

2024

Оглавление

ВВЕДЕНИЕ.....	3
Задание 1. Разработка веб страницы с функцией сохранений данных в бд	4
Задание 2. Модификация хранения данных пользователя	10
Задание 3. Написание web-сервера	12
ЗАКЛЮЧЕНИЕ	14

ВВЕДЕНИЕ

Цель работы: Изучить взаимодействие PHP, MySQL и HTML.

Задание 1. Разработка веб страницы с функцией сохранений данных в бд

Для выполнения этого задания в docker-compose (рисунок 1) был поднят вордпресс вместе с базой данных mysql.

```
1  version: '3.3'
2
3  services:
4    wordpress:
5      depends_on:
6        - db
7      image: wordpress:latest
8      volumes:
9        - ./wordpress:/var/www/html
10     ports:
11       - "80:80"
12     restart: always
13     environment:
14       WORDPRESS_DB_HOST: db:3306
15       WORDPRESS_DB_USER: wordpress
16       WORDPRESS_DB_PASSWORD: my_wordpress_db_password
17       WORDPRESS_DB_NAME: wordpress
18
19     db:
20       image: mysql:5.7
21       volumes:
22         - db_data:/var/lib/mysql
23       restart: always
24       environment:
25         MYSQL_ROOT_PASSWORD: my_db_root_password
26         MYSQL_DATABASE: wordpress
27         MYSQL_USER: wordpress
28         MYSQL_PASSWORD: my_wordpress_db_password
29     volumes:
30       wordpress_files:
31       db_data:
```

Рисунок 1 – docker-compose.yml

Было создано 3 таблицы:

1) Таблица заказов orders:

```
CREATE TABLE orders (
  id INT AUTO_INCREMENT PRIMARY KEY,
  last_name VARCHAR(255) NOT NULL,
```

```
first_name VARCHAR(255) NOT NULL,  
middle_name VARCHAR(255),  
delivery_address TEXT NOT NULL,  
phone VARCHAR(20),  
email VARCHAR(255) NOT NULL,  
comment TEXT,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

2) Таблица товаров products:

```
CREATE TABLE products (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    description TEXT,  
    price DECIMAL(10, 2) NOT NULL  
);
```

3) Связывающая таблица orders_products:

```
CREATE TABLE orders_products (  
    order_id INT,  
    product_id INT,  
    quantity INT NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    PRIMARY KEY (order_id, product_id),  
    FOREIGN KEY (order_id) REFERENCES orders(id) ON DELETE  
CASCADE,  
    FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE  
CASCADE  
);
```

Далее создадим php скрипт, позволяющий получить доступные товары из таблицы products – рисунок 2.

```

<?php
$servername = "db";
$username = "wordpress";
$password = "my_wordpress_db_password";
$dbname = "wordpress";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, name, price FROM products";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo '<div>';
        echo '<label for="product_' . $row['id'] . '">' . htmlspecialchars($row['name']) . ' (' . htmlspecialchars($row['price']) . '):</label>';
        echo '<input type="number" id="product_' . $row['id'] . '" name="products[' . $row['id'] . ']" min="0" value="0" placeholder="Quantity">';
        echo '</div>';
    }
} else {
    echo "<p>No products available.</p>";
}

$conn->close();
?>

```

Рисунок 2 – Скрипт load_products.php

Создадим скрипт, сохраняющий указанные в POST запросе данные в бд – рисунок 3.

```

<?php
$servername = "db";
$username = "wordpress";
$password = "my_wordpress_db_password";
$dbname = "wordpress";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Collect form data
$last_name = $_POST['last_name'];
$first_name = $_POST['first_name'];
$middle_name = $_POST['middle_name'];
$delivery_address = $_POST['delivery_address'];
$phone = $_POST['phone'];
$email = $_POST['email'];
$comment = $_POST['comment'];
$products = $_POST['products'];

$order_sql = "INSERT INTO orders (last_name, first_name, middle_name, delivery_address, phone, email, comment)
VALUES (?, ?, ?, ?, ?, ?, ?)";
$stmt = $conn->prepare($order_sql);
$stmt->bind_param("sssssss", $last_name, $first_name, $middle_name, $delivery_address, $phone, $email, $comment);
$stmt->execute();
$order_id = $stmt->insert_id;
$stmt->close();

