

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 4

По дисциплине Web-программирование

Тема работы Работа с базой данных MySQL и создание форм

Обучающийся Алексеев Тимофей Юрьевич

Факультет Факультет инфокоммуникационных технологий

Группа K3221

Направление подготовки 11.03.02 Инфокоммуникационные технологии и системы связи

Образовательная программа Программирование в инфокоммуникационных системах

Обучающийся	<u>17.12.2024</u> (дата)	<u> </u> (подпись)	<u>Алексеев Т.Ю.</u> (Ф.И.О.)
--------------------	-----------------------------	--	----------------------------------

Руководитель	<u> </u> (дата)	<u> </u> (подпись)	<u>Марченко Е.В.</u> (Ф.И.О.)
---------------------	---------------------------------------	--	----------------------------------

Санкт-Петербург
2024 г.

Цель

Научиться работать с базой данных MySQL с помощью скриптов PHP, получить навыки настройки wordpress и написать с помощью языка программирования web-сервер.

Задачи

1. Разработать веб-страницу и PHP-скрипт, который сохраняет данные в MySQL;
2. Создать запись логина и пароля в MySQL в движке wordpress в двух вариантах: в исходном и инвертированном побитовым;
3. Выбрать среду разработки и написать свой web-сервер.

Ход работы

Задание 1

В данном задании было реализовать форму, в который пользователь заполнял бы информацию о доставке.

Для этого была на HTML была написана следующая форма, содержащая в себе поля для имени, фамилии, отчества, адреса доставки, номера телефона, электронной почты, выпадающего меню с выбором продукта и комментария к заказу (использовался <textarea>).

Также, форма была связана с файлом send_info.php. И при нажатии на кнопку отправить происходил post-запрос.

```
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Форма доставки</title>
</head>
<body>
  <h2>Форма обратной связи</h2>
  <form action="send_info.php" method="post">
    <p>Имя:</p>
    <input type="text" name="first-name">
    <p>Фамилия:</p>
    <input type="text" name="second-name">
    <p>Отчество:</p>
    <input type="text" name="third-name">
    <p>Физический адрес доставки товара:</p>
    <input type="text" name="address">
    <p>Телефон:</p>
    <input type="text" name="phone">
    <p>Email:</p>
    <input type="email" name="email">
    <p>Выберите товар:</p>
    <select id="product" name="product">
      <option value="Product 1">Утюг</option>
      <option value="Product 2">Миксер</option>
      <option value="Product 3">Блендер</option>
      <option value="Product 4">Хлебопечка</option>
    </select>
    <p>Комментарий к заказу:</p>
    <textarea name="feedback" cols="20" row="20"></textarea><br>
    <input type="submit" value="Отправить">
  </form>
</body>
```

Рисунок 1 – HTML-код формы

В файле send_info.php. Была представлена логика обработки заказа и записи информации в базу данных.

Сначала были прописаны переменные, необходимые для подключения, после чего создано само подключение. Далее была получена вся информация из формы, заполненной пользователем. После чего был написан SQL-запрос, который вставлял в таблицу строчку, содержащую полученную информацию.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $host = 'localhost';
    $dbname = 'orders';
    $username = 'root';
    $password = '';

    $mysqli = new mysqli($host, $username, $password, $dbname);

    if ($mysqli->connect_error) {
        echo "Errno: ".$mysqli->connect_errno;
        exit();
    }

    $first_name = $_POST['first-name'];
    $second_name = $_POST['second-name'];
    $third_name = $_POST['third-name'];
    $address = $_POST['address'];
    $phone = $_POST['phone'];
    $email = $_POST['email'];
    $product = $_POST['product'];
    $comment = $_POST['feedback'];

    $sql = "INSERT INTO user_order (first_name, second_name, third_name, address, phone, email, product, comment)
        VALUES ('$first_name', '$second_name', '$third_name', '$address', '$phone', '$email', '$product', '$comment')";

    if ($mysqli->query($sql)) {
        echo "Форма успешно отправлена!";
    } else {
        echo "Ошибка!";
    }

    $mysqli->close();
} else {
    echo "Ошибка";
}
?>
```

Рисунок 2 – PHP-скрипт для записи в БД

Для работы с базой данных MySQL использовался XAMPP. Там, с помощью файла phpMyAdmin была создана база данных orders, в которой была создана таблица user_order. В ней создан первичный ключ (id) и 8 столбцов для каждой единицы информации от пользователя.

#	Имя	Тип
1	id 	int(11)
2	first_name	varchar(100)
3	second_name	varchar(100)
4	third_name	varchar(100)
5	address	varchar(300)
6	phone	varchar(100)
7	email	varchar(100)
8	product	varchar(100)
9	comment	text

Рисунок 3 – Структура таблицы user_order

Попробуем заполнить форму и отправить ее.

Форма обратной связи

Имя:

Фамилия:

Отчество:

Физический адрес доставки товара:

Телефон:

Email:

Выберите товар:

Комментарий к заказу:

Рисунок 4 – Заполненная форма

Проверим получение данных в базе данных. Как мы видим, наши данные успешно были получены.

id	first_name	second_name	third_name	address	phone	email	product	comment
1	Тимофей	Алексеев	Юрьевич	гор. Канаш, ул. Калинина, д. 2, кв. 77	9370102830	alekseev-timof@mail.ru	Product 1	Приезжайте скорее
2	Иванов	Иван	Иванович	г. Санкт-Петербург, ул. Пушкина, д. 52	+79370102830	alekseev-timof@mail.ru	Product 2	Домофон не работает, звоните по телефону

Рисунок 5 – Данные в таблице user_order

Задание 2

В этом задании было необходимо модернизировать скрипт wordpress, который записывает новых пользователей в базу данных MySQL.

Необходимо модифицировать php-скрипт, чтобы дополнительно логин и пароль записывались в отдельную новую таблицу в БД MySQL, а именно в двух вариантах: пароль записывается в исходном виде и в нем инвертируются биты (ноль меняется на единицу, единица меняется на ноль).

Для дальнейшей работы была создана таблица wp_users_new, в которую записывался логин, исходный пароль и инвертированный битовый.


#	Имя	Тип
1	id 	int(11)
2	user_name	varchar(100)
3	password_base	text
4	password_bit	text

Рисунок 6 – Схема таблицы wp_users_new

Далее необходимо найти сам скрипт, отвечающий за запись новых пользователей. Он находится по адресу ХАМРР\htdocs\wordpress\wp-includes\user.php. В нем необходимо найти строку 2422-2423, именно они отвечают за запись новых пользователей в базу данных. Их необходимо дополнить.

Для начала необходимо получить логин и пароль из списка по ключу, после чего применить функцию insert, которая вставляет в таблицу wp_user_new логин, исходный пароль и закодированный.

Для перевода пароля в десятичное число использовалась сумма crc32, после чего с помощью оператора побитовой инверсии ~ инвертировалось.

