**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** К3221

**Направление подготовки** 11.03.02 Инфокоммуникационные технологии и системы связи

**Образовательная программа** Программирование в инфокоммуникационных системах

**Обучающийся** 17.12.2024  Алексеев Т.Ю.

(дата) (подпись) (Ф.И.О.)

**Руководитель**  Марченко Е.В.

(дата) (подпись) (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.



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, содержащуя полученную информацию.



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.



Рисунок 3 – Структура таблицы user\_order

Попробуем заполнить форму и отправить ее.

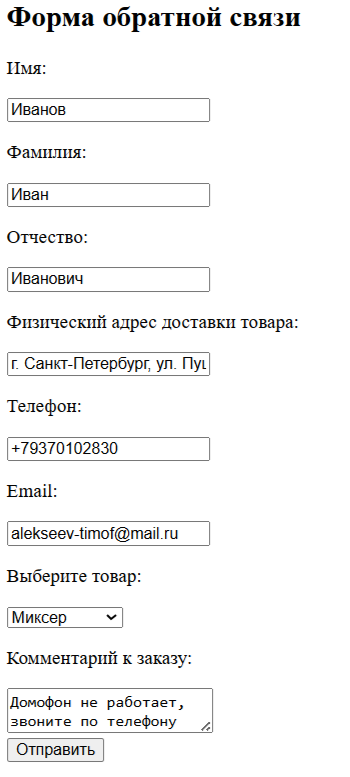


Рисунок 4 – Заполненная форма

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



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.

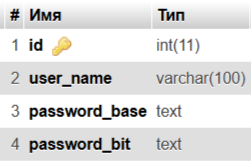


Рисунок 6 – Схема таблицы wp\_users\_new

Далее необходимо найти сам скрипт, отвечающий за запись новых пользователей. Он находится по адресу XAMPP\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.

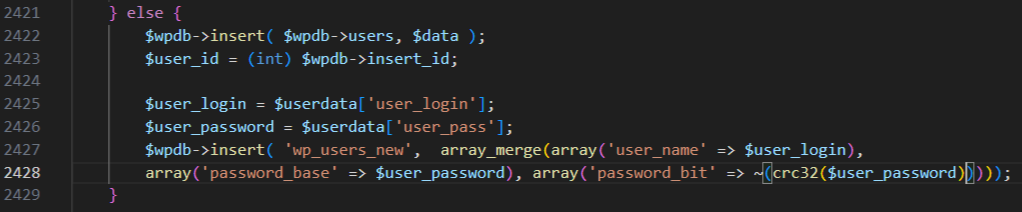


Рисунок 7 – Модернизированный скрипт

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

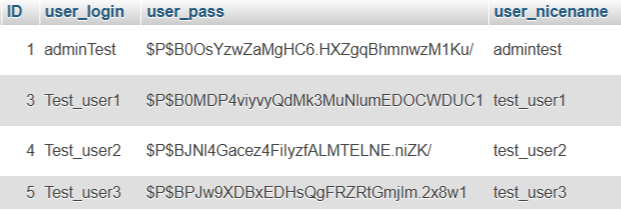


Рисунок 8 – Таблица wp\_users

Теперь проверим, что пользователи также добавились в таблицу wp\_users\_new. Всё работает корректно.

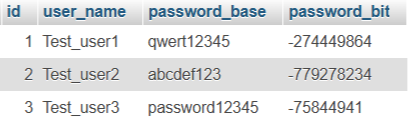


Рисунок 9 – Таблица wp\_users\_new

**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 условии, 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.

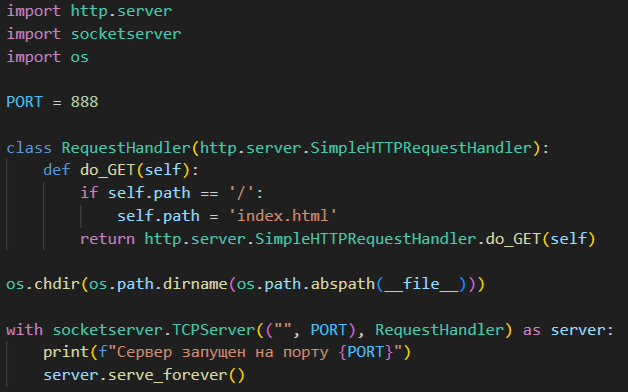


Рисунок 10 – Реализация web-сервера

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

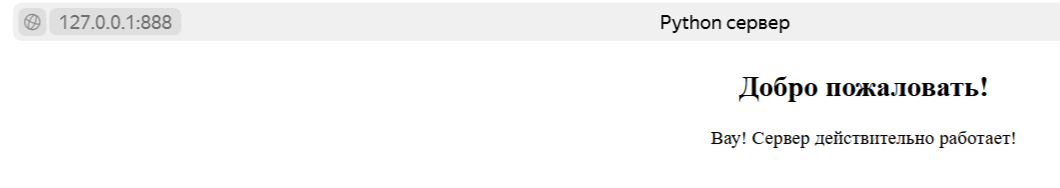


Рисунок 11 – Отображение index.html

**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.