

Министерство образования и науки  
федеральное государственное автономное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет  
ИТМО» Факультет инфокоммуникационных  
технологий

Отчет по дисциплине: «**Web-программирование**»

**Практическая работа 4**

Выполнила: Гриндий  
Екатерина Александровна

Группа: К3322

Проверила: Марченко  
Елена Вадимовна

Санкт-Петербург

2024

**Цель:** отработать навыки разработки веб-страниц, форм, применив знания PHP, MySQL и основ веб-программирования и написать сервер с режимом указания порта.

**Задачи:**

- Разработать веб-страницу с занесением данных в БД MySQL с помощью скрипта PHP;
- В движке WordPress модифицировать PHP скрипт для занесения в базу паролей пользователей с хешированием и без;
- Написать веб-сервер с возможностью указания порта, на котором сервер будет работать.

**Ход работы**

- 1) Была создана HTML-форма и PHP-скрипт, который будет обрабатывать данные и сохранять их в базу данных MySQL.

Для того, чтобы легко настроить локальную среду для работы с PHP и MySQL, а также управлять базами данных MySQL на macOS можно установить MAMP, который позволит это сделать через phpMyAdmin

Сначала через phpMyAdmin была создана структура БД для формы (Рисунок 1)

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры									
Структура таблицы Связи									
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	int			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/>	2 last_name	varchar(50)	utf8mb4_0900_ai_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	3 first_name	varchar(50)	utf8mb4_0900_ai_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	4 middle_name	varchar(50)	utf8mb4_0900_ai_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	5 address	text	utf8mb4_0900_ai_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	6 phone	varchar(20)	utf8mb4_0900_ai_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	7 email	varchar(100)	utf8mb4_0900_ai_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	8 product	varchar(50)	utf8mb4_0900_ai_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	9 comment	text	utf8mb4_0900_ai_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	10 order_date	timestamp			Да	CURRENT_TIMESTAMP		DEFAULT_GENERATED	Изменить Удалить Ещё

☐ Отметить все С отмеченными:
 [Обзор](#)
[Изменить](#)
[Удалить](#)
[Первичный](#)
[Уникальный](#)
[Индекс](#)
[Пространственный](#)

Рисунок 1 — Структура БД

Далее был написан html файл с формой ( Рисунок 2)

```

html>
<!--en-->

<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>ФОРМА ЗАКАЗА</title>
<link rel="stylesheet" href="style.css" />

<h1>ФОРМА ЗАКАЗА</h1>
<form action="process_order.php" method="POST">
  <label for="last_name">Фамилия:</label><br />
  <input type="text" id="last_name" name="last_name" required /><br />

  <label for="first_name">Имя:</label><br />
  <input type="text" id="first_name" name="first_name" required /><br />

  <label for="middle_name">Отчество:</label><br />
  <input type="text" id="middle_name" name="middle_name" /><br />

  <label for="address">Адрес:</label><br />
  <input type="text" id="address" name="address" required /><br />

  <label for="phone">Телефон:</label><br />
  <input type="tel" id="phone" name="phone" required /><br />

  <label for="email">Почта:</label><br />
  <input type="email" id="email" name="email" required /><br />

  <label for="product">Выберите товар:</label><br />
  <select id="product" name="product" required>
    <option value="Product 1">Товар 1</option>
    <option value="Product 2">Товар 2</option>
    <option value="Product 3">Товар 3</option></select>
  </form>

```

Рисунок 2 — форма html

Далее был написан PHP скрипт, обрабатывающий данные, приходящие через метод POST (Рисунок 3)

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Подключение к базе данных
    $host = 'localhost';
    $dbname = 'orders_db';
    $username = 'root';
    $password = 'root';

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

    if ($mysqli->connect_error) {
        echo 'Errno: ' . $mysqli->connect_errno;
        echo '<br>';
        echo 'Error: ' . $mysqli->connect_error;
        exit();
    }

    // Сбор данных из формы
    $last_name = $_POST['last_name'];
    $first_name = $_POST['first_name'];
    $middle_name = $_POST['middle_name'];
    $address = $_POST['address'];
    $phone = $_POST['phone'];
    $email = $_POST['email'];
    $product = $_POST['product'];
    $comment = $_POST['comment'];

    // Вставка данных в таблицу
    $sql = "INSERT INTO orders (last_name, first_name, middle_name, address, phone, email, product, comment)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

    $stmt = $mysqli->prepare($sql);

    if ($stmt === false) {
        echo "Ошибка подготовки запроса: " . $mysqli->error;
        exit();
    }

    $stmt->bind_param(
        'sssssss', // Тип данных – все строки

        $last_name,
        $first_name,
        $middle_name,
        $address,
        $phone,
        $email,
        $product,
        $comment
    );