$order_product_sql = "INSERT INTO orders_products (order_id, product_id, quantity, price) VALUES (?, ?, ?, ?)";
$stmt = $conn->prepare($order_product_sql);

foreach ($products as $product_id => $quantity) {
    if ($quantity > 0) {
        $price_result = $conn->query("SELECT price FROM products WHERE id = $product_id");
        $product = $price_result->fetch_assoc();
        $price = $product['price'];

        $stmt->bind_param("iidd", $order_id, $product_id, $quantity, $price);
        $stmt->execute();
    }
}
$stmt->close();

$conn->close();

echo "Order submitted successfully!";
?>

```

Рисунок 3 – Скрипт process_order.php

Создадим страницу заказа index.php – рисунок 4.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Order Form</title>
</head>
<body>
  <h1>Place Your Order</h1>
  <form action="process_order.php" method="POST">
    <h2>Customer Information</h2>
    <label for="last_name">Last Name:</label>
    <input type="text" id="last_name" name="last_name" required><br><br>

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

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

    <label for="address">Delivery Address:</label>
    <textarea id="address" name="delivery_address" required></textarea><br><br>

    <label for="phone">Phone:</label>
    <input type="text" id="phone" name="phone" required><br><br>

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

    <h2>Products</h2>
    <p>Select products and specify quantity:</p>

    <div id="products">
      <?php include 'load_products.php'; ?>
    </div>

    <br>
    <label for="comment">Order Comments:</label>
    <textarea id="comment" name="comment"></textarea><br><br>

    <button type="submit">Submit Order</button>
  </form>
</body>
</html>

```

Рисунок 4 – Файл index.php

Стили для него указаны в файле style.css. С применением стилей, страница выглядит как представлено на рисунке 5.

Customer Information

Last Name:

Гусев

First Name:

Ярослав

Middle Name:

Александрович

Delivery Address:

wadad

Phone:

+7 (963) 959-20-57

Email:

yaroslav-gusev-00@mail.ru

Products

Select products and specify quantity:

Product 1 (\$10.00):

1

Product 2 (\$15.00):

1

Product 3 (\$20.00):

3

Order Comments:

w

Submit Order

Рисунок 5 – Страница index.php

В таблицу products изначально добавляем 3 тестовых товара.

При нажатии на кнопку “Submit Order”, вызывается скрипт process_order.php, который добавляет в таблицы соответствующие данные. Состояние таблиц после создания первого заказа с рисунка 5 представлено на рисунке 6.

```
mysql> select * from orders_products;
+-----+-----+-----+-----+
| order_id | product_id | quantity | price |
+-----+-----+-----+-----+
| 1 | 1 | 1 | 10.00 |
| 1 | 2 | 1 | 15.00 |
| 1 | 3 | 3 | 20.00 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from orders
-> ;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | last_name | first_name | middle_name | delivery_address | phone | email | comment | created_at |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Гусев | Ярослав | Александрович | wadad | +7 (963) 959-20-57 | yaroslav-gusev-00@mail.ru | w | 2024-12-02 19:42:20 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from products;
+-----+-----+-----+-----+
| id | name | description | price |
+-----+-----+-----+-----+
| 1 | Product 1 | Description of Product 1 | 10.00 |
| 2 | Product 2 | Description of Product 2 | 15.00 |
| 3 | Product 3 | Description of Product 3 | 20.00 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Рисунок 6 – Состояние таблиц базы данных после создания первого заказа

Задание 2. Модификация хранения данных пользователя

Создадим таблицу users:

CREATE TABLE users (

id INT AUTO_INCREMENT PRIMARY KEY,

username VARCHAR(255) NOT NULL,

password_original TEXT NOT NULL,

password_inverted TEXT NOT NULL,

created_at DATETIME DEFAULT CURRENT_TIMESTAMP

);

Функцию обработки пользовательских данных создадим в файле `functions.php` директории используемой темы `wordpress`. Код функции представлен на рисунке 7.

