

## ЛАБОРАТОРНА РОБОТА № 6

### Асинхронні запити на сервер та їх обробка

**Мета:** ознайомитися з асинхронними запитами на сервер та їх обробкою.

#### Хід роботи:

Завдання 1: Система керування користувачами.

Створіть веб-додаток для системи керування користувачами, який використовує асинхронні запити для взаємодії між клієнтом та сервером. У цьому завданні передбачено дві основні функціональності: реєстрація нового користувача та перегляд списку користувачів.

#### 1. Реєстрація користувача (Frontend):

- Створіть HTML-форму для реєстрації нового користувача з полями для ім'я, електронної адреси та пароля.
- Використовуйте JavaScript для збору даних з форми та відправки їх на сервер за допомогою fetch API.
- Перевірте, чи всі поля заповнені перед відправкою на сервер.

#### 2. Реєстрація користувача (Backend):

- Створіть PHP-скрипт для обробки запиту на реєстрацію нового користувача.
- Перевірте правильність введених даних, такі як унікальність електронної адреси та мінімальна довжина пароля.
- Збережіть дані нового користувача в базі даних MySQL.

#### 3. Отримання списку користувачів (Frontend):

- Створіть кнопку або посилання для відображення списку користувачів.
- Використовуйте JavaScript для відправки запиту на сервер для отримання списку користувачів.

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.08.000 – Пр6		
Розроб.	Дзінзілевич Д.О.						
Перевір.	Ковтун В.В.						
Керівник							
Н. контр.							
Зав. каф.							
Звіт з лабораторної роботи					Lіт.	Арк.	Аркушів
						1	37
					ФІКТ Гр. ІПЗ-23-2[1]		

- Виведіть отриманий список користувачів на веб-сторінці.

#### 4. Отримання списку користувачів (Backend):

- Створіть PHP-скрипт, який вибирає всіх користувачів з бази даних.
- Поверніть список користувачів у форматі JSON, щоб його можна було легко обробити на клієнті.
- Додайте можливість входу користувача за допомогою електронної адреси та пароля.
- Реалізуйте можливість видалення користувача.
- Додайте можливість редагування даних користувача.

*База даних:*

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	<b>id</b> ↗	int			Ні	Немає		AUTO_INCREMENT	<span>Змінити</span> <span>Знищити</span> <span>Більше</span>
2	<b>name</b>	varchar(100)	utf8mb4_unicode_ci		Ні	Немає			<span>Змінити</span> <span>Знищити</span> <span>Більше</span>
3	<b>email</b> ↗	varchar(100)	utf8mb4_unicode_ci		Ні	Немає			<span>Змінити</span> <span>Знищити</span> <span>Більше</span>
4	<b>password</b>	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			<span>Змінити</span> <span>Знищити</span> <span>Більше</span>
5	<b>created_at</b>	timestamp			Ні	CURRENT_TIMESTAMP		DEFAULT_GENERATED	<span>Змінити</span> <span>Знищити</span> <span>Більше</span>

Рис.1.1 Створена БД

*Лістинг коду:*

*config.php*

```
<?php
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
Authorization, X-Requested-With");

error_reporting(E_ALL);
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

ini_set('display_errors', 0);

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "user_management";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $conn->exec("SET NAMES utf8");
} catch(PDOException $e) {
    die(json_encode(["error" => "Database connection failed"]));
}

```

### *register.php*

```

<?php
global $conn;
require_once 'config.php';

$data = json_decode(file_get_contents("php://input"));

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    echo json_encode(["error" => "Method not allowed"]);
    exit();
}

if (empty($data->name) || empty($data->email) || empty($data->password)) {
    echo json_encode(["error" => "All fields are required"]);
    exit();
}

if (strlen($data->password) < 6) {
    echo json_encode(["error" => "Password must be at least 6 characters long"]);
    exit();
}

$stmt = $conn->prepare("SELECT id FROM users WHERE email = ?");
$stmt->execute([$data->email]);
if ($stmt->rowCount() > 0) {
    echo json_encode(["error" => "Email already exists"]);
    exit();
}

$hashedPassword = password_hash($data->password, PASSWORD_DEFAULT);
$stmt = $conn->prepare("INSERT INTO users (name, email, password) VALUES (?, ?, ?)");
$stmt->execute([$data->name, $data->email, $hashedPassword]);

echo json_encode(["success" => "User registered successfully"]);

```

### *login.php*

```

<?php
global $conn;
require_once 'config.php';

$data = json_decode(file_get_contents("php://input"));

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    echo json_encode(["error" => "Method not allowed"]);
    exit();
}

```

Змн.	Арк.	Дзінзілевич Д.О.		
		Ковтун В.В.		

```

if (empty($data->email) || empty($data->password)) {
    echo json_encode(["error" => "Email and password are required"]);
    exit();
}

$stmt = $conn->prepare("SELECT id, name, email, password FROM users WHERE email = ?");
$stmt->execute([$data->email]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$user || !password_verify($data->password, $user['password'])) {
    echo json_encode(["error" => "Invalid email or password"]);
    exit();
}

session_start();
$_SESSION['user_id'] = $user['id'];
$_SESSION['user_email'] = $user['email'];
$_SESSION['user_name'] = $user['name'];

echo json_encode([
    "success" => "Login successful",
    "user" => [
        "id" => $user['id'],
        "name" => $user['name'],
        "email" => $user['email']
    ]
]);

```

### *auth.php*

```

<?php
global $conn;
require_once 'config.php';

session_start();

if (!isset($_SESSION['user_id'])) {
    echo json_encode(["error" => "Not authenticated"]);
    exit();
}

$stmt = $conn->prepare("SELECT id, name, email FROM users WHERE id = ?");
$stmt->execute([$_SESSION['user_id']]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$user) {
    echo json_encode(["error" => "User not found"]);
    exit();
}

echo json_encode([
    "authenticated" => true,
    "user" => $user
]);

```

### *users.php*

```

<?php
global $conn;
require_once 'config.php';

header("Content-Type: application/json; charset=UTF-8");

```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25.121.08.000 – Пр6

Арк.

4

```

try {
    session_start();
    if (!isset($_SESSION['user_id'])) {
        throw new Exception("Not authorized");
    }

    switch ($_SERVER['REQUEST_METHOD']) {
        case 'GET':
            $stmt = $conn->query("SELECT id, name, email, created_at FROM users");
            $users = $stmt->fetchAll(PDO::FETCH_ASSOC);
            echo json_encode($users, JSON_UNESCAPED_UNICODE);
            break;

        case 'DELETE':
            $input = json_decode(file_get_contents("php://input"), true);
            if (empty($input['id'])) {
                throw new Exception("User ID is required");
            }

            $stmt = $conn->prepare("DELETE FROM users WHERE id = ?");
            $stmt->execute([$input['id']]);
            echo json_encode(["success" => "User deleted successfully"]);
            break;

        case 'PUT':
            $input = json_decode(file_get_contents("php://input"), true);
            if (empty($input['id']) || empty($input['name']) || empty($input['email'])) {
                throw new Exception("All fields are required");
            }

            $stmt = $conn->prepare("SELECT id FROM users WHERE email = ? AND id != ?");
            $stmt->execute([$input['email'], $input['id']]);
            if ($stmt->rowCount() > 0) {
                throw new Exception("Email already exists");
            }

            $stmt = $conn->prepare("UPDATE users SET name = ?, email = ? WHERE id = ?");
            $stmt->execute([$input['name'], $input['email'], $input['id']]);
            echo json_encode(["success" => "User updated successfully"]);
            break;

        default:
            throw new Exception("Method not allowed");
    }
} catch (Exception $e) {
    http_response_code(400);
    echo json_encode(["error" => $e->getMessage()]);
}

```

### ***logout.php***

```

<?php
require_once 'config.php';

session_start();
session_unset();
session_destroy();

echo json_encode(["success" => "Logged out successfully"]);

```

Змн.	Арк.	Дзінзілевич Д.О.		
		Ковтун В.В.		

## *index.html*

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RegisterForm</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
<div class="container">
    <h1>Registration</h1>
    <form id="registerForm">
        <div class="form-group">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>
        </div>
        <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
        </div>
        <div class="form-group">
            <label for="password">Password (Min Length > 6):</label>
            <input type="password" id="password" name="password" required
minlength="6">
        </div>
        <button type="submit">SignUp</button>
    </form>
    <p>Already have an account? <a href="login.html">LogIn</a></p>
    <div id="message"></div>
</div>
<script src="js/register.js"></script>
</body>
</html>
```

## *login.html*

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>LoginForm</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
<div class="container">
    <h1>Login</h1>
    <form id="loginForm">
        <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
        </div>
        <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
        </div>
        <button type="submit">SignIn</button>
    </form>
    <p>Not registered yet? <a href="index.html">Registration</a></p>
    <div id="message"></div>
</div>
<script src="js/auth.js"></script>
```

Змн.	Арк.	Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				6

```
</body>
</html>
```

## *users.html*

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Users</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
<div class="container">
    <h1>UsersList</h1>
    <div id="userControls">
        <button id="logoutBtn">Log Out</button>
    </div>
    <div id="usersTable">
        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Email</th>
                    <th>Registration Date</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody id="usersList">
                <!-- Users will be loaded here -->
            </tbody>
        </table>
    </div>
    <div id="message"></div>
</div>
<!-- Edit User Modal -->
<div id="editModal" class="modal">
    <div class="modal-content">
        <span class="close">&times;</span>
        <h2>Edit User</h2>
        <form id="editUserForm">
            <input type="hidden" id="editUserId">
            <div class="form-group">
                <label for="editName">Name:</label>
                <input type="text" id="editName" required>
            </div>
            <div class="form-group">
                <label for="editEmail">Email:</label>
                <input type="email" id="editEmail" required>
            </div>
            <button type="submit">Save Changes</button>
        </form>
    </div>
</div>

<script src="js/users.js"></script>
</body>
</html>
```

## *register.js*

```
document.getElementById('registerForm').addEventListener('submit', async function(e) {
```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25.121.08.000 – Пр6

Арк.

```

e.preventDefault();

const name = document.getElementById('name').value;
const email = document.getElementById('email').value;
const password = document.getElementById('password').value;

if (!name || !email || !password) {
    showMessage('error', 'Будь ласка, заповніть всі поля');
    return;
}

if (password.length < 6) {
    showMessage('error', 'Пароль повинен містити щонайменше 6 символів');
    return;
}

try {
    await registerUser({ name, email, password });
} catch (error) {
    console.error('Error:', error);
    showMessage('error', 'Сталася помилка при реєстрації');
}
);

async function registerUser(userData) {
    const response = await fetch('api/register.php', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(userData)
    });

    const data = await response.json();
    console.log(data);

    if (response.ok) {
        showMessage('success', 'Реєстрація успішна!');
    } else {
        showMessage('error', data.message || 'Помилка реєстрації');
    }
}

return data;
}

function showMessage(type, text) {
    const messageDiv = document.getElementById('message');
    messageDiv.textContent = text;
    messageDiv.className = type;

    setTimeout(() => {
        messageDiv.textContent = '';
        messageDiv.className = '';
    }, 5000);
}

```

### ***auth.js***

```

document.getElementById('loginForm').addEventListener('submit', async function(e)
{
    e.preventDefault();

    const email = document.getElementById('email').value;
    const password = document.getElementById('password').value;

    if (!email || !password) {

```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25.121.08.000 – Пр6

Арк.

```

        showMessage('error', 'Будь ласка, заповніть всі поля');
        return;
    }

    try {
        await loginUser({ email, password });
    } catch (error) {
        showMessage('error', 'Помилка під час входу: ' + error.message);
    }
});

async function loginUser(credentials) {
    const response = await fetch('api/login.php', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify(credentials)
    });

    const data = await response.json();

    if (!response.ok) {
        throw new Error(data.error || 'Невідома помилка сервера');
    }

    showMessage('success', 'Вхід успішний! Перенаправлення...');
    setTimeout(() => {
        window.location.href = 'users.html';
    }, 1500);
}

function showMessage(type, text) {
    const messageDiv = document.getElementById('message');
    messageDiv.textContent = text;
    messageDiv.className = type;

    setTimeout(() => {
        messageDiv.textContent = '';
        messageDiv.className = '';
    }, 5000);
}

```

### *users.js*

```

document.addEventListener('DOMContentLoaded', async function() {
    try {
        await checkAuth();
        document.getElementById('logoutBtn').addEventListener('click', logout);
    } catch (error) {
        console.error('Помилка ініціалізації:', error);
        window.location.href = 'login.html';
    }
});

async function checkAuth() {
    try {
        const response = await fetch('api/auth.php');
        const data = await response.json();

        if (data.error || !data.authenticated) {
            window.location.href = 'login.html';
        } else {
            await loadUsers();
        }
    } catch (error) {
        console.error('Помилка перевірки авторизації:', error);
        window.location.href = 'login.html';
    }
}

```

Змн.	Арк.	Дзінзілевич Д.О.		
		Ковтун В.В.		