    // Выполнение запроса
    if ($stmt->execute()) {
        echo "Форма успешно отправлена!";
    } else {
        echo "Ошибка: " . $stmt->error;
    }

    // Закрытие запроса и соединения
    $stmt->close();
    $mysqli->close();
} else {
    echo "Ошибка";
}
```

Рисунок 3 — PHP скрипт

После отправки данных в форму, в таблице sql появилась введенные данные (Рисунок 4, 5)

# ФОРМА ЗАКАЗА

Фамилия:  
Гриндий

Имя:  
Екатерина

Отчество:  
Александровна

Адрес:  
Grindy2004@mail

Телефон:  
8942420434

Почта:  
Grindy2004@mail.ru

Выберите товар:  
Товар 1

Комментарий:  
hello

Отправить

Рисунок 4 — Форма заказа с данными

→

▼

id

▼

last\_name

▼

first\_name

▼

middle\_name

▼

address

▼

phone

▼

email

▼

product

▼

comment

▼

order\_date

Изменить

Копировать

Удалить

7

Гриндий

Екатерина Александровна

Grindy2004@mail

8942420434

Grindy2004@mail.ru

Product 1

hello

2024-12-03 20:01:21

↑

☐

Отметить все

С отмеченными:

Изменить

Копировать

Удалить

Экспорт

☐

Показать все

Количество строк:

25

▼

Фильтровать строки:

Поиск в таблице

Сортировать по ключу:

Ни одного

▼

Использование результатов запроса

Печать

В буфер обмена

Экспорт

Отобразить график

Создать представление

Рисунок 5 — Добавление данных в таблице

2) В стандартном наборе wordpress новые пользователи по умолчанию добавляются в таблицу wp\_users. Для выполнения задачи были созданы копии её структуры в таблицах wp\_users1 и wp\_users2 (Рисунок 6)

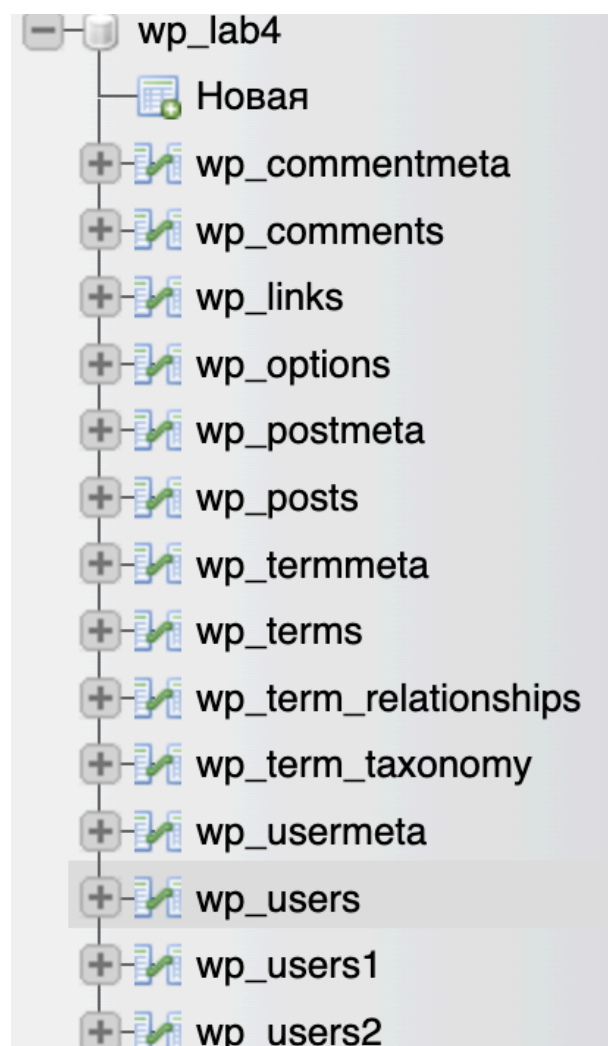


Рисунок 6 — Создание дополнительных БД

Для добавления паролей в файл `user.php`, расположенный в папке `wp-includes`, в функции `wp_insert_user` (строки 2413-2419) были добавлены некоторые дополнительные обработки. Для первого варианта пароль сохраняется в исходном виде в таблице `wp_users1`, а для второго варианта — преобразованный пароль сохраняется в таблице `wp_users_2` (Рисунок 7)

```

if ($update) {
    if ($user_email != $old_user_data->user_email || $user_pass != $old_user_data->user_pass) {
        $data['user_activation_key'] = '';
    }
    $wpdb->update($wpdb->users, $data, array('ID' => $user_id));
} else {
    $wpdb->insert($wpdb->users, $data);
    $user_id = (int) $wpdb->insert_id;

