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 SQL и настройка веб-сервера						
Обучающийся Адрат Олеся Александровна						
Факультет Факультет инфокоммуникационных технологий						
Группа К3321						
Направление подготовки 11.03.02 Инфокоммуникационные технологии и системы связи						
Образовательная программа Программирование в инфокоммуникационных системах						
Обучающийся		(подпись)	<u>Адрат О.А.</u> (Ф.И.О.)			
Руководитель	(дата)	(подпись)	<u>Марченко Е.В.</u> (Ф.И.О.)			

Санкт-Петербург 2024 г.

СОДЕРЖАНИЕ

		Стр.
BI	ВЕДЕНИЕ	3
1	Задание 1	4
2	Задание 2	7
3	Задание 3	10
3	АКЛЮЧЕНИЕ	12

ВВЕДЕНИЕ

Цель работы:

Получить навыки разработки веб-страниц с отправкой данных формы на сервер и записью в базу данных, навыки написания php-скриптов с логической составляющей, написать собственный веб-сервер

Задачи:

- 1. Разработать веб-страницу, на которой пользователь может оставить данные о себе, реализовать запись данных в SQL
- 2. Модифицировать php скрипт, чтобы дополнительно логин и пароль записывались в отдельную новую таблицу в захэшированном виде
- 3. Написать собственный web-сервер с возможностью выбрать порт

1 Задание 1

В первом задании следовало создать страницу с формой, где пользователь вводит данные о себе.

После отправки данных, информация о пользователе должна записываться в отдельную SQL таблицу. Для начала была создана структура самой таблицы(рис. 1.1)



Рисунок 1.1 — Структура таблицы SQL

После этого была написана форма, куда пользователь вносит данные о себе для оформления заказа(рис. 1.2)

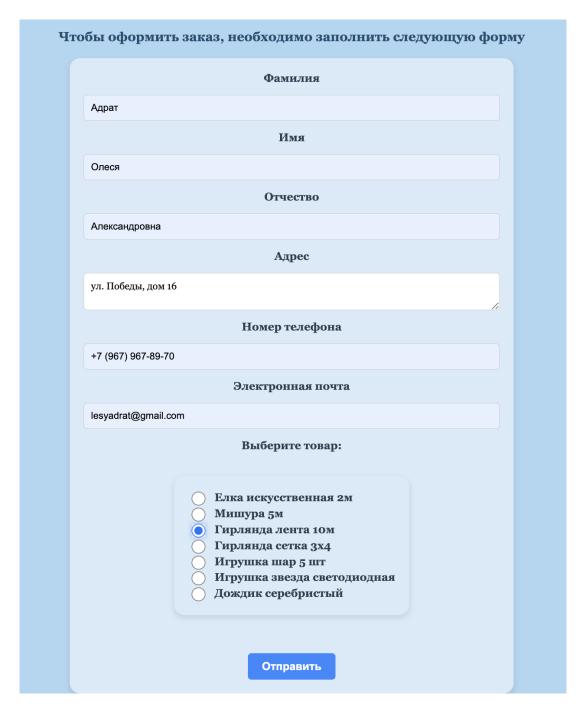


Рисунок $1.2 - \Phi$ орма для получения данных о пользователе

При отправке данных (нажатии на кнопку «Отправить») на сервер отправляется POST-запрос, после чего введенные данные можно увидеть в таблице, созданной на первом шаге (рис. 1.3)



Рисунок 1.3 - Проверка наличия записи данных в таблице

2 Задание 2

Во втором задании следовало написать php-скрипт, который бы заносил пароли авторизованных пользователей в отдельную таблицу, в обычном виде и захэшированном.