```

        }
    } catch (error) {
        console.error('Помилка перевірки авторизації:', error);
        window.location.href = 'login.html';
    }
}

async function loadUsers() {
    try {
        const response = await fetch('api/users.php');
        if (!response.ok) throw new Error('Помилка завантаження');

        const users = await response.json();
        displayUsers(users);
    } catch (error) {
        console.error('Помилка завантаження:', error);
        showMessage('error', 'Не вдалося завантажити користувачів');
    }
}

function displayUsers(users) {
    const usersList = document.getElementById('usersList');
    usersList.innerHTML = '';

    users.forEach(user => {
        const row = document.createElement('tr');
        row.innerHTML =
            ` ${user.id} | ${user.name} | ${user.email} | ${new Date(user.created_at).toLocaleString()} |
            <td>
                <button class="action-btn edit-btn" data-
id="${user.id}">Edit</button>
                <br><br>
                <button class="action-btn delete-btn" data-
id="${user.id}">Delete</button>
            </td>
        `;
        usersList.appendChild(row);
    });
}

document.querySelectorAll('.edit-btn').forEach(btn => {
    btn.addEventListener('click', () => {
        const userId = btn.getAttribute('data-id');
        openEditModal(userId);
    });
});

document.querySelectorAll('.delete-btn').forEach(btn => {
    btn.addEventListener('click', async () => {
        const userId = btn.getAttribute('data-id');
        await deleteUser(userId);
    });
});

async function openEditModal(userId) {
    try {
        const modal = document.getElementById('editModal');
        const span = document.getElementsByClassName('close')[0];
        const response = await fetch(`api/users.php?id=${userId}`);

```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

        if (!response.ok) throw new Error('Помилка завантаження даних');

        const userData = await response.json();

        document.getElementById('editUserId').value = userData.id;
        document.getElementById('editName').value = userData.name;
        document.getElementById('editEmail').value = userData.email;

        modal.style.display = 'block';

        span.onclick = () => modal.style.display = 'none';
        window.onclick = (event) => {
            if (event.target === modal) modal.style.display = 'none';
        };

        document.getElementById('editUserForm').onsubmit = async (e) => {
            e.preventDefault();
            const updatedUser = {
                id: document.getElementById('editUserId').value,
                name: document.getElementById('editName').value,
                email: document.getElementById('editEmail').value
            };
            await updateUser(updatedUser);
        };
    } catch (error) {
        console.error('Помилка відкриття модального вікна:', error);
        showMessage('error', 'Не вдалося завантажити дані користувача');
    }
}

async function updateUser(userData) {
    try {
        const response = await fetch('api/users.php', {
            method: 'PUT',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify(userData)
        });

        if (!response.ok) {
            const errorData = await response.json();
            throw new Error(errorData.error || 'Помилка оновлення');
        }

        document.getElementById('editModal').style.display = 'none';
        showMessage('success', 'Дані користувача оновлено');
        await loadUsers();
    } catch (error) {
        console.error('Помилка оновлення:', error);
        showMessage('error', error.message);
    }
}

async function deleteUser(userId) {
    if (!confirm('Ви впевнені, що хочете видалити цього користувача?')) return;

    try {
        const response = await fetch('api/users.php', {
            method: 'DELETE',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ id: userId })
        });

        if (!response.ok) {

```

Змн.	Арк.	Дзінзілевич Д.О.		
		Ковтун В.В.		

```

        const errorData = await response.json();
        throw new Error(errorData.error || 'Помилка видалення');
    }

    showMessage('success', 'Користувача успішно видалено');
    await loadUsers();
} catch (error) {
    console.error('Помилка видалення:', error);
    showMessage('error', error.message);
}
}

async function logout() {
    try {
        await fetch('api/logout.php', { method: 'POST' });
        window.location.href = 'login.html';
    } catch (error) {
        console.error('Помилка виходу:', error);
        showMessage('error', 'Помилка під час виходу з системи');
    }
}

function showMessage(type, text) {
    const messageDiv = document.getElementById('message');
    messageDiv.textContent = text;
    messageDiv.className = type;

    setTimeout(() => {
        messageDiv.textContent = '';
        messageDiv.className = '';
    }, 5000);
}

```

*Результат виконання програми:*

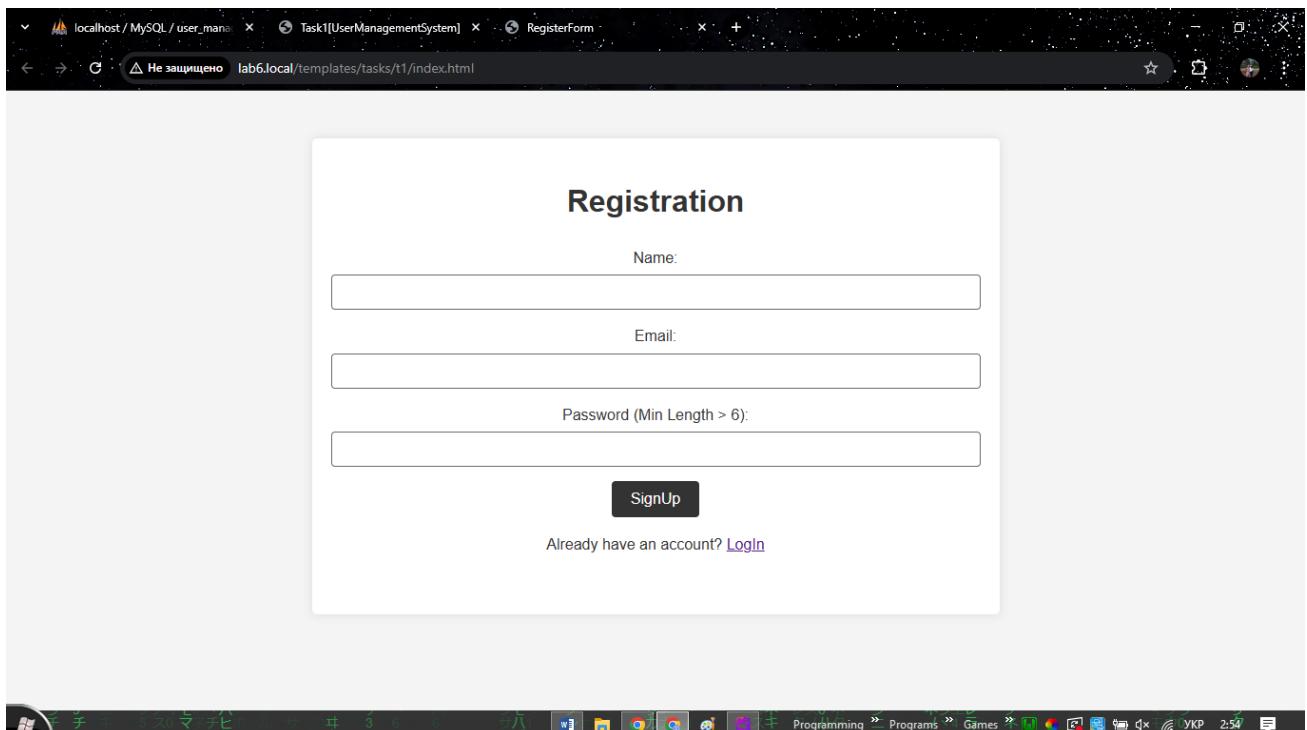


Рис.1.2 Сторінка реєстрації

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

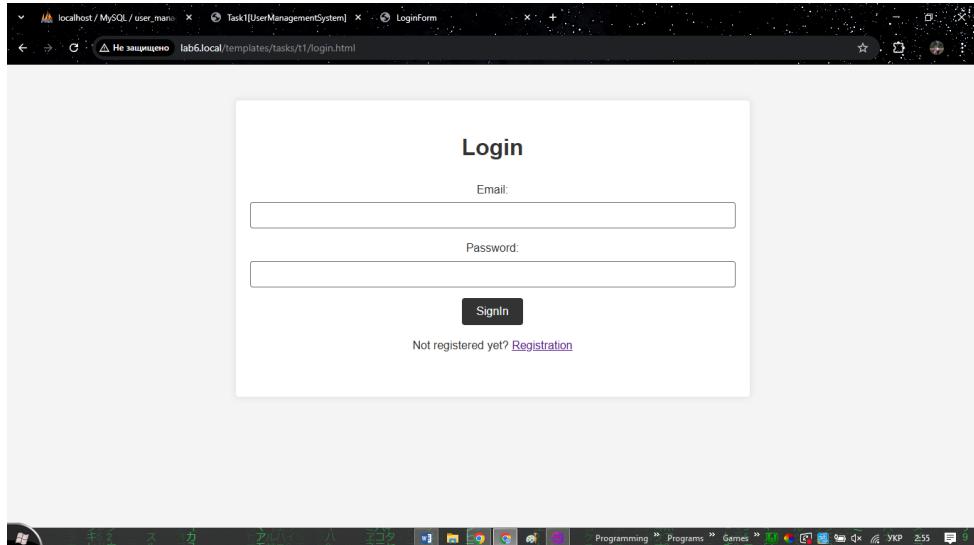


Рис.1.3 Сторінка входу в систему

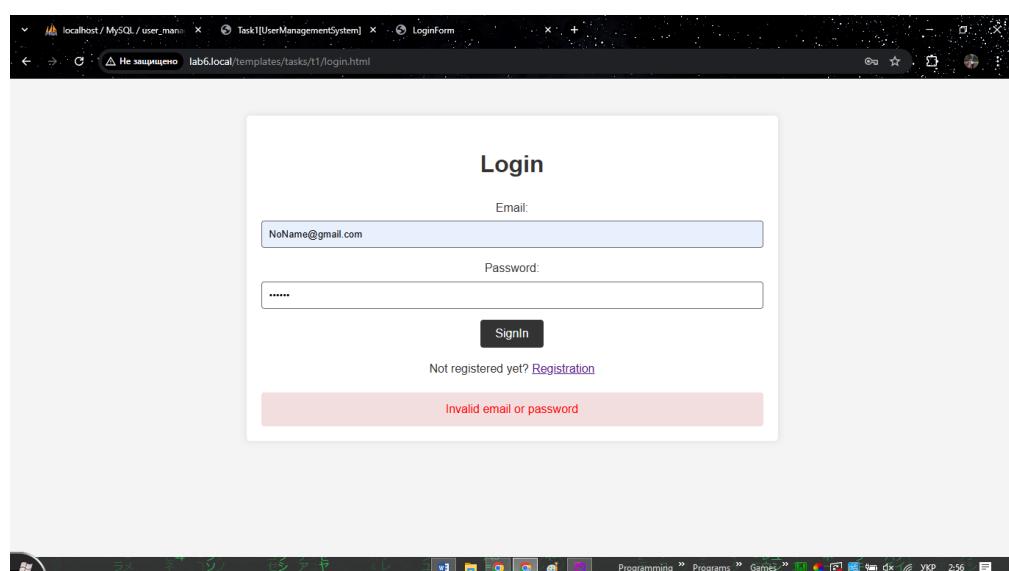


Рис.1.4 Спроба входу незареєстрованого користувача

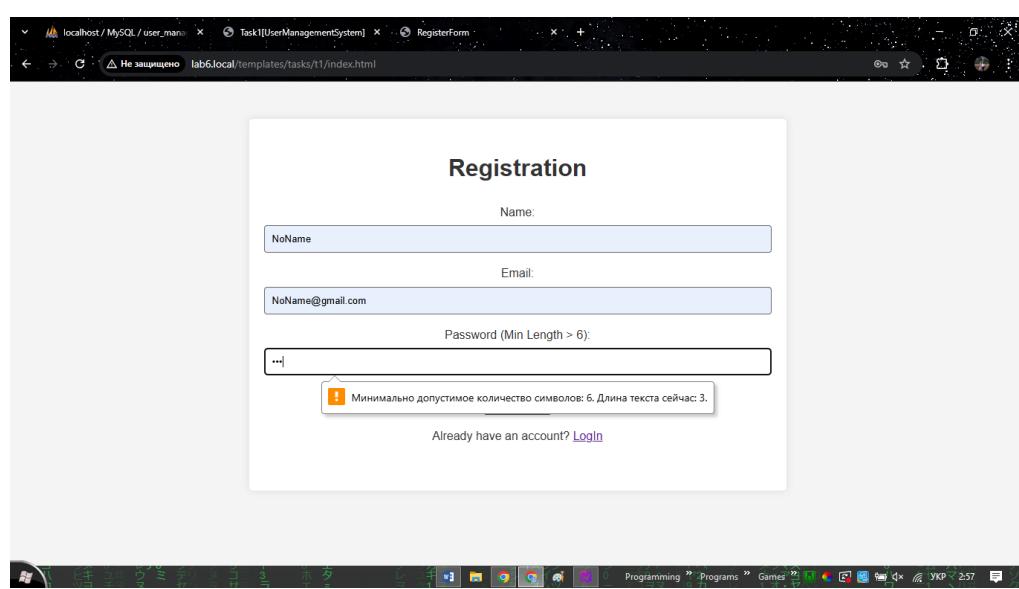


Рис.1.5 Перевірка довжини паролю

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Рис.1.6 Спроба створення акаунту з уже використовуваною поштою

	id	name	email	password	created_at
1	1	Test	test@gmail.com	\$2y\$12\$xtu3oF6QICZaf7PyFKPReefyqUfLzYs88GqhWGKY...	2025-03-26 00:20:36
2	2	NoName	NoName@gmail.com	\$2y\$12\$PRNTlBs0dbKUKVxpO6WelqdtkT5cwe4rY7vKewR...	2025-03-26 02:57:48