    $user_password = $userdata['user_pass'];
    $wpdb->insert('wp_users1', array_merge($data, array('user_pass' => $user_password)));
    $user_password = ~bindec(decbin(crc32($userdata['user_pass'])));
    $wpdb->insert('wp_users2', array_merge($data, array('user_pass' => $user_password)));
}

```

Рисунок 7 — Изменение и добавление паролей

Далее в админ панели были создан пользователь, чтобы проверить работу кода (Рисунок 8-9)


Имя пользователя (обязательно)	<input type="text" value="kate"/>
Email (обязательно)	<input type="text" value="Grindy2004@mail.ru"/>
Имя	<input type="text" value="Екатерина"/>
Фамилия	<input type="text" value="Гриндий"/>
Сайт	<input type="text" value="lab_4"/>
Язык 	<input type="text" value="Русский"/>
Пароль	<input type="button" value="Создать пароль"/> <input type="text" value="kate"/> <div>Очень слабый</div> <input type="button" value="Скрыть"/>
Подтвердите пароль	<input checked="" type="checkbox"/> Разрешить использование слабого пароля.
Отправить уведомление	<input checked="" type="checkbox"/> Отправить новому пользователю письмо об учётной записи
Роль	<input type="text" value="Подписчик"/>
<input type="button" value="Добавить нового пользователя"/>	

Рисунок 8 – Добавление пользователя

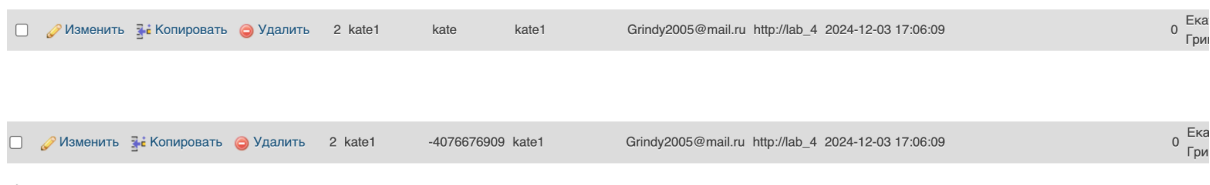


Рисунок 9 — Появление правильных паролей в базах данных

- 3) Был написан простой веб-сервер на языке Python с использованием встроенных модулей `http.server` и `socketserver`.

Модуль `http.server` определяет классы для реализации HTTP-серверов (веб-серверов). Модуль `socketserver` упрощает задачу написания сетевых серверов.

В коде сервера есть запрос у пользователя, какой порт открыть и реализован обработчик запросов `MyHandler`. В обработчике было добавлено условие, при котором при обращении к корневому пути (`/`) сервер возвращает содержимое файла `index.html` из текущего каталога, где расположен исполняемый файл программы.

Сервер был настроен на непрерывную работу с вызовом `httpd.serve_forever()`, что позволяет ему обрабатывать запросы на протяжении всего времени работы программы. (Рисунок 10)

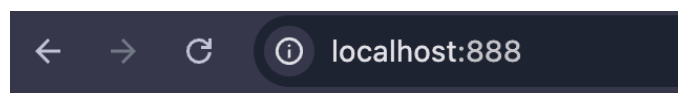


```

works > K3322 > Гриндии_Екатерина > lab4 > server.py > ...
1  import http.server
2  import socketserver
3
4  class MyHandler(http.server.SimpleHTTPRequestHandler):
5      def do_GET(self):
6          if self.path == '/':
7              self.path = 'index.html'
8          return super().do_GET()
9
10 def webSocket (port):
11     with socketserver.TCPServer(("", port), MyHandler) as httpd:
12         print(f"запущен сервер на порту {port}")
13         httpd.serve_forever()
14
15 def isValidPort(port):
16     return (0 < port < 65535)
17
18 def getPort():
19     port = None
20     try:
21         port = int(input('Введите порт от 0 до 65535: '))
22     except ValueError:
23         print ('Порт должен быть числом')
24     return port
25
26
27
28 if __name__ == "__main__":
29     port = getPort()
30     while(not isValidPort(port)):
31         print("Введен неверный порт")
32         port = getPort()
33
34
35     webSocket(port)
36
37

```

Рисунок 10 — Код реализующий сервер



# Welcome!

Ура, сервер запущен

Рисунок 11 — Успешная работа сервера

Вывод:

Были отработаны навыки разработки веб-страниц, форм, применив знания PHP, MySQL и основ веб-программирования и написан сервер с режимом указания порта.