items

$result = mysqli\_query($connect,$query) or die("Ошибка удаления!" . mysqli\_error($connect));

@unlink('images/catalog/'.$v.'.jpg');

}

}

else echo "<center><font color='red'><b>Выберите товары для удаления!</b></font></center><br>";

}

?>

<br>

<table class="catalog">

<tr>

<th >Название</th>

<th width="25%">Комплектация</th>

<th width="15%">Цена</th>

<th width="3%"></th>

</tr>

<?php

$query = "SELECT id,name,equipment,price FROM items ORDER BY id ASC"; // получаем записи из таблицы items

$result = mysqli\_query($connect,$query) or die("Ошибка вывода данных" . mysqli\_error($connect));

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

{

echo "<tr>";

echo "<td><a href='index.php?page=item&id=".$row['id']."' class='catalog-link'>".$row['name']."</a></td>";

echo "<td>".$row['equipment']."</td>";

echo "<td>".$row['price']."</td>";

echo "<td><input type='checkbox' name='deleteTov[]' value=".$row['id']."></td>";

echo "</tr>";

}

mysqli\_close($connect);

?>

</table>

</form>

</br>

</br>

</div>

###### **add.php**

<h3><center>Добавить товар</center></h3>

<?php

if (isset($\_POST['add']))

{

if (!empty($\_POST['name']) && !empty($\_POST['article']) && !empty($\_POST['manufacturer']) && !empty($\_POST['price']) && !empty($\_POST['description']))

{

$image = "no-image"; // устанавливаем по умолчанию что у товара нет изображения

$name = clearData($\_POST['name']);

$article = clearData($\_POST['article']);

$manufacturer = clearData($\_POST['manufacturer']);

$equipment = clearData($\_POST['equipment']);

$description = clearData($\_POST['description']);

$price = clearData($\_POST['price']);

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

// проверяем нет ли в таблицы записи с таким названием

$query = "SELECT id FROM items WHERE name = '$name' ";

$result = mysqli\_query($connect,$query) or die("Ошибка проверки названия!" . mysqli\_error($connect));

if (count($result->fetch\_assoc()) == 0) // если таки наименованй нет, то добавляем запись

{

if ($\_FILES['item-img']['tmp\_name']) // если выбрали изображение

$image = loadImage(); // грузим его

if ($image != "error"){ // если нет никаких ошибок добавляем в каталог

$query = "INSERT INTO items (name,article,manufacturer,equipment,price,description,image) VALUES ('$name','$article','$manufacturer','$equipment','$price','$description','$image')";

$result = mysqli\_query($connect,$query) or die("Ошибка добавления записи!" . mysqli\_error($connect));

mysqli\_close($connect);

header("Location: index.php?page=catalog");

exit;

}

}

else

echo '<center><font color="red"><b>Товар с таким названием уже есть в таблице!</b></font></center><br>';

}

else

echo '<center><font color="red"><b>Заполните все поля!</b></font></center><br>';

}

?>

<form method='POST' action='index.php?page=add' ENCTYPE='multipart/form-data'>

<table style="text-align:left;" >

<tr>

<th>Наименование:</th>

<td><input type='text' name='name' size="70"></td>

</tr>

<tr>

<th>Артикул:</th>

<td><input type='text' name='article' size="55"></td>

</tr>

<tr>

<th>Производитель:</th>

<td><input type='text' name='manufacturer' size="60"></td>

</tr>

<tr>

<th>Комплектация:</th>

<td>

<select size="1" name="equipment">

<option value="Без задней стенки" selected>Без задней стенки</option>

<option value="С задней стенкой">С задней стенкой</option>

</select>

</td>

</tr>

<tr>

<th>Цена:</th>

<td><input type='number' name='price' min="0" step="0.01">&nbsp;руб.</td>

</tr>

<tr>

<th>Описание:</th>

<td><textarea name='description' rows='15' cols='70' ></textarea>

</tr>

<tr>

<th>Изображение:</th>

<td><input type="file" name="item-img" accept="image/jpeg"></td>

</tr>

<rt>

<td colspan="2"><input type="submit" value="Добавить товар" name="add"></td>

</tr>

</table>

</form>

###### **edit.php**

<?php

$id = clearData($\_GET['id']);

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

$query = "SELECT \* FROM items WHERE id = '$id'"; // получаем всю инфу по данному id

$result = mysqli\_query($connect,$query) or die("Ошибка получения данных!" . mysqli\_error($connect));

$result = $result->fetch\_assoc();

?>

<center><h3>Редактирование товара</h3></center>

<?php

if (isset($\_POST['edit']))

