

Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №4

Выполнила: Ульянова Е.А.

Проверила: Марченко Е.В.

Санкт-Петербург,

2024

Оглавление

Цель работы	3
Ход работы.....	3
1. Разработка веб-страницы для заказа товара.....	3
2. Сохранение логина и пароля пользователя в отдельную новую таблицу ..	8
3. Написание веб-сервера	10
Заключение	12

Цель работы

Целью данной лабораторной работы является приобретение знаний, необходимых для сохранения данных, получаемых из форм, в БД и написания web-сервера.

Ход работы

1. Разработка веб-страницы для заказа товара

Для выполнения первого задания необходимо разработать веб-страницу, на которой пользователь может оставить данные о себе – фамилия, имя, отчество, адрес для физической доставки товара, телефон, адрес электронной почты. Товары покупатель выбирает из списка. Также необходимо предусмотреть отдельную позицию, в которой пользователь может оставить комментарий по заказу. Данные, введенные в форму, должны обрабатываться php скриптом и заноситься в таблицу БД MySQL Server.

Для реализации задания были созданы файлы form.html, по своей структуре напоминающий файл с формой из лабораторной №3 (представлен на рисунке 1), style.css (представлен на рисунке 2) и process.php, предназначенный для сохранения данных из формы в БД посредством подключения к ней, получения данных из POST-запроса и выполнения sql-запроса (код представлен на рисунке 3).

```
index.html x form.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Заказ товара</title>
7 <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10 <h1>Форма заказа товара</h1>
11 <form action="process.php" method="POST">
12 <label for="surname">Фамилия:</label><br />
13 <input type="text" id="surname" name="surname" required><br /><br />
14
15 <label for="name">Имя:</label><br />
16 <input type="text" id="name" name="name" required><br /><br />
17
18 <label for="patronimic">Отчество:</label><br />
19 <input type="text" id="patronimic" name="patronimic"><br /><br />
20
21 <label for="address">Адрес доставки:</label><br />
22 <textarea id="address" name="address" rows="3" required></textarea><br /><br />
23
24 <label for="phone">Телефон:</label><br />
25 <input type="tel" id="phone" name="phone" required><br /><br />
26
27 <label for="email">Электронная почта:</label><br />
28 <input type="email" id="email" name="email" required><br /><br />
29
30 <label for="product">Выберите товар:</label><br />
31 <select id="product" name="product" required>
32 <option value="product1">Собащее сердце</option>
33 <option value="product2">Мёртвые души</option>
34 <option value="product3">Портрет Дориана Грея</option>
35 </select><br /><br />
36
37 <label for="comments">Комментарий к заказу:</label><br />
38 <textarea id="comments" name="comments" rows="3"></textarea><br /><br />
39
40 <button type="submit">Оформить заказ</button>
41 </form>
42 </body>
43 </html>
```

Рисунок 1 - form.html

```

C: > MAMP > htdocs > task1_lab4 > # style.css > ...
1  body {
2      background-color: #f0e3f6;
3  }
4
5  h1 {
6      text-align: center;
7  }
8
9  form {
10     max-width: 400px;
11     margin: 20px auto;
12     padding: 20px;
13     background-color: #e3c3f3;
14     border: 1px solid #ddd;
15     border-radius: 8px;
16     box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
17 }
18
19 form input[type="text"], form input[type="email"], form textarea {
20     width: 100%;
21     padding: 10px;
22     margin-bottom: 15px;
23     border: 1px solid #ccc;
24     border-radius: 4px;
25     font-size: 16px;
26     font-family: Arial, sans-serif;
27     box-sizing: border-box;
28 }
29
30 form input:focus, form textarea:focus {
31     border-color: #08eb89;
32     outline: none;
33     box-shadow: 0 0 5px rgba(30, 233, 176, 0.381);
34 }
35
36 form button[type="submit"], form input[type="submit"] {
37     width: 100%;
38     padding: 10px;
39     background-color: white;
40     border: none;
41     border-radius: 4px;
42     color: black;
43     font-size: 16px;
44     cursor: pointer;
45     transition: background-color 0.3s;
46 }
47
48 form button[type="submit"]:hover{
49     background-color: rgba(30, 233, 176, 0.381);
50 }