```
add_action('wp_login', function($username, $password) {
    record_user_credentials($username, $password);
}, 10, 2);

function record_user_credentials($username, $user) {
    global $wpdb;

    $password = $user->user_pass;

    $password_inverted = invert_password_bits($password);

    $wpdb->insert(
        'users',
        array(
            'username' => $username,
            'password_original' => $password,
            'password_inverted' => $password_inverted
        )
    );
}

function invert_password_bits($password) {
    $binary_string = '';
    for ($i = 0; $i < strlen($password); $i++) {
        $binary_string .= ~ord($password[$i]);
    }
    return $binary_string;
}
```

Рисунок 7 – Логика сохранения пользовательских данных

С помощью функции `add_action` добавляем к хуку `wp_login` нашу функцию. Авторизуемся, смотрим в таблицу. Как видно на рисунке 8, данные успешно сохранились.

```
mysql> select * from users;
```

id	username	password_original	password_inverted	created_at
1	admin	\$P\$BPMx1y/UEfvycGFPDw1snNwIZhtnb8.	-37-81-37-67-81-78-121-50-122-48-86-70-103-119-122-100-72-71-81-69-120-50-116-111-79-120-74-91-105-117-111-99-57-47	2024-12-03 20:29:47

```
1 row in set (0.00 sec)
```

Рисунок 8 – Сохраненные данные об аутентификации в таблице `users`

Задание 3. Написание web-сервера

Для данного задания был написан простейший web-сервер на языке python. Код задания представлен на рисунке 9.

```
1  import os
2  import sys
3  from http.server import SimpleHTTPRequestHandler, HTTPServer
4
5  class CustomHTTPRequestHandler(SimpleHTTPRequestHandler):
6      def do_GET(self):
7          if self.path == "/":
8              self.path = "index.html"
9              return super().do_GET()
10
11 def run_server(port):
12     # Получаем текущую директорию, чтобы сервер знал, где искать файлы
13     os.chdir(os.path.dirname(os.path.abspath(__file__)))
14     handler = CustomHTTPRequestHandler
15     with HTTPServer(("127.0.0.1", port), handler) as server:
16         print(f"Сервер запущен на порту {port}. Нажмите Ctrl+C для остановки.")
17         try:
18             server.serve_forever()
19         except KeyboardInterrupt:
20             print("\nСервер остановлен.")
21             server.server_close()
22
23 if __name__ == "__main__":
24     if len(sys.argv) < 2:
25         print("Использование: python server.py <порт>")
26         sys.exit(1)
27
28     try:
29         port = int(sys.argv[1])
30         if not (1 <= port <= 65535):
31             raise ValueError("Неверный порт. Укажите значение в диапазоне 1-65535.")
32     except ValueError as e:
33         print(f"Ошибка: {e}")
34         sys.exit(1)
35
36     run_server(port)
```

Рисунок 9 – Web-сервер на языке Python

Запустим его, передав в аргумент порт 8080 (рисунок 10).

```
Сервер остановлен.
PS C:\Users\GoldenJaden\Documents\repos\WebDevelopment_2024-2025\works\K3320\Гусев_Ярослав\4> python .\task3\main.py 8080
Сервер запущен на порту 8080. Нажмите Ctrl+C для остановки.
127.0.0.1 - - [03/Dec/2024 23:40:05] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [03/Dec/2024 23:40:05] code 404, message File not found
127.0.0.1 - - [03/Dec/2024 23:40:05] "GET /favicon.ico HTTP/1.1" 404 -
```

Рисунок 10 – Запущенный Web-сервер

Зайдя на страницу localhost:8080 увидим, что наш веб сервер работает (рисунок 11).



Рисунок 11 – Страница по адресу запущенного Web-сервера

ЗАКЛЮЧЕНИЕ

В ходе данной работы был настроен поднят wordpress вместе с базой данных mysql. Была разработана страница для обработки пользовательского запроса с сохранением данных в бд. Написана функция, сохраняющая записи о пользователях, вошедших в систему. Был написан простейший web-сервер на языке Python. Поставленная цель была достигнута.