{

if (!empty($\_POST['name']) && !empty($\_POST['article']) && !empty($\_POST['manufacturer']) && !empty($\_POST['price']) && !empty($\_POST['description']))

{

$image = $result['image'];

$name = clearData($\_POST['name']);

$article = clearData($\_POST['article']);

$manufacturer = clearData($\_POST['manufacturer']);

$equipment = clearData($\_POST['equipment']);

$price = clearData($\_POST['price']);

$description = clearData($\_POST['description']);

$query = "SELECT id FROM items WHERE name = '$name' and id<>'$id'";

$result = mysqli\_query($connect,$query) or die("Ошибка проверки названия!" . mysqli\_error($connect));

if (count($result->fetch\_assoc()) == 0) // если таки наименованй нет, то добавляем запись

{

if ($\_FILES['item-img']['tmp\_name']) // если выбрали изображение

$image = loadImage(); // грузим его

if ($image != "error"){

$query = "UPDATE items SET name = '$name', article = '$article', manufacturer = '$manufacturer', equipment = '$equipment', price = '$price', description = '$description', image ='$image' WHERE id = '$id'";

$result = mysqli\_query($connect,$query) or die("Ошибка изменения записи!" . mysqli\_error($connect));

mysqli\_close($connect);

header("Location: index.php?page=catalog");

exit;

}

}

else

echo '<center><font color="red"><b>Товар с таким названием уже есть в таблице!</b></font></center>';

}

else

echo '<center><font color="red"><b>Заполните все поля!</b></font></center>';

}

?>

<form method='POST' action='index.php?page=edit&id=<?php echo $id; ?>' ENCTYPE='multipart/form-data'>

<table style="text-align:left;" >

<tr>

<th>Наименование:</th>

<td><input type='text' name='name' value='<?=$result['name']?>' size="70"></td>

</tr>

<tr>

<th>Артикул:</th>

<td><input type='text' name='article' value='<?=$result['article']?>' size="55"></td>

</tr>

<tr>

<th>Производитель:</th>

<td><input type='text' name='manufacturer' value='<?=$result['manufacturer']?>' size="60"></td>

</tr>

<tr>

<th>Комплектация:</th>

<td>

<select size="1" name="equipment">

<option value="Без задней стенки" <?php if ($result['equipment'] == "Без задней стенки") echo "selected" ?> >Без задней стенки</option>

<option value="С задней стенкой" <?php if ($result['equipment'] == "С задней стенкой") echo "selected" ?> >С задней стенкой</option>

</select>

</td>

</tr>

<tr>

<th>Цена:</th>

<td><input type='number' name='price' min="0" step="0.01" value='<?=$result['price']?>' >&nbsp;руб.</td>

</tr>

<tr>

<th>Описание:</th>

<td><textarea name='description' rows='15' cols='70'><?=$result['description']?></textarea>

</tr>

<tr>

<th>Изображение:</th>

<td><input type="file" name="item-img" accept="image/jpeg"></td>

</tr>

<rt>

<td colspan="2"><input type="submit" value="Сохранить" name="edit"></td>

</tr>

</table>

</form>

###### **item.php**

<?php

$id = clearData($\_GET['id']);

$connect = mysqli\_connect($host, $user, $password, $database) or die("Не удалось подключиться к БД");

mysqli\_set\_charset($connect,"utf8");

$query = "SELECT \* FROM items WHERE id = '$id'"; // получаем инфу по данному id

$result = mysqli\_query($connect,$query) or die("Ошибка получения данных!" . mysqli\_error($connect));

$result = $result->fetch\_assoc();

mysqli\_close($connect);

?>

<a href='index.php?page=catalog' style='margin-left:40px' class='catalog-link'>Назад</a>

<a href='index.php?page=edit&id=<?php echo $id; ?>' style='margin-left:20px' class='catalog-link'>Редактировать</a>

<br/><br/>

<table class="item">

<tr>

<th>Название</th>

<td><?= $result['name'] ?></td>

<td rowspan="6" style="text-align:center;"><img src='images\catalog\<?php echo $result['image'].'.jpg';?> '></td>

</tr>

<tr>

<th>Артикул</th>

<td><?= $result['article']?></td>

</tr>

<tr>

<th>Производитель</th>

<td><?= $result['manufacturer']?></td>

</tr>

<tr>

<th>Комплектация</th>

<td><?= $result['equipment']?></td>

</tr>

<tr>

<th>Цена</th>

<td><?= $result['price'] ?> руб.</td>

</tr>

<tr >

<th>Описание</th>

<td ><?= $result['description'] ?></td>

</tr>

</table>

</div>