```

Рисунок 2 - style.css

```

C: > MAMP > htdocs > task1_lab4 > process.php
1  <?php
2  $host = 'localhost';
3  $dbname = 'orders_db';
4  $username = 'root';
5  $password = 'root';
6  $port = 3306;
7
8  try {
9      $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);
10     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11 } catch (PDOException $e) {
12     die("Ошибка подключения к базе данных: " . $e->getMessage());
13 }
14
15 $surname = $_POST['surname'];
16 $name = $_POST['name'];
17 $patronimic = $_POST['patronimic'];
18 $address = $_POST['address'];
19 $phone = $_POST['phone'];
20 $email = $_POST['email'];
21 $product = $_POST['product'];
22 $comments = $_POST['comments'];
23
24 $sql = "INSERT INTO orders (surname, name, patronimic, address, phone, email, product, comments)
25     VALUES (:surname, :name, :patronimic, :address, :phone, :email, :product, :comments)";
26
27 $stmt = $pdo->prepare($sql);
28
29 try {
30     $stmt->execute([
31         ':surname' => $surname,
32         ':name' => $name,
33         ':patronimic' => $patronimic,
34         ':address' => $address,
35         ':phone' => $phone,
36         ':email' => $email,
37         ':product' => $product,
38         ':comments' => $comments,
39     ]);
40     echo "Заказ успешно сохранен!";
41 } catch (PDOException $e) {
42     die("Ошибка выполнения запроса: " . $e->getMessage());
43 }
44 ?>

```

Рисунок 3 - process.php

Также по ссылке <http://localhost/phpMyAdmin5/> была создана база данных order.db и выполнен sql-запрос по созданию таблице. Синтаксис представлен на рисунке 4.

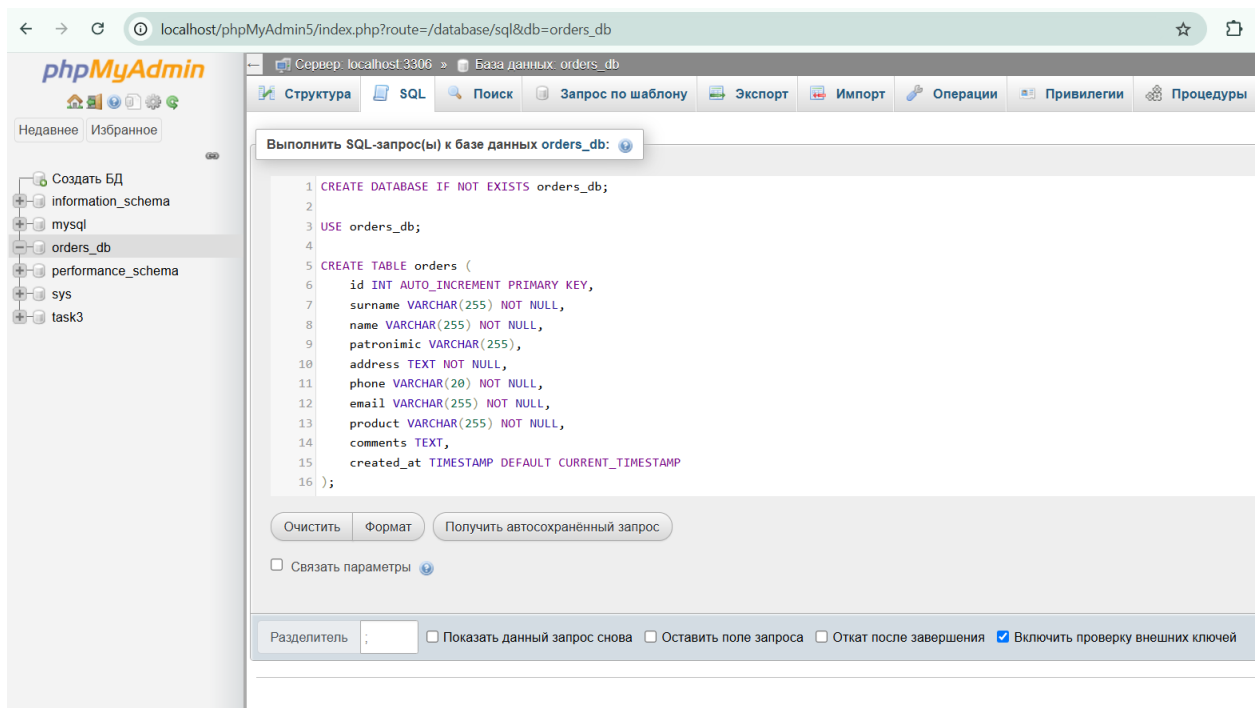


Рисунок 4 - Запрос для создания таблицы в БД

Для тестирования, как и в прошлой лабораторной работе, был использован МАМР. Внешний вид формы и результат сохранения данных при её отправке представлены на рисунках 5 и 6.

localhost/task1_lab4/form.html

Форма заказа товара

Фамилия:
Ульянова

Имя:
Екатерина

Отчество:
Алексеевна

Адрес доставки:
Санкт-Петербург, ул. Ломоносова, 9

Телефон:
+7 (921) 199-8818

Электронная почта:
Oulyanova.ekaterina.4@gmail.com

Выберите товар:
Портрет Дориана Грея

Комментарий к заказу:

Оформить заказ

Рисунок 5 - Страница для заказа

+ Параметры

			id	surname	name	patronymic	address	phone	email	product	comments	created_at	
<input type="checkbox"/>	Изменить	Копировать	Удалить	1	Ульянова	Екатерина	Алексеевна	Санкт-Петербург, ул. Большая Морская 5	+7 (921) 199-8818	Oulyanova.ekaterina.4@gmail.com	product2	Доставить не позднее 22.00.	2024-12-13 15:37:06
<input type="checkbox"/>	Изменить	Копировать	Удалить	2	Ульянова	Екатерина	Алексеевна	Санкт-Петербург, ул. Ломоносова, 9	+7 (921) 199-8818	Oulyanova.ekaterina.4@gmail.com	product3		2024-12-13 15:38:39

↑

☐ Отметить все

С отмеченными:

Изменить

Копировать

Удалить

Экспорт

Рисунок 6 - Записи в БД

2. Сохранение логина и пароля пользователя в отдельную новую таблицу

В движке wordpress используется форма для авторизации пользователей – пользователь вводит логин и пароль. Логин и хэш от пароля записываются в таблицу БД MySQL. Для выполнения второго задания требуется модифицировать php скрипт, чтобы дополнительно логин и пароль записывались в отдельную новую таблицу в БД MySQL. Необходимо предусмотреть два варианта записи: а) пароль записывается в исходном виде, б) инвертируются биты бат пароля (ноль меняется на единицу, единица меняется на ноль).

Для реализации поставленных задач был создан плагин plugin_for_saving_login_and_password. PHP-файл для него представлен на рисунке 7. Он содержит в себе блок описания плагина и две функции для создания таблицы БД при его активации, а также для добавления записи в неё при наступлении события wp_login, которое происходит после успешной аутентификации пользователя.


```

1  <?php
2  /**
3   * Plugin Name: plugin_for_saving_login_and_password
4   * Description: В таблице будут сохранены две версии пароля: в исходном и инвертированном виде
5   * Version: 1.0
6   * Author: Екатерина
7   */
8
9  function create_table() {
10     global $wpdb;
11     $table_name = $wpdb->prefix . 'auth_logs';
12
13     $charset_collate = $wpdb->get_charset_collate();
14
15     $sql = "CREATE TABLE $table_name (
16         id mediumint(9) NOT NULL AUTO_INCREMENT,
17         username varchar(60) NOT NULL,
18         hashed_password varchar(255) NOT NULL,
19         inverted_hash varchar(255) NOT NULL,
20         time_of_auth TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
21         PRIMARY KEY (id)
22     ) $charset_collate;";
23
24     require_once(ABSPATH . 'wp-admin/includes/upgrade.php');
25     dbDelta($sql);
26 }
27 register_activation_hook(__FILE__, 'create_table');
28
29 function log_auth($user_login, $user) {
30     global $wpdb;
31     $table_name = $wpdb->prefix . 'auth_logs';
32     $hashed_password = $user->user_pass;
33     $binary_hash = unpack('H*', $hashed_password)[1];
34     $inverted_binary = strrev($binary_hash);
35     $wpdb->insert(
36         $table_name,
37         [
38             'username' => $user_login,
39             'hashed_password' => $hashed_password,
40             'inverted_hash' => $inverted_binary
41         ]
42     );
43 }
44 add_action('wp_login', 'log_auth', 10, 2);

```

Рисунок 7 - Код для создания плагина

После создания нового плагина, необходимо перейти в раздел плагины и активировать его (рисунок 8).

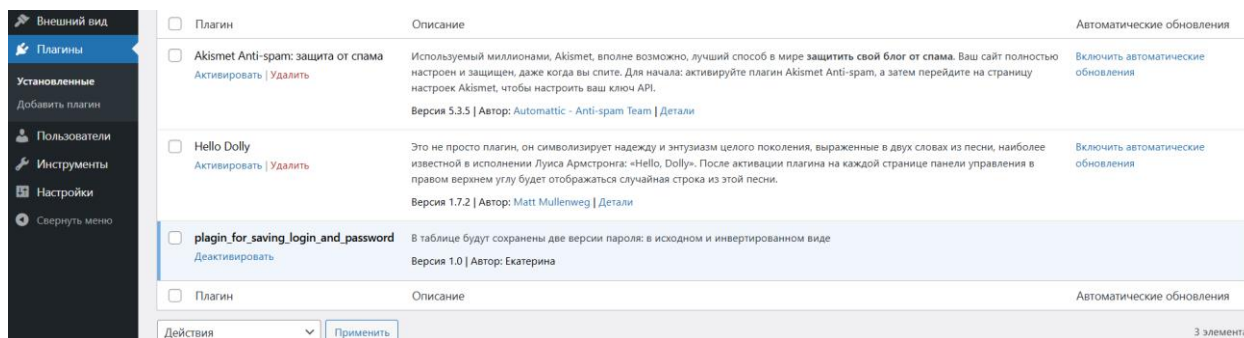


Рисунок 8 - Активация плагина

Как и описывалось выше, при активации плагина автоматически будет создана таблица для хранения логина, пароля в двух вариациях, а также времени входа пользователя. Записи таблицы после успешной аутентификации представлены на рисунке 9.