Рис.1.7 Перевірка чи добався користувач в БД

ID	Name	Email	Registration Date	Actions
1	Test	test@gmail.com	26.03.2025, 00:20:36	<a href="#">Edit</a> <a href="#">Delete</a>
2	NoName	NoName@gmail.com	26.03.2025, 02:57:48	<a href="#">Edit</a> <a href="#">Delete</a>

Рис.1.8 Таблиця користувачів

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

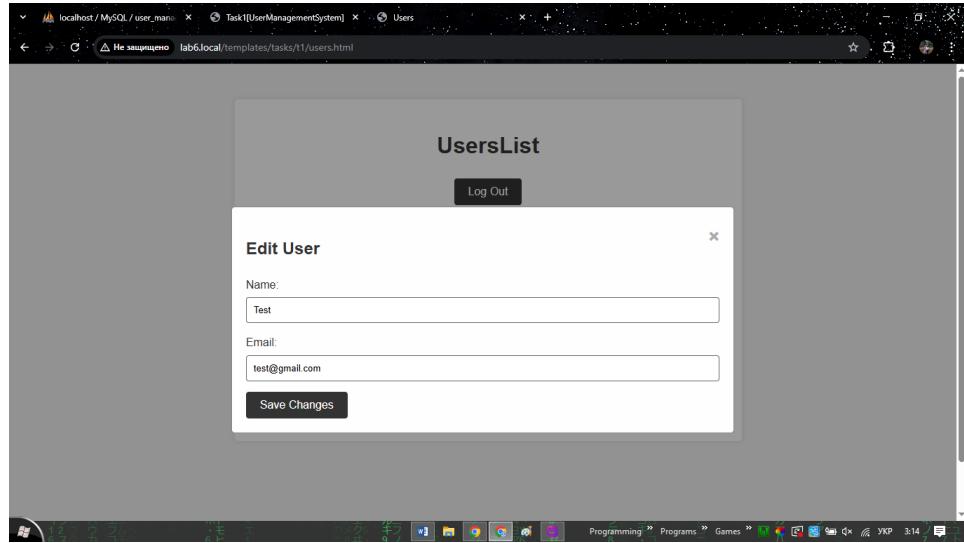


Рис.1.9 Редагування користувача

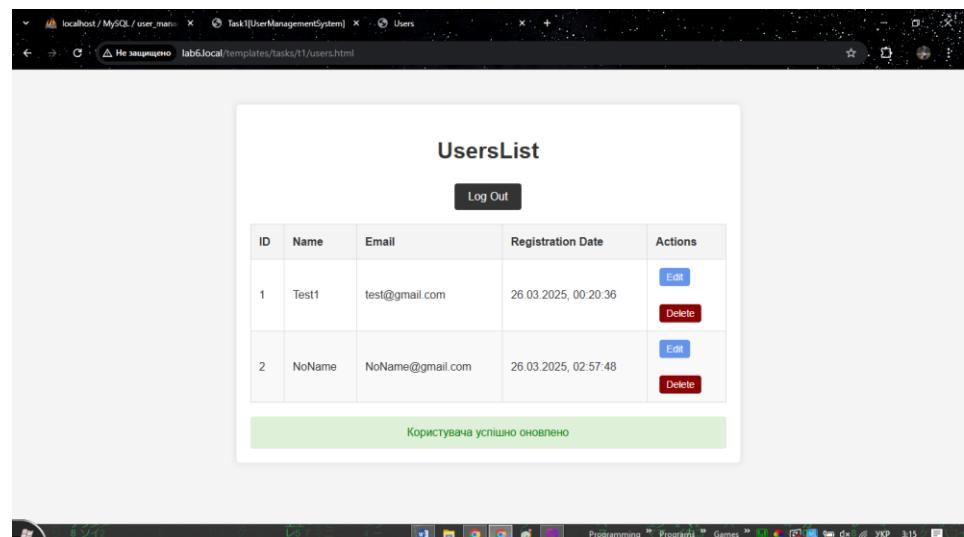


Рис.1.10 Оновлені дані в таблиці

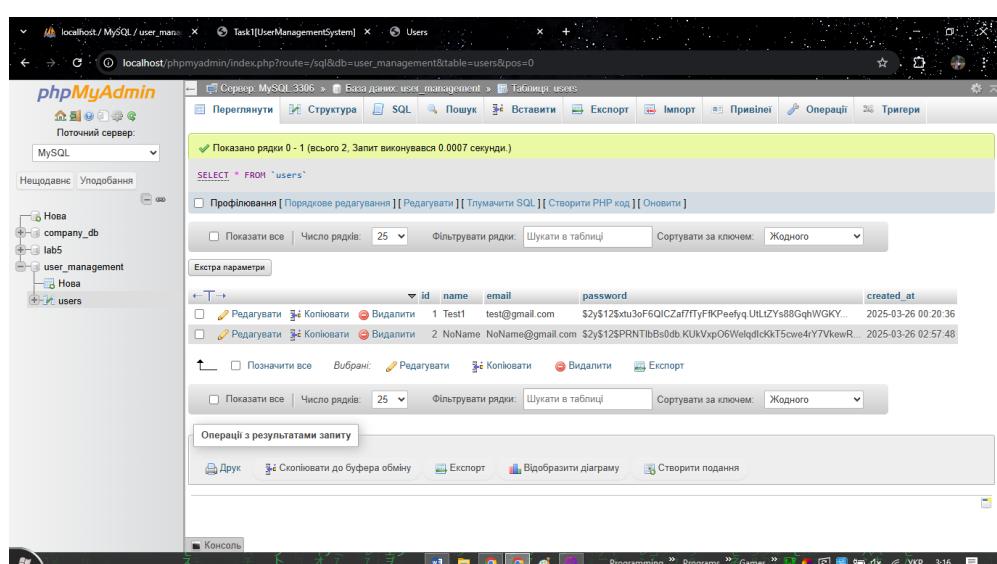


Рис.1.11 Оновлені дані в БД

		Дзінзілевич Д.О.			Арк.
		Ковтун В.В.			
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.08.000 – Пр6 15