Для начала был написан сам скрипт с созданием нового плагина в движке wordpress(рис. 2.1)

```
Plugin Name: my_plugin-for-lab4
Description: Задание 2 Лабораторная 4
Version: 1.0
Author: Адрат Олеся
function invert($input) {
    $bin = '';
    for ($i = 0; $i < strlen($input); $i++) {</pre>
        $bin .= decbin(ord($input[$i]));
    $bin_inv = '';
    for ($i = 0; $i < strlen($bin); $i++) {
        $bin_inv .= $bin[$i] == '0' ? '1' : '0';
    return wp_hash_password(bindec($bin_inv));
}
function lab4_task2($username) {
    global $wpdb;
    $password1 = $_POST['pwd'];
    $password2 = invert($password1);
    $wpdb->insert(
        'wp_users_passwords',
            'username' => $username,
            'password1' => $password1,
            'password2' => $password2,
add_action('wp_login', 'lab4_task2');
```

Рисунок 2.1 — РНР-скрипт

После этого на wp-admin был активирован созданный плагин(рис. 2.2)

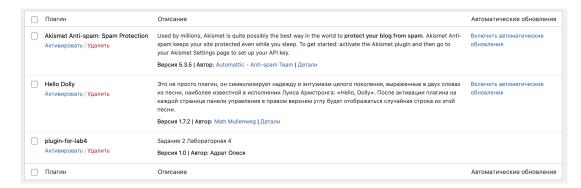


Рисунок 2.2 — Активация плагина

Далее был добавлен новый пользователь, чтобы проверить, что при авторизации под этим логином и паролем, пароль записывается в таблицу wpusers-passwords в обычном и захешированном виде. (рис. 2.3)

Имя пользователя (обязательно)	angel	
Email (обязательно)	angel@angel.com	
Имя		
Фамилия		
Сайт		
Язык 🕰	Основной язык сайта 🗸	
Пароль	Создать пароль	
	frt0p0407 Средний	% Скрыть

Рисунок 2.3 — Добавление нового пользователя

Затем произошла авторизация под логином нового пользователя, результат можно наблюдать в таблице SQL(рис. 2.4). Пароль хранится в обычном и захэшированном виде, как и требовалось.



Рисунок 2.4- Проверка добавления пароля пользователя в таблицу

3 Задание 3

В следующем задании нужно было написать собственный web-сервер. Для этого был выбран язык программирования python.

Был написан сервер(рис. 3.1) и его клиент для обработки запросов(рис. 3.2)

```
import http.server
import socketserver

# Define the folder to serve static files from
FOLDER_TO_SERVE = "/Users/olesaadrat/Desktop/itmo/5 semester/web/lab4/task3/page"

# Define the port on which the server will run
PORT = 8000

def main():
    class CustomHandler(http.server.SimpleHTTPRequestHandler):
        def __init__(self, *args, **kwargs):
            # Specify the directory to serve
            super().__init__(*args, directory=FOLDER_TO_SERVE, **kwargs)

# Set up the HTTP server
    with socketserver.TCPServer(("", PORT), CustomHandler) as httpd:
        print(f"Serving files from {FOLDER_TO_SERVE} on port {PORT}")
        httpd.serve_forever()

if __name__ == '__main__':
    main()
```

Рисунок 3.1 — Код web-сервера

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.connect(('localhost', 8000)) # Подключаемся к нашему серверу.
s.sendall('Lab4 is done!'.encode('utf-8')) # Отправляем фразу.
data = s.recv(1024) #Получаем данные из сокета.
s.close()
```

Рисунок 3.2 — Код клиента web-сервера

После этого была написана небольшая web-страница и запущен наш сервер(рис. 3.3)

```
olesaadrat@MacBook-Air-Olesa-3 task3 % python3 socket_server.py
Serving files from /Users/olesaadrat/Desktop/itmo/5 semester/web/lab4/task3/page on port 8000
127.0.0.1 - - [05/Jan/2025 22:46:54] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [05/Jan/2025 22:46:54] "GET /style.css HTTP/1.1" 200 -
127.0.0.1 - - [05/Jan/2025 22:46:54] "GET /snake.png HTTP/1.1" 200 -
```

Рисунок 3.3 — Запуск сервера

После перехода на http://localhost:8000 в браузере, можно наблюдать html-страницу, которая была написана ранее (рис. 3.4)



Рисунок 3.4 — Результат запуска сервера

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы была настроена запись данных из формы в базу данных, запись пароля в двух видах в базу данных, а также был написан собственный веб-сервер для обработки запросов и web-страница к нему.