```
2421     } else {  
2422         $wpdb->insert( $wpdb->users, $data );  
2423         $user_id = (int) $wpdb->insert_id;  
2424  
2425         $user_login = $userdata['user_login'];  
2426         $user_password = $userdata['user_pass'];  
2427         $wpdb->insert( 'wp_users_new', array_merge(array('user_name' => $user_login),  
2428             array('password_base' => $user_password), array('password_bit' => ~([crc32($user_password)])));  
2429     }
```

Рисунок 7 – Модернизированный скрипт

Проверим, что все работает. Для этого через личный кабинет добавим несколько пользователей. Проверим их наличие в основной таблице wp_users.

ID	user_login	user_pass	user_nicename
1	adminTest	\$P\$B0OsYzwZaMgHC6.HXZgqBhmnwzM1Ku/	admintest
3	Test_user1	\$P\$B0MDP4viyvyQdMk3MuNlumEDOCWDUC1	test_user1
4	Test_user2	\$P\$BJNI4Gacez4FilyzfALMTELNE.niZK/	test_user2
5	Test_user3	\$P\$BPJw9XDBxEDHsQgFRZRtGmjlm.2x8w1	test_user3

Рисунок 8 – Таблица wp_users

Теперь проверим, что пользователи также добавились в таблицу wp_users_new. Всё работает корректно.

id	user_name	password_base	password_bit
1	Test_user1	qwerty12345	-274449864
2	Test_user2	abcdef123	-779278234
3	Test_user3	password12345	-75844941

Рисунок 9 – Таблица wp_users_new

Задание 3

В данном задании необходимо было написать web-сервер, работающий по указанному порту.

Для написания web-сервера был выбран язык Python и библиотеки `http.server` (необходим для реализации HTTP обработчика запросов – `SimpleHTTPRequestHandler`) и `socketserver` (обеспечивает базовые классы для создания серверов).

Для начала от `SimpleHTTPRequestHandler` был унаследован класс `RequestHandler`, в котором был перегружен метод `do_GET`, а именно в него было добавлено условие, если получен путь «/», то менять его на «index.html». То есть при обращении к <http://127.0.0.1/> будет производиться запуск файла.

После этого с помощью библиотеки `os` был поменян путь исполняемого файла, чтобы мы точно находились в одной директории с файлом `index.html`.

Далее запускался `TCPServer` с помощью библиотеки `socketserver`, используя вышеуказанный порт (переменная `PORT`). А вызов функции `serve_forever()` поддерживает работу и позволяет обрабатывать запросы до момента принудительной остановки.


```
import http.server
import socketserver
import os

PORT = 888

class RequestHandler(http.server.SimpleHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.path = 'index.html'
        return http.server.SimpleHTTPRequestHandler.do_GET(self)

os.chdir(os.path.dirname(os.path.abspath(__file__)))

with socketserver.TCPServer(("", PORT), RequestHandler) as server:
    print(f"Сервер запущен на порту {PORT}")
    server.serve_forever()
```

Рисунок 10 – Реализация web-сервера

Попробуем открыть в браузере адрес <http://127.0.0.1/>. В итоге нам откроется приветственное сообщение из index.html.

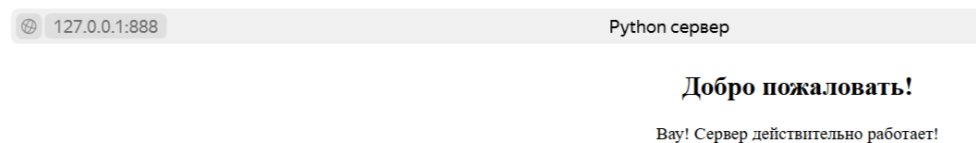


Рисунок 11 – Отображение index.html

Вывод

В ходе выполнения лабораторной работы цель была достигнута. Научились работать с базой данных MySQL с помощью скриптов PHP, получили навыки настройки wordpress и написали с помощью Python web-сервер.

В течение выполнения работы была настроена база данных MySQL и написан php-скрипт, отправляющий данные из формы в БД. Также, был модернизирован скрипт авторизации в wordpress и добавлена таблица в базу данных MySQL. После этого на языке Python был написан собственный web-сервер, отображающий index.html.