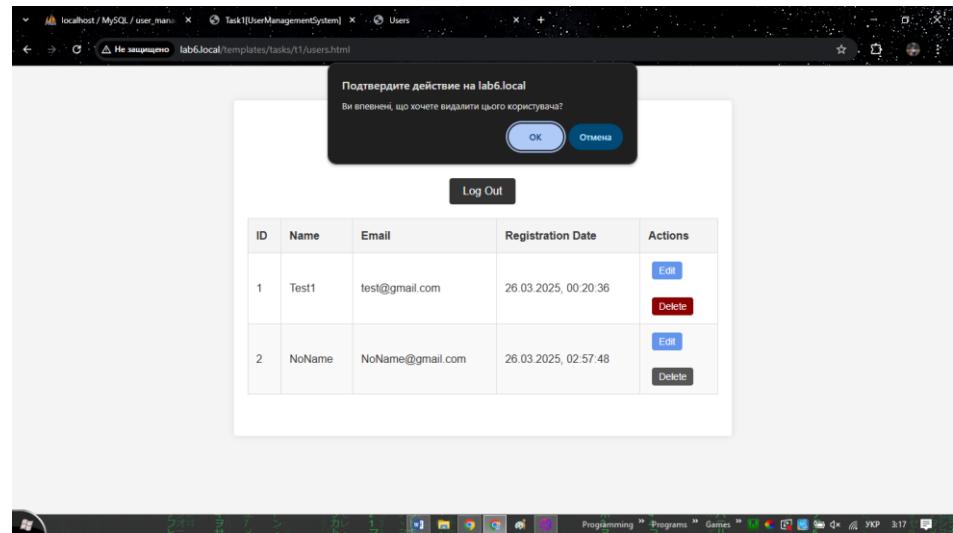


Рис.1.12 Спроба видалення користувача

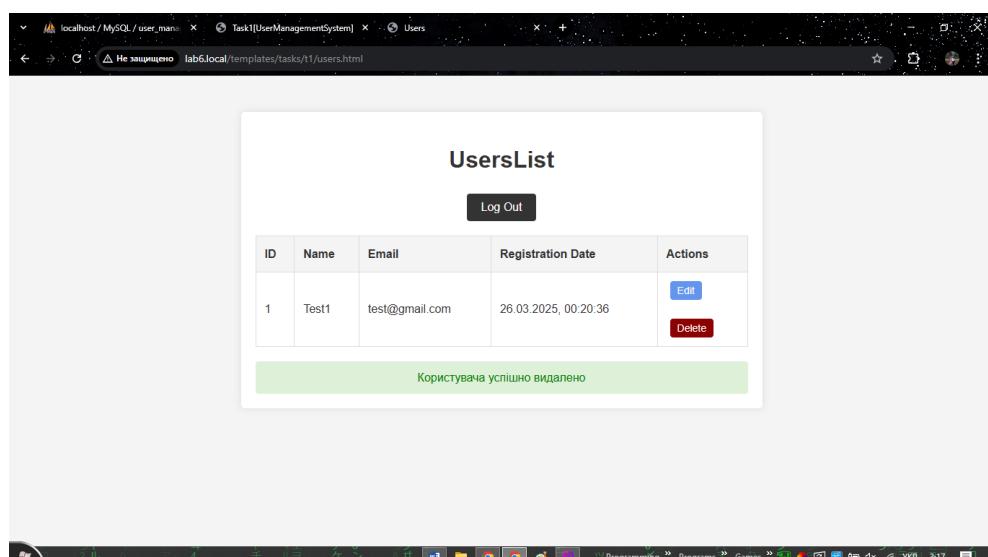


Рис.1.13 Видалений користувач з таблиці

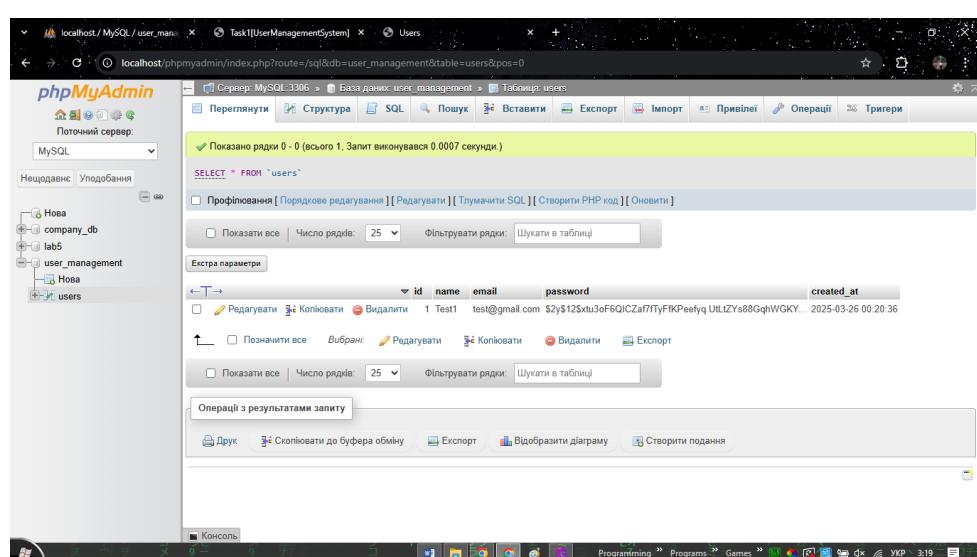


Рис.1.14 Видалений користувач з БД

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

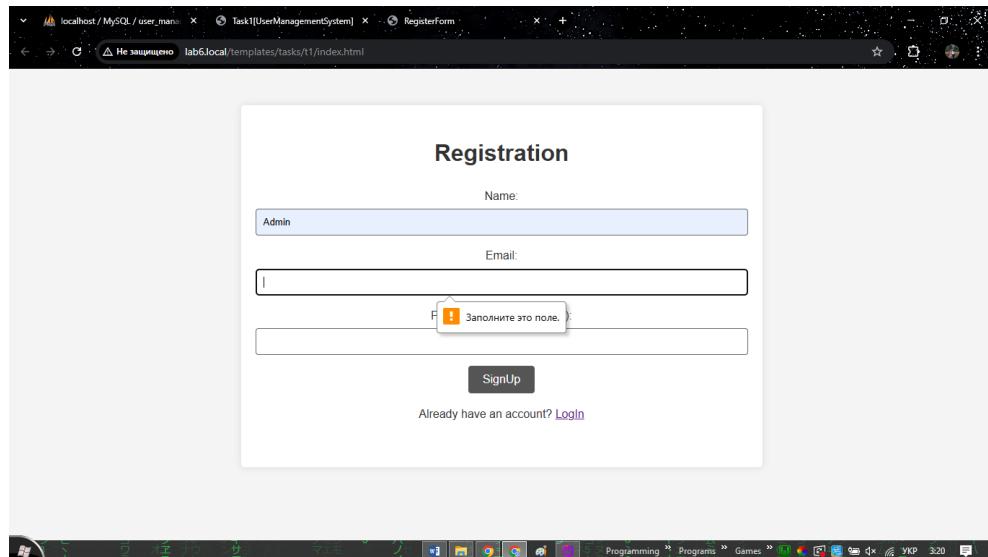


Рис.1.15 Перевірка чи заповнені всі поля

Завдання 2: Реалізація заміток (Notes App) за допомогою FETCH та PHP.

Створіть веб-додаток, який дозволяє користувачам додавати, переглядати, оновлювати та видаляти замітки. Користувачі можуть створювати замітки, вказуючи заголовок та текст. Всі замітки повинні зберігатися на сервері. Взаємодія між клієнтом та сервером повинна здійснюватися за допомогою асинхронних запитів та PHP на серверній стороні.

Ключові вимоги:

- Додавання заміток:** Користувач повинен мати можливість додавати нові замітки, вказуючи заголовок та текст, та надсилати їх на сервер за допомогою асинхронних запитів.
- Перегляд заміток:** Усі замітки, збережені на сервері, повинні відображатися на головній сторінці.
- Оновлення заміток:** Користувач повинен мати можливість оновлювати замітки, змінюючи їх заголовок або текст. Зміни повинні відображатися на сервері після натискання відповідної кнопки або події.
- Видалення заміток:** Користувач повинен мати можливість видаляти замітки за допомогою відповідної опції або кнопки. Видалення заміток повинно бути асинхронним, без перезавантаження сторінки.

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

- Валідація даних на клієнті та сервері:** Перед надсиланням даних на сервер, вони повинні бути перевірені на клієнті (заголовок та текст мають бути заповнені).
- Відправка та отримання даних за допомогою fetch API:** Взаємодія з сервером повинна здійснюватися за допомогою fetch API на клієнтській стороні для відправки та отримання даних.
- Зберігання даних на сервері з використанням PHP:** Замітки повинні зберігатися на сервері за допомогою PHP. Використовуйте MySQL або іншу базу даних для зберігання інформації про замітки.
- Асинхронність та безперервність додатку:** Усі операції (додавання, оновлення, видалення заміток) повинні відбуватися без перезавантаження сторінки, з використанням асинхронних запитів.

#### База даних:

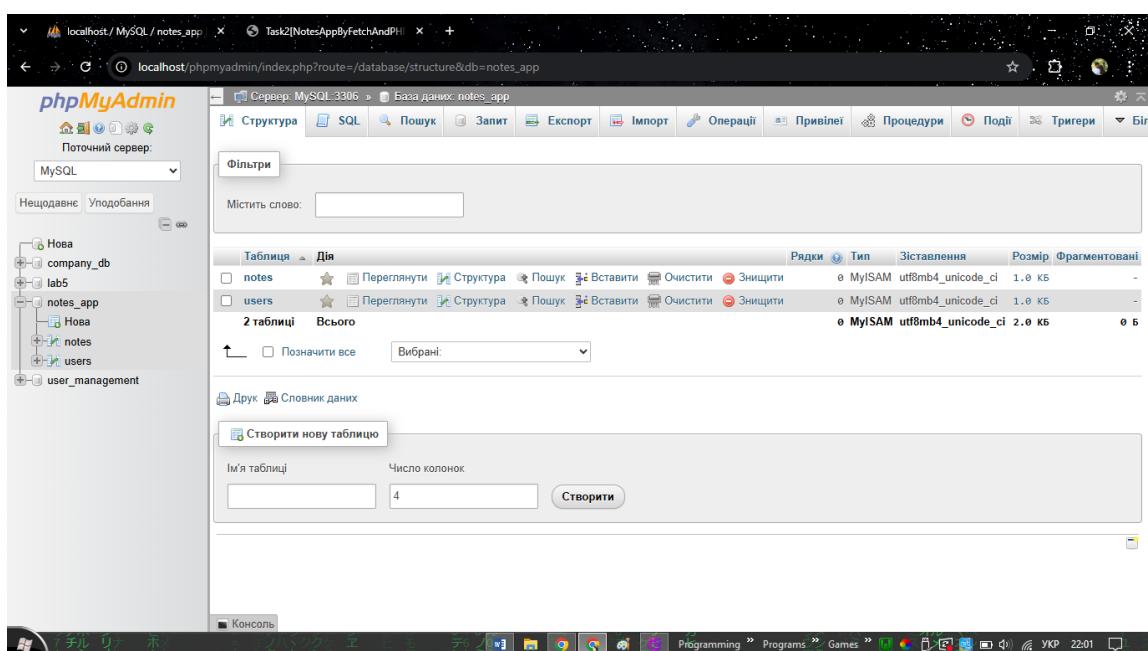


Рис.2.1 Створена БД

Лістинг коду:

#### auth.php

```
<?php
require_once 'config.php';

session_start();

function isLoggedIn() {
    return isset($_SESSION['user_id']);
```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

}

function getCurrentUserId() {
    return $_SESSION['user_id'] ?? null;
}

function login($username, $password) {
    global $conn;
    $stmt = $conn->prepare("SELECT id, password FROM users WHERE username = ?");
    $stmt->execute([$username]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($user && password_verify($password, $user['password'])) {
        $_SESSION['user_id'] = $user['id'];
        return true;
    }
    return false;
}

function register($username, $password) {
    global $conn;
    $hashed = password_hash($password, PASSWORD_DEFAULT);
    $stmt = $conn->prepare("INSERT INTO users (username, password) VALUES (?, ?)");
    return $stmt->execute([$username, $hashed]);
}

```

### *check\_auth.php*

```

<?php
global $conn;
require_once 'auth.php';

error_log("Current session: " . print_r($_SESSION, true));

if (isLoggedIn()) {
    $stmt = $conn->prepare("SELECT username FROM users WHERE id = ?");
    $stmt->execute([getCurrentUserId()]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    echo json_encode([
        'authenticated' => true,
        'username' => $user['username'],
        'user_id' => getCurrentUserId()
    ]);
} else {
    echo json_encode(['authenticated' => false]);
}

```

### *check\_note.php*

```

<?php
global $conn;
require_once 'config.php';
require_once 'auth.php';

header("Content-Type: application/json");

if (!isLoggedIn()) {
    die(json_encode(["isOwner" => false]));
}

$data = json_decode(file_get_contents("php://input"));

$stmt = $conn->prepare("SELECT id FROM notes WHERE id = ? AND user_id = ?");

```

Змн.	Арк.	Дзінзілевич Д.О.		
		Ковтун В.В.		

```

$stmt->execute([$data->id, getCurrentUserId()]);

echo json_encode(["isOwner" => $stmt->rowCount() > 0]);

```

***config.php***

```

<?php
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
Authorization, X-Requested-With");

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "notes_app";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    echo json_encode(["error" => "Connection failed: " . $e->getMessage()]);
    exit();
}

```

***create.php***

```

<?php
global $conn;
require_once 'config.php';
require_once 'auth.php';

$data = json_decode(file_get_contents("php://input"));
$user_id = getCurrentUserId();

$query = "INSERT INTO notes (title, content, user_id) VALUES (?, ?, ?)";
$stmt = $conn->prepare($query);
$stmt->execute([$data->title, $data->content, $user_id]);

http_response_code(201);
echo json_encode(["message" => "Note created successfully."]);

if (!isLoggedIn()) {
    http_response_code(401);
    die(json_encode(["error" => "Для додавання нотаток увійдіть у систему"]));
}

if (empty($data->title) || empty($data->content)) {
    http_response_code(400);
    die(json_encode(["error" => "Заголовок і текст нотатки обов'язкові"]));
}

```

***delete.php***

```

<?php
global $conn;
require_once 'config.php';
require_once 'auth.php';

header("Content-Type: application/json");

if (!isLoggedIn()) {
    http_response_code(401);
    die(json_encode(["error" => "Необхідно авторизуватися"]));
}

```

Змн.	Арк.	Дзінзілевич Д.О.		
		Ковтун В.В.		

```

}

$data = json_decode(file_get_contents("php://input"));

if (empty($data->id)) {
    http_response_code(400);
    die(json_encode(["error" => "Не вказано ID нотатки"]));
}

$user_id = getCurrentUserId();

$checkQuery = "SELECT id FROM notes WHERE id = ? AND user_id = ?";
$checkStmt = $conn->prepare($checkQuery);
$checkStmt->execute([$data->id, $user_id]);

if ($checkStmt->rowCount() === 0) {
    http_response_code(403);
    die(json_encode(["error" => "Ця нотатка вам не належить"]));
}

$deleteQuery = "DELETE FROM notes WHERE id = ?";
$deleteStmt = $conn->prepare($deleteQuery);

if ($deleteStmt->execute([$data->id])) {
    echo json_encode(["success" => true, "message" => "Нотатку видалено"]);
} else {
    http_response_code(500);
    echo json_encode(["error" => "Помилка при видаленні"]);
}

```

### ***login.php***

```

<?php
global $conn;
require_once 'auth.php';

$data = json_decode(file_get_contents("php://input"));

error_log("Login attempt: " . print_r($data, true));

if (empty($data->username) || empty($data->password)) {
    http_response_code(400);
    die(json_encode(["error" => "Будь ласка, заповніть всі поля"]));
}

error_log("Querying user: " . $data->username);
$stmt = $conn->prepare("SELECT id, password FROM users WHERE username = ?");
$stmt->execute([$data->username]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$user) {
    error_log("User not found: " . $data->username);
    http_response_code(401);
    die(json_encode(["error" => "Користувача не знайдено"]));
}

error_log("Found user: " . print_r($user, true));

if (!password_verify($data->password, $user['password'])) {
    error_log("Password verification failed for user: " . $data->username);
    http_response_code(401);
    die(json_encode(["error" => "Невірний пароль"]));
}

```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
$_SESSION['user_id'] = $user['id'];
error_log("Login successful for user ID: " . $user['id']);

echo json_encode(["success" => true]);
```

***logout.php***

```
<?php
session_start();

session_unset();
session_destroy();

echo json_encode(["success" => true]);
```

***read.php***

```
<?php
global $conn;
require_once 'config.php';
require_once 'auth.php';

$isLoggedIn = isLoggedIn();
$userId = $isLoggedIn ? getCurrentUserId() : null;

$query = "SELECT id, title, content, created_at, updated_at,
           CASE WHEN user_id = ? THEN 1 ELSE 0 END as is_owner
      FROM notes
     ORDER BY updated_at DESC";

$stmt = $conn->prepare($query);
$stmt->execute([$userId]);

echo json_encode($stmt->fetchAll(PDO::FETCH_ASSOC));
```

***register.php***

```
<?php
require_once 'auth.php';

$data = json_decode(file_get_contents("php://input"));

if (empty($data->username) || empty($data->password)) {
    http_response_code(400);
    die(json_encode(["error" => "Будь ласка, заповніть всі поля"]));
}

if (strlen($data->password) < 6) {
    http_response_code(400);
    die(json_encode(["error" => "Пароль повинен містити щонайменше 6 символів"]));
}

try {
    if (register($data->username, $data->password)) {
        echo json_encode(["success" => true]);
    } else {
        http_response_code(400);
        echo json_encode(["error" => "Registration failed"]);
    }
} catch (PDOException $e) {
    http_response_code(400);
    echo json_encode(["error" => "Username already exists"]);
}
```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

## *update.php*

```

<?php
global $conn;
require_once 'config.php';
require_once 'auth.php';

header("Content-Type: application/json");

if (!isLoggedIn()) {
    http_response_code(401);
    die(json_encode(["error" => "Необхідно авторизуватися"]));
}

$data = json_decode(file_get_contents("php://input"));

if (empty($data->id) || empty($data->title) || empty($data->content)) {
    http_response_code(400);
    die(json_encode(["error" => "Недостатньо даних"]));
}

$user_id = getCurrentUserId();

$checkQuery = "SELECT id FROM notes WHERE id = ? AND user_id = ?";
$checkStmt = $conn->prepare($checkQuery);
$checkStmt->execute([$data->id, $user_id]);

if ($checkStmt->rowCount() === 0) {
    http_response_code(403);
    die(json_encode(["error" => "Ця нотатка вам не належить"]));
}

$updateQuery = "UPDATE notes SET title = ?, content = ?, updated_at = CURRENT_TIMESTAMP WHERE id = ?";
$updateStmt = $conn->prepare($updateQuery);

if ($updateStmt->execute([$data->title, $data->content, $data->id])) {
    echo json_encode(["success" => true, "message" => "Нотатку оновлено"]);
} else {
    http_response_code(500);
    echo json_encode(["error" => "Помилка при оновленні"]);
}

```

## *index.html*

```

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Notes App</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

<div class="auth-buttons">
    <button id="loginBtn">Увійти</button>
    <button id="registerBtn">Реєстрація</button>
    <span id="userGreeting"></span>
    <button id="logoutBtn" style="display:none;">Вийти</button>
</div>

<div id="authModal" class="modal">
    <div class="modal-content">

```

		Дзінзілевич Д.О.			Арк.
		Ковтун В.В.			
Змн.	Арк.	№ докум.	Підпис	Дата	23

```

<span class="close">&times;</span>
<h2 id="modalTitle">Увійти</h2>
<form id="authForm">
    <input type="text" id="username" placeholder="Логін" required>
    <input type="password" id="password" placeholder="Пароль" required>
    <button type="submit">Підтвердити</button>
</form>
</div>
</div>

<div id="editModal" class="modal">
    <div class="modal-content">
        <span class="close-edit">&times;</span>
        <h2>Редагувати нотатку</h2>
        <form id="editNoteForm">
            <div class="form-group">
                <label for="editTitle">Заголовок:</label>
                <input type="text" id="editTitle" required>
            </div>
            <div class="form-group">
                <label for="editContent">Текст:</label>
                <textarea id="editContent" rows="4" required></textarea>
            </div>
            <button type="submit">Зберегти зміни</button>
        </form>
    </div>
</div>

<div class="container">
    <h1>My Notes</h1>

    <div class="note-form">
        <h2>Add New Note</h2>
        <form id="noteForm">
            <div class="form-group">
                <label for="title">Title:</label>
                <input type="text" id="title" name="title" required>
            </div>
            <div class="form-group">
                <label for="content">Text:</label>
                <textarea id="content" name="content" rows="4" required></textarea>
            </div>
            <button type="submit">Save</button>
        </form>
    </div>

    <div class="notes-list">
        <h3>Save Notes</h3>
        <div id="notesContainer"></div>
    </div>
</div>

<script src="scripts.js"></script>
</body>
</html>

```

### *scripts.js*

```

document.addEventListener('DOMContentLoaded', function() {
    const authModal = document.getElementById('authModal');
    const editModal = document.getElementById('editModal');
    const authForm = document.getElementById('authForm');
    const noteForm = document.getElementById('noteForm');

```

Дзінзілевич Д.О.		
Ковтун В.В.		
Змн.	Арк.	№ докум.

```

const editNoteForm = document.getElementById('editNoteForm');
const notesContainer = document.getElementById('notesContainer');
const loginBtn = document.getElementById('loginBtn');
const registerBtn = document.getElementById('registerBtn');
const closeAuthBtn = document.querySelector('.close');
const closeEditBtn = document.querySelector('.close-edit');

let isLoginPage = true;
let currentNoteId = null;

function showMessage(type, text) {
    const messageDiv = document.createElement('div');
    messageDiv.className = `message ${type}`;
    messageDiv.textContent = text;
    document.body.appendChild(messageDiv);

    setTimeout(() => messageDiv.remove(), 5000);
}

//Modal Windows
function setupModal(modal, openButtons, closeButton) {
    openButtons.forEach(btn => {
        btn.addEventListener('click', () => {
            modal.style.display = 'block';
        });
    });

    closeButton.addEventListener('click', () => {
        modal.style.display = 'none';
    });

    window.addEventListener('click', (event) => {
        if (event.target === modal) {
            modal.style.display = 'none';
        }
    });
}

setupModal(authModal, [loginBtn, registerBtn], closeAuthBtn);
setupModal(editModal, [], closeEditBtn);

//Authorization
authForm.addEventListener('submit', async (e) => {
    e.preventDefault();
    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;

    try {
        const endpoint = isLoginPage ? 'api/login.php' : 'api/register.php';
        const response = await fetch(endpoint, {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ username, password })
        });

        const data = await response.json();

        if (!response.ok) {
            throw new Error(data.error || 'Помилка авторизації');
        }

        authModal.style.display = 'none';
        updateAuthUI();
    } catch (error) {
        console.error(error);
    }
});

```

		Дзінзілевич Д.О.			Арк.
		Ковтун В.В.			
Змн.	Арк.	№ докум.	Підпис	Дата	25

```

        loadNotes();
        showMessage('success', isLoginPage ? 'Успішний вхід!' : 'Реєстрація
успішна!');
    } catch (error) {
        showMessage('error', error.message);
    }
});

//Note Form
noteForm.addEventListener('submit', async (e) => {
    e.preventDefault();
    const title = document.getElementById('title').value;
    const content = document.getElementById('content').value;

    if (!title || !content) {
        showMessage('error', 'Заповніть всі поля');
        return;
    }

    try {
        const response = await fetch('api/create.php', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ title, content })
        });

        const data = await response.json();

        if (!response.ok) {
            throw new Error(data.error || 'Помилка при додаванні');
        }

        noteForm.reset();
        loadNotes();
        showMessage('success', 'Нотатку додано!');
    } catch (error) {
        showMessage('error', error.message);
    }
});

```

```

//Note Edit Form
editNoteForm.addEventListener('submit', async (e) => {
    e.preventDefault();
    const title = document.getElementById('editTitle').value;
    const content = document.getElementById('editContent').value;

    try {
        const response = await fetch('api/update.php', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({
                id: currentNoteId,
                title: title,
                content: content
            })
        });

        const data = await response.json();

        if (!response.ok) {
            throw new Error(data.error || 'Помилка при оновленні');
        }
    }
});

```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

        editModal.style.display = 'none';
        loadNotes();
        showMessage('success', 'Нотатку оновлено!');
    } catch (error) {
        showMessage('error', error.message);
    }
}

//Work With Note
async function loadNotes() {
    try {
        const response = await fetch('api/read.php');
        const notes = await response.json();
        displayNotes(notes);
    } catch (error) {
        showMessage('error', 'Помилка завантаження нотаток');
    }
}

function displayNotes(notes) {
    notesContainer.innerHTML = notes.length ? '' : '<p>Немає нотаток. Додайте першу!</p>';

    notes.forEach(note => {
        const noteElement = document.createElement('div');
        noteElement.className = 'note';
        noteElement.dataset.id = note.id;
        noteElement.innerHTML =
            `<h3>${escapeHtml(note.title)}</h3>
            <p>${escapeHtml(note.content)}</p>
            <div class="date">Оновлено: ${new Date(note.updated_at).toLocaleString()}</div>
            <div class="actions">
                <button class="edit" data-id="${note.id}">Редагувати</button>
                <button class="delete" data-id="${note.id}">Видалити</button>
            </div>`;
        notesContainer.appendChild(noteElement);
    });
}

//Buttons
document.querySelectorAll('.edit').forEach(btn => {
    btn.addEventListener('click', function() {
        const noteId = this.getAttribute('data-id');
        const noteElement = this.closest('.note');
        const title = noteElement.querySelector('h3').textContent;
        const content = noteElement.querySelector('p').textContent;
        openEditModal(noteId, title, content);
    });
});

document.querySelectorAll('.delete').forEach(btn => {
    btn.addEventListener('click', function() {
        const noteId = this.getAttribute('data-id');
        deleteNote(noteId);
    });
});
}

//Edit Modal Window
async function openEditModal(noteId, title, content) {
    try {
        const response = await fetch('api/check_note.php', {

```

Змн.	Арк.	Дзінзілевич Д.О.		
		Ковтун В.В.		

```

        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ id: noteId })
    });

    const data = await response.json();

    if (!data.isOwner) {
        showMessage('error', 'Ви можете редагувати лише свої нотатки');
        return;
    }

    currentNoteId = noteId;
    document.getElementById('editTitle').value = decodeHtmlEntities(title);
    document.getElementById('editContent').value = decodeHtmlEntities(content);
    document.getElementById('editModal').style.display = 'block';

} catch (error) {
    console.error('Помилка при перевірці прав доступу:', error);
    showMessage('error', 'Не вдалося перевірити права доступу');
}
}

//+ View HTML Content
function decodeHtmlEntities(text) {
    const textArea = document.createElement('textarea');
    textArea.innerHTML = text;
    return textArea.value;
}

//Delete Note
async function checkNoteOwnership(noteId) {
    try {
        const response = await fetch('api/check_note.php', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ id: noteId })
        });
        const data = await response.json();
        return data.isOwner;
    } catch (error) {
        console.error('Помилка перевірки власності:', error);
        return false;
    }
}

window.deleteNote = async (noteId) => {
    const isOwner = await checkNoteOwnership(noteId);
    if (!isOwner) {
        showMessage('error', 'Ви можете видаляти лише свої нотатки');
        return;
    }

    if (!confirm('Видалити цю нотатку?')) return;

    try {
        const response = await fetch('api/delete.php', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ id: noteId })
        });
    }
}

```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

        if (!response.ok) {
            throw new Error('Помилка видалення');
        }

        loadNotes();
        showMessage('success', 'Нотатку видалено!');
    } catch (error) {
        showMessage('error', error.message);
    }
};

//Edit Note
window.openEditModal = async (noteId, currentTitle, currentContent) => {
    const isOwner = await checkNoteOwnership(noteId);
    if (!isOwner) {
        showMessage('error', 'Ви можете редагувати лише свої нотатки');
        return;
    }

    currentNoteId = noteId;
    document.getElementById('editTitle').value = currentTitle;
    document.getElementById('editContent').value = currentContent;
    document.getElementById('editModal').style.display = 'block';
};

//Additionally
function escapeHtml(unsafe) {
    return unsafe
        .replace(/&/g, "&")
        .replace(/</g, "<")
        .replace(/>/g, ">")
        .replace(/\"/g, """)
        .replace(/\'/g, "'");
}

//Authentication
async function updateAuthUI() {
    try {
        const response = await fetch('api/check_auth.php');
        const data = await response.json();

        if (data.authenticated) {
            document.getElementById('userGreeting').textContent = `Вітаю, ${data.username}!`;
            document.getElementById('logoutBtn').style.display = 'inline';
            loginBtn.style.display = 'none';
            registerBtn.style.display = 'none';
            noteForm.style.display = 'block';
        } else {
            document.getElementById('logoutBtn').style.display = 'none';
            loginBtn.style.display = 'inline';
            registerBtn.style.display = 'inline';
            noteForm.style.display = 'none';
        }
    } catch (error) {
        console.error('Помилка перевірки авторизації:', error);
    }
}

//Initialization
updateAuthUI();
loadNotes();

```

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

//Logout
document.getElementById('logoutBtn').addEventListener('click', async () => {
    try {
        const response = await fetch('api/logout.php');
        if (response.ok) {
            await updateAuthUI();
            await loadNotes();
            showMessage('success', 'Ви вийшли з системи');
        }
    } catch (error) {
        showMessage('error', 'Помилка при виході');
    }
});

```

*Результатом виконання програми:*

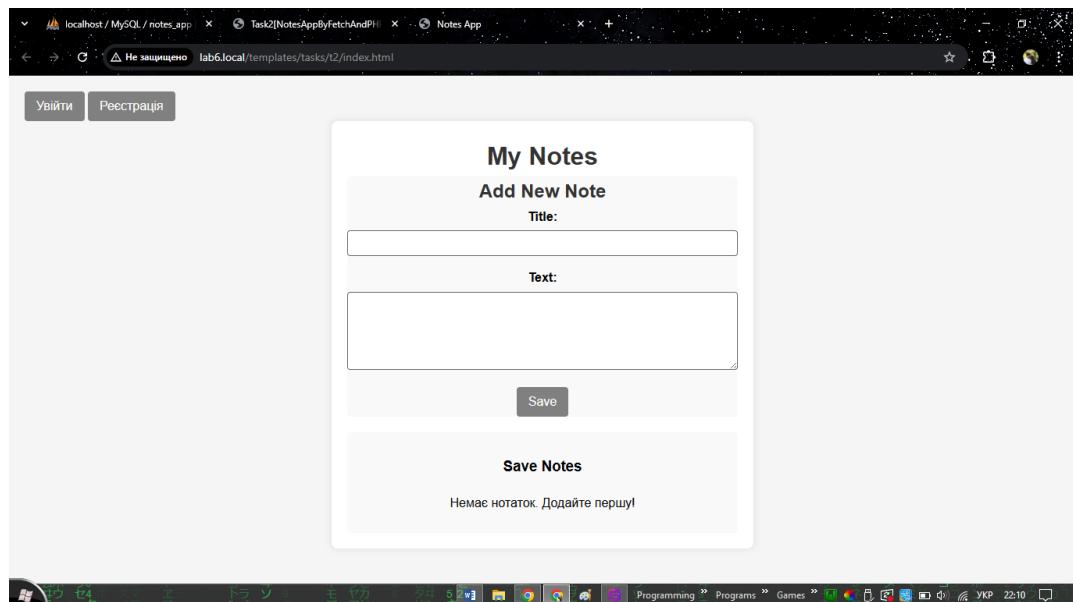


Рис.2.2 Сторінка веб-додатку Notes App

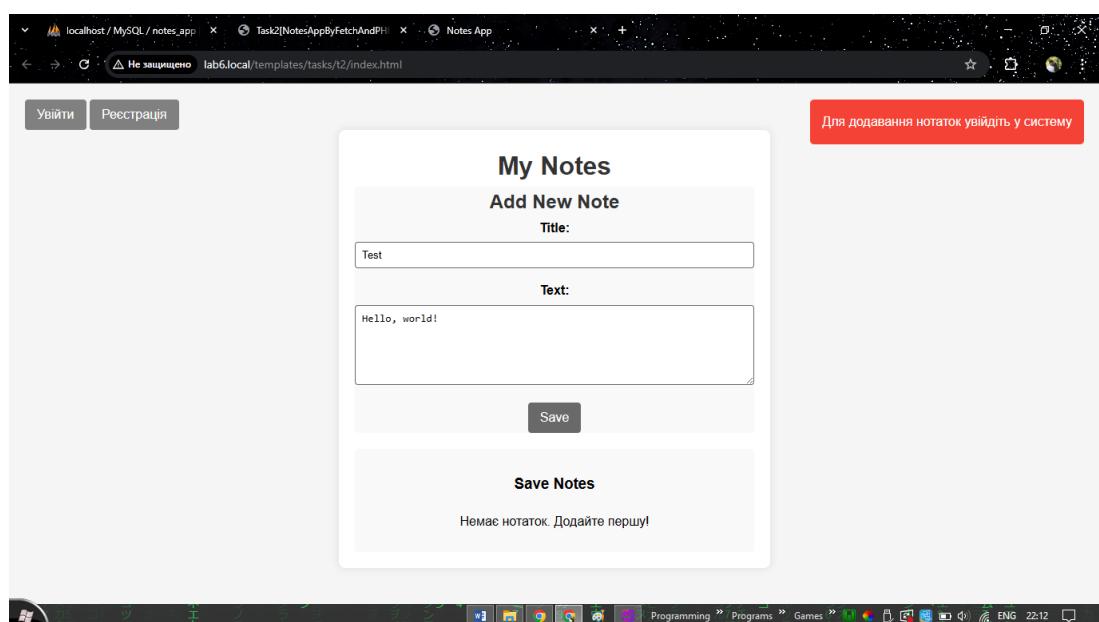


Рис.2.3 Заборона вводити нотатки без авторизації

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25.121.08.000 – Пр6

Арк.

30

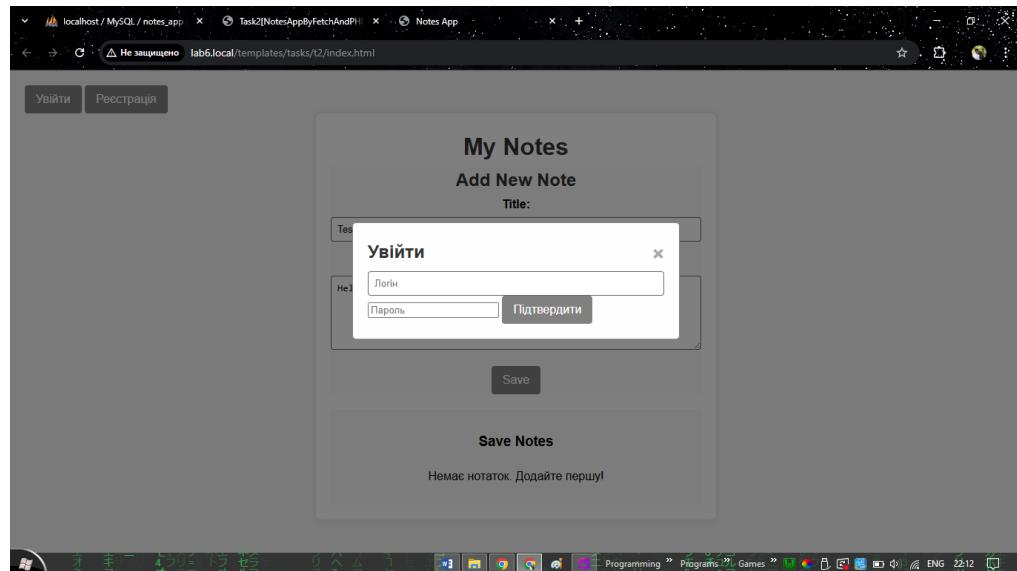


Рис.2.4 Вікно входу в систему

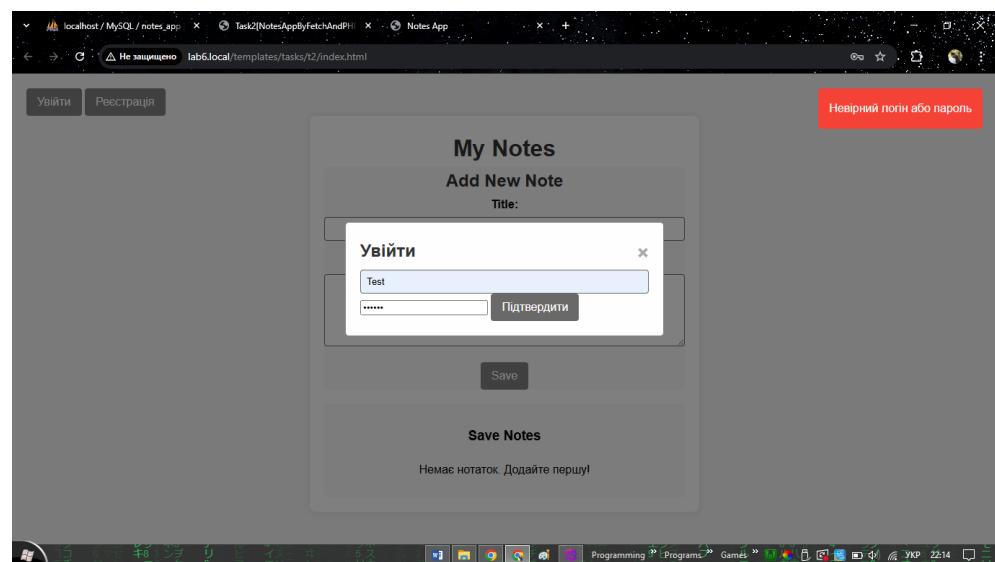


Рис.2.5 Помилка входу незареєстрованого користувача

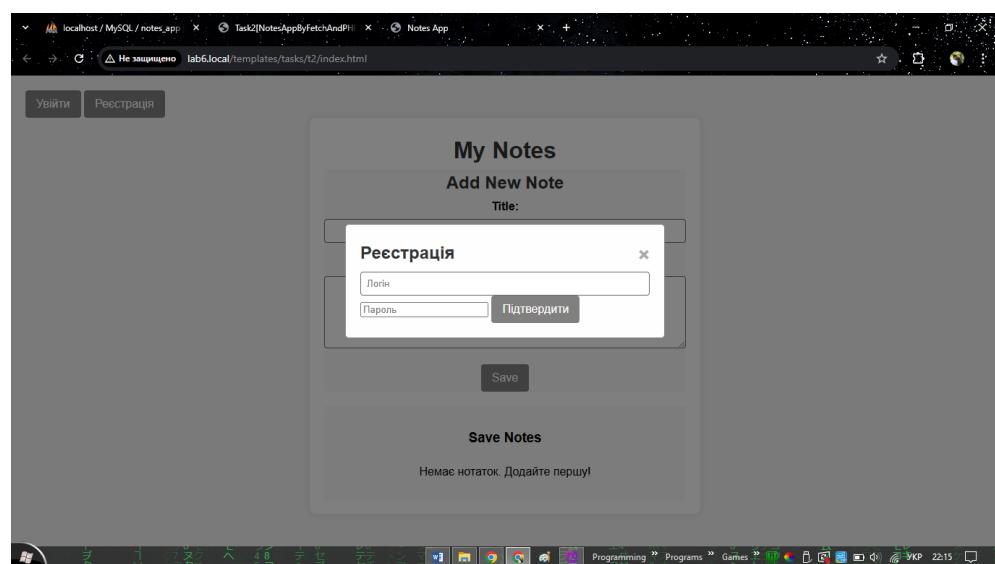


Рис.2.6 Вікно реєстрації

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25.121.08.000 – Пр6

Арк.

31

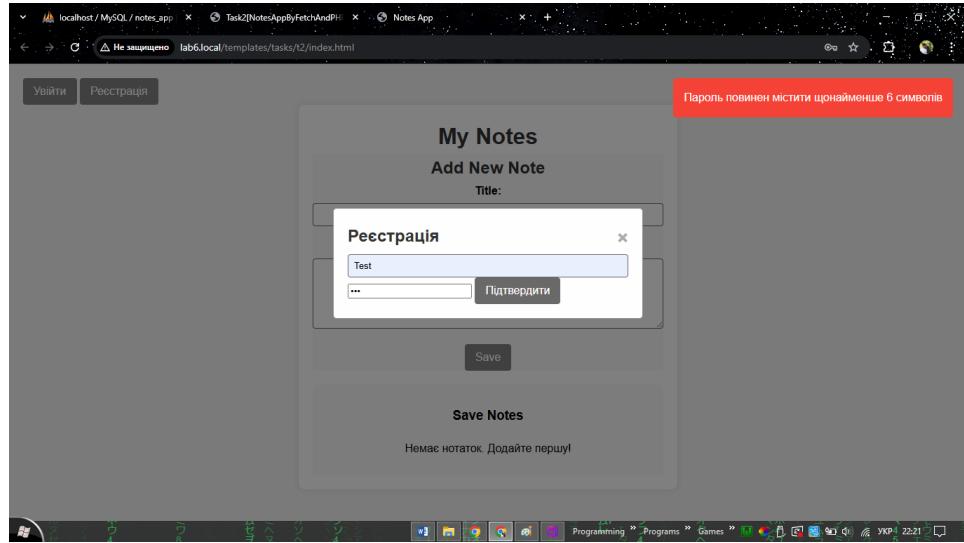


Рис.2.7 Помилка правильності довжини паролю

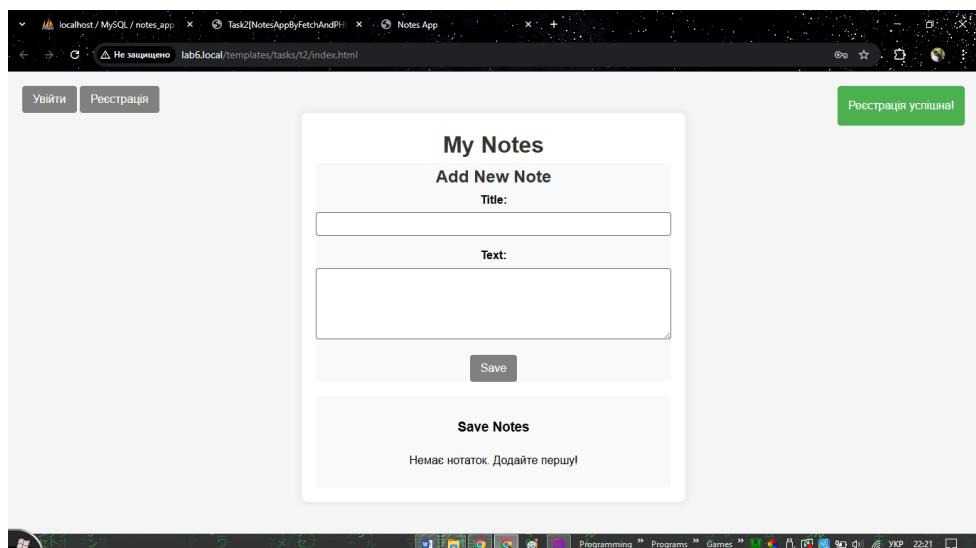


Рис.2.8 Успішна реєстрація

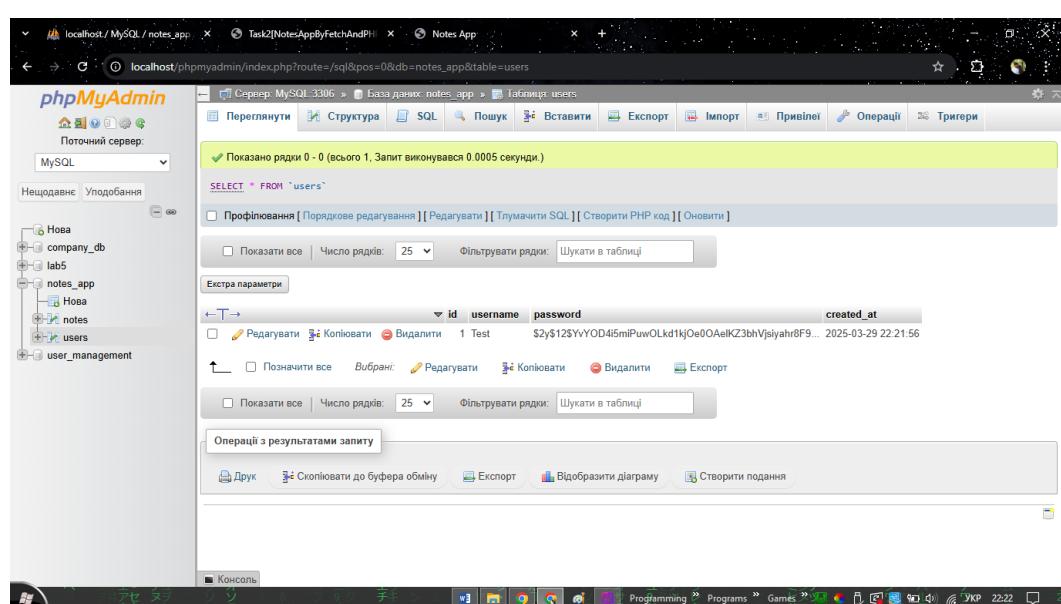


Рис.2.9 Доданий користувач в БД

		Дзінзілевич Д.О.			Арк.
		Ковтун В.В.			
Змн.	Арк.	№ докум.	Підпис	Дата	32

ДУ «Житомирська політехніка».25.121.08.000 – Пр6

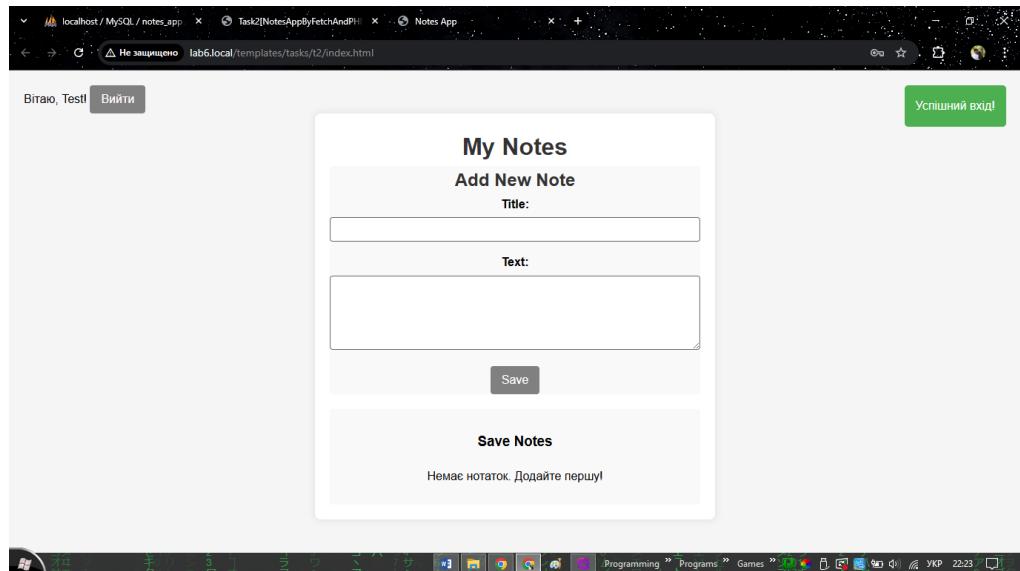


Рис.2.10 Успішний вхід

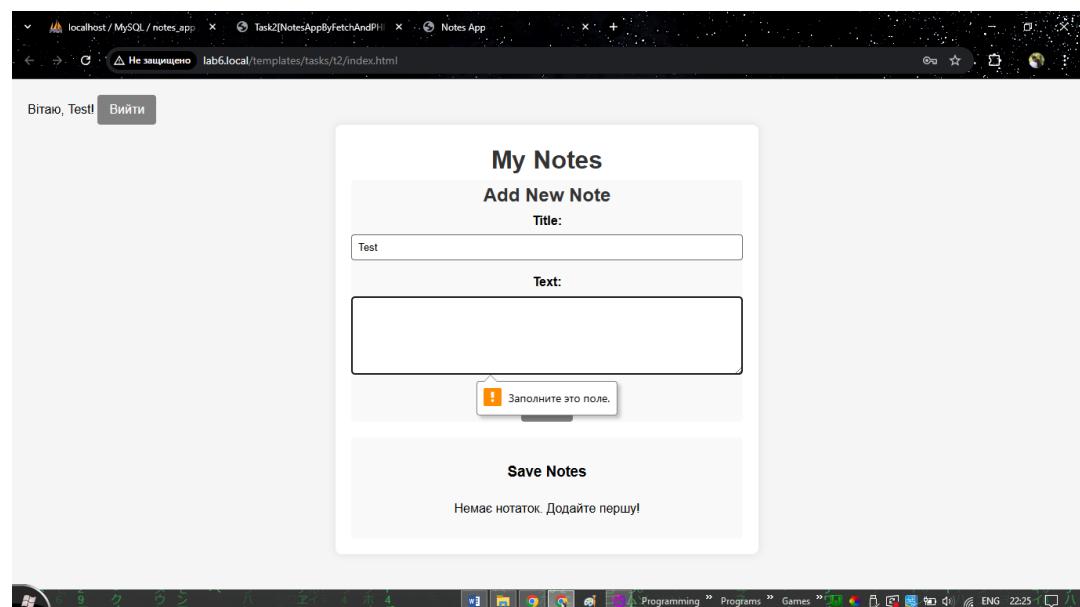


Рис.2.11 Спроба додавання замітки з незаповненим полем

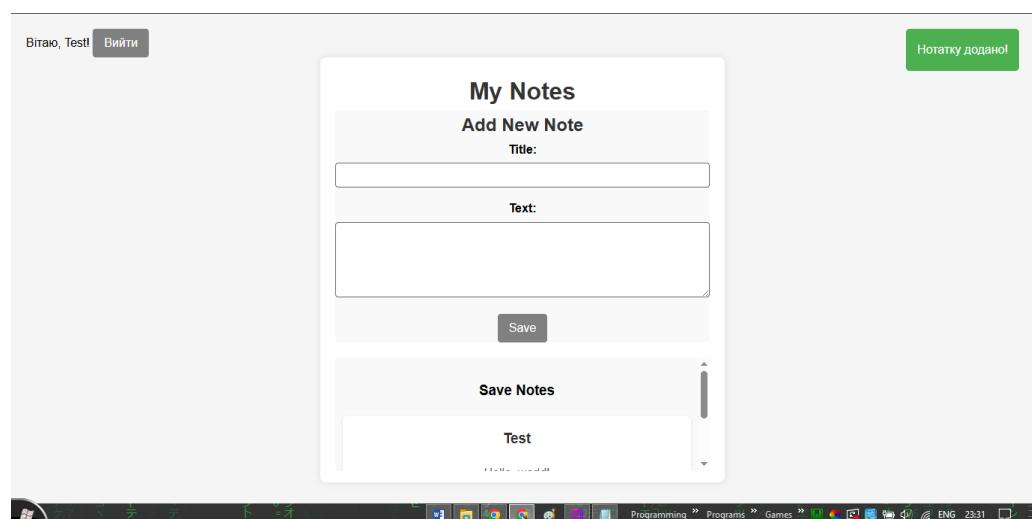


Рис.2.12 Додана замітка в додатку

Змн.	Арк.	Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				33

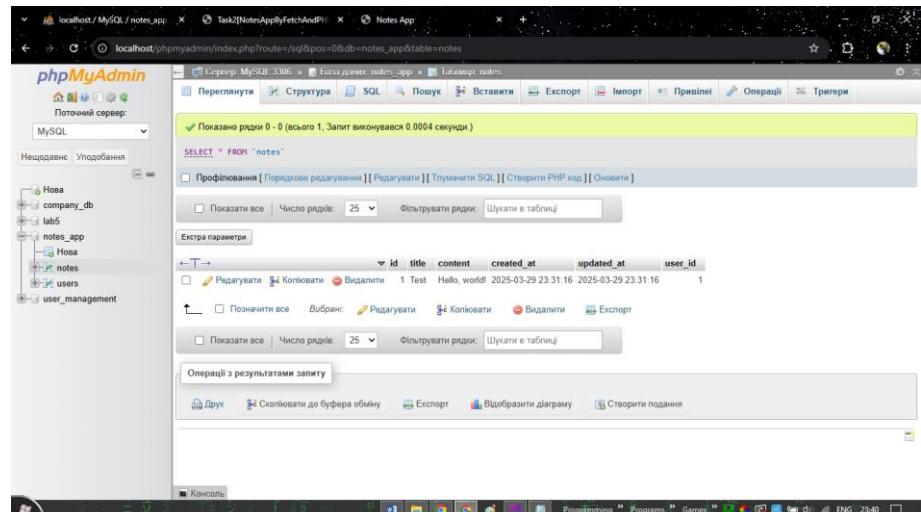


Рис.2.13 Додана замітка в БД

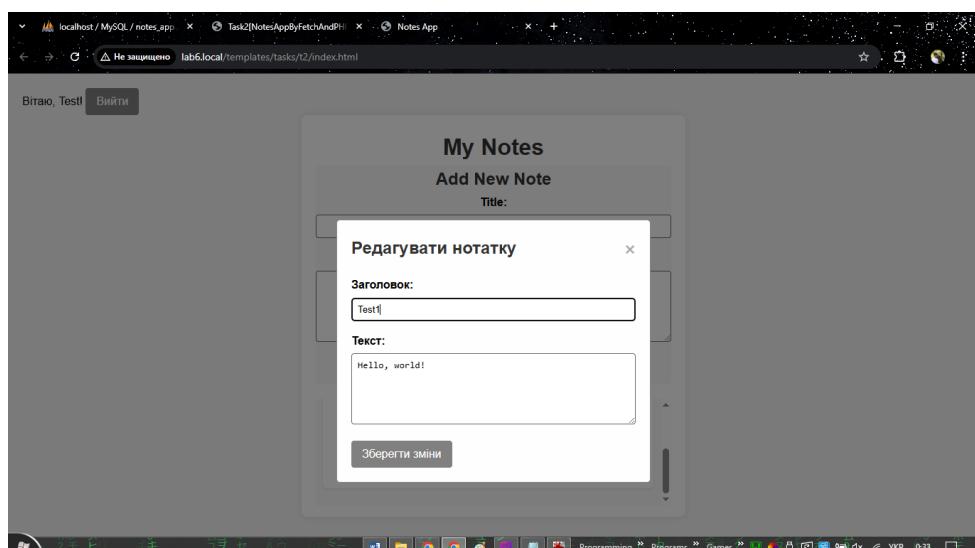


Рис.2.14 Редагування замітки

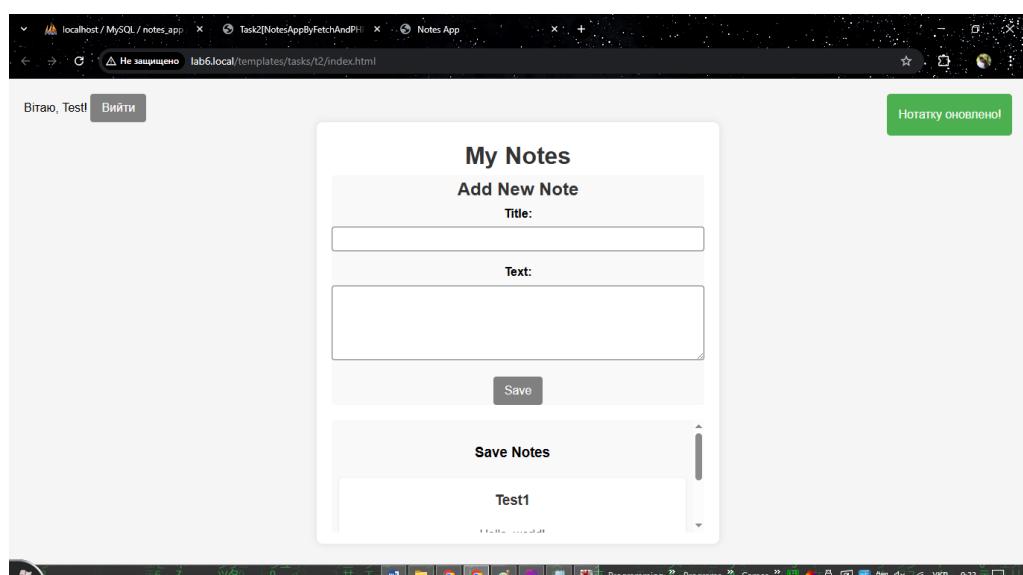


Рис.2.15 Змінені дані в додатку

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25.121.08.000 – Пр6

Арк.  
34

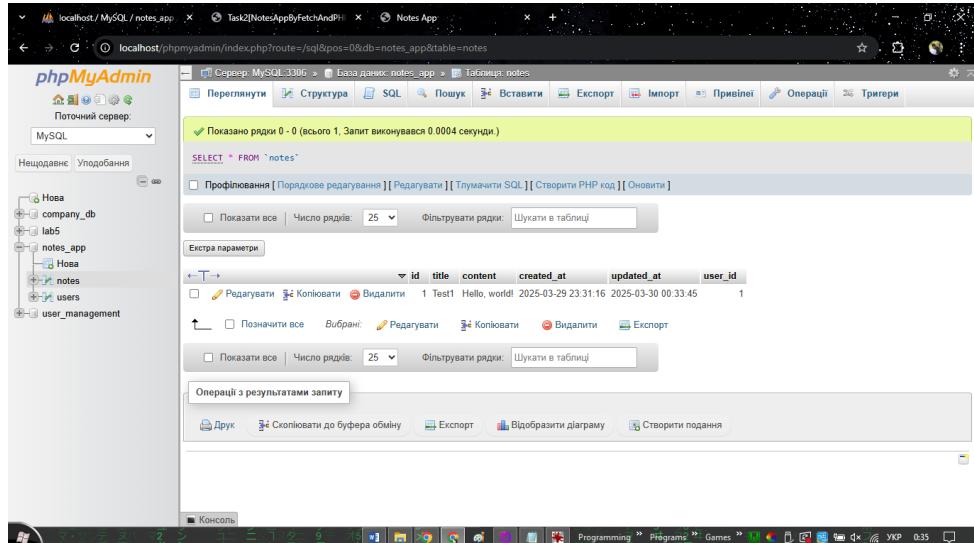


Рис.2.16 Змінені дані в БД

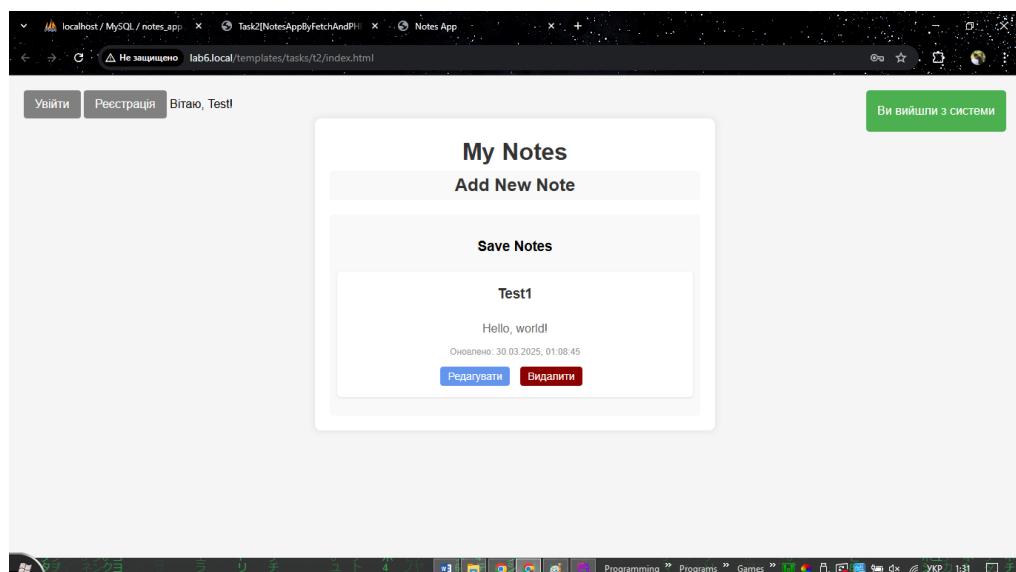


Рис.2.17 Вихід з акаунту

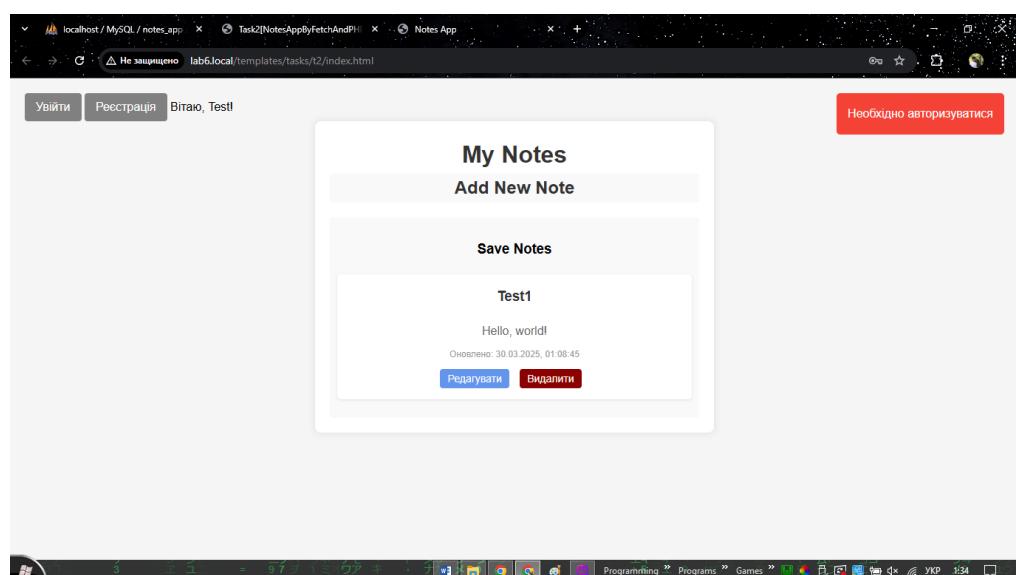


Рис.1.18 Неможливість змінювати замітки без авторизації

		Дзінзілевич Д.О.		
		Ковтун В.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25.121.08.000 – Пр6

Арк.

35

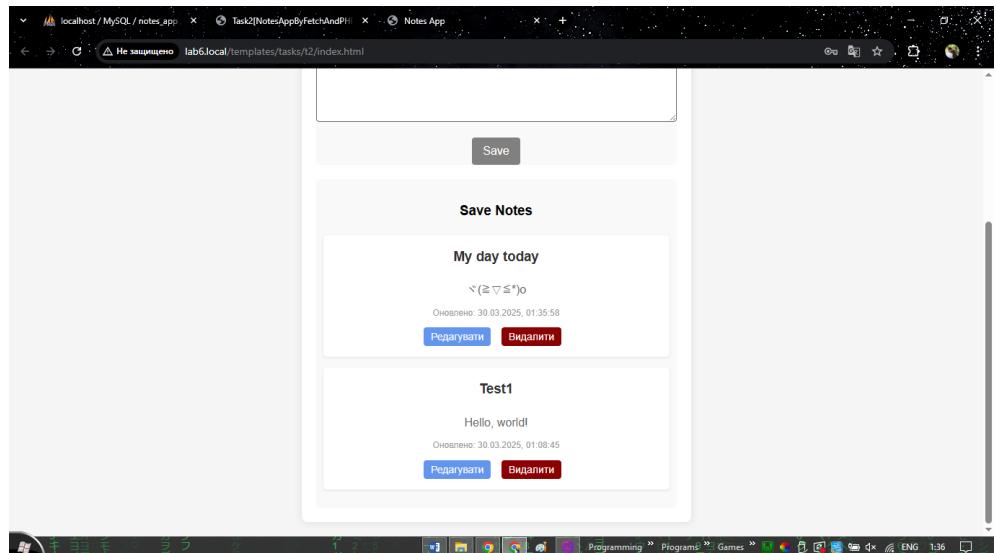


Рис.2.19 Вхід під іншим акаунтом і додавання нової нотатки

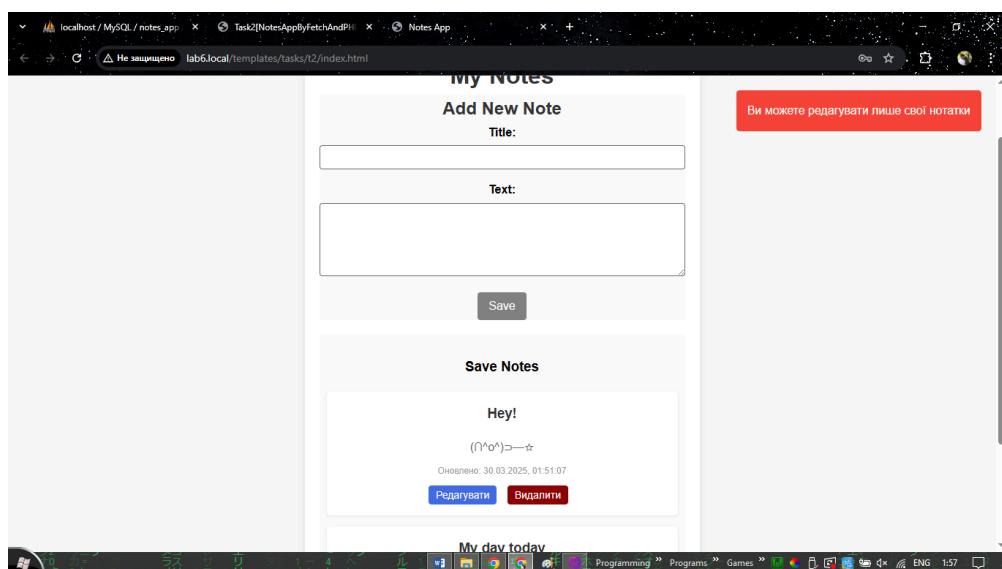


Рис.1.20 Неможливість змінювати чужу нотатку