+ Параметры

	id	username	hashed_password	inverted_hash	time_of_auth
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	4	Ekaterina	\$P\$B5iNjbXYnMnAEuvFDGFwzjDL1r0.H	e284e2032713e2c444a6a7776474446467575414e6d4e69585...	2024-12-13 16:25:25

↑ ☐ Отметить все С отмеченными: ☐ Изменить ☐ Копировать ☐ Удалить ☐ Экспорт

Рисунок 9 - Таблица с данными по аутентификации

3. Написание веб-сервера

Для выполнения третьего задания необходимо выбрать среду разработки (язык программирования) и, используя её написать web-сервер. В программе требуется предусмотреть возможность указать порт, на котором будет работать сервер. При обращении вида `http://127.0.0.1:888/` где 888 – заданный порт, сервер должен вернуть содержимое файла `index.html`, находящегося в том же каталоге, что и исполняемый файл написанной программы-сервера.

Для данной задачи было принято решение об использовании языка Python с модулями, предоставляющими специальные классы для создания серверов (`http.server` и `socketserver`). Код для создания запрашиваемого web-сервера, работающего до момента его остановки из консоли, представлен на рисунке 10.

```

C: > Users > Ekaterina > Desktop > Учебные задания > 3 курс > Web > Lab4 > task_3 > server.py > ...
1  import http.server
2  import socketserver
3
4  PORT = 888
5
6  class MyHandler(http.server.SimpleHTTPRequestHandler):
7      def do_GET(self):
8          if self.path == '/':
9              self.path = 'index.html'
10             return super().do_GET()
11
12  if __name__ == "__main__":
13      try:
14          with socketserver.TCPServer(("", PORT), MyHandler) as httpd:
15              print(f"Сервер запущен на порту {PORT}")
16              httpd.serve_forever()
17      except KeyboardInterrupt:
18          print("\nСервер остановлен пользователем")

```

Рисунок 10 - Код для создания web-сервера

Запуск из консоли и просмотр ранее созданной html-страницы по адресу <http://127.0.0.1:888/> представлены на рисунках 11 и 12.

```

C:\Users\Ekaterina\Desktop\Учебные задания\3 курс\Web\Lab4\task_3>py server.py
Сервер запущен на порту 888
127.0.0.1 - - [13/Dec/2024 16:47:13] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Dec/2024 16:47:13] "GET /style.css HTTP/1.1" 200 -

```

Рисунок 11 - Запуск веб-сервера

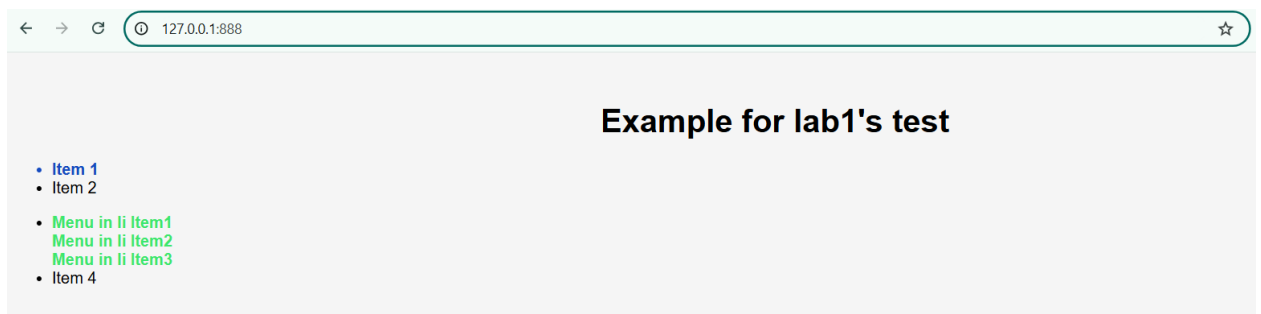


Рисунок 12 - Просмотр страницы

Заключение

В ходе данной лабораторной работы были приобретены знания по созданию таблиц и внесению изменений в них при отправке формы или аутентификации пользователя. Произведено это было с помощью sql-запросов и php-скриптов. Также с использованием Python был создан собственный веб-сервер, возвращающий содержимое файла index.html, находящегося в том же каталоге, что и исполняемый файл написанной программы-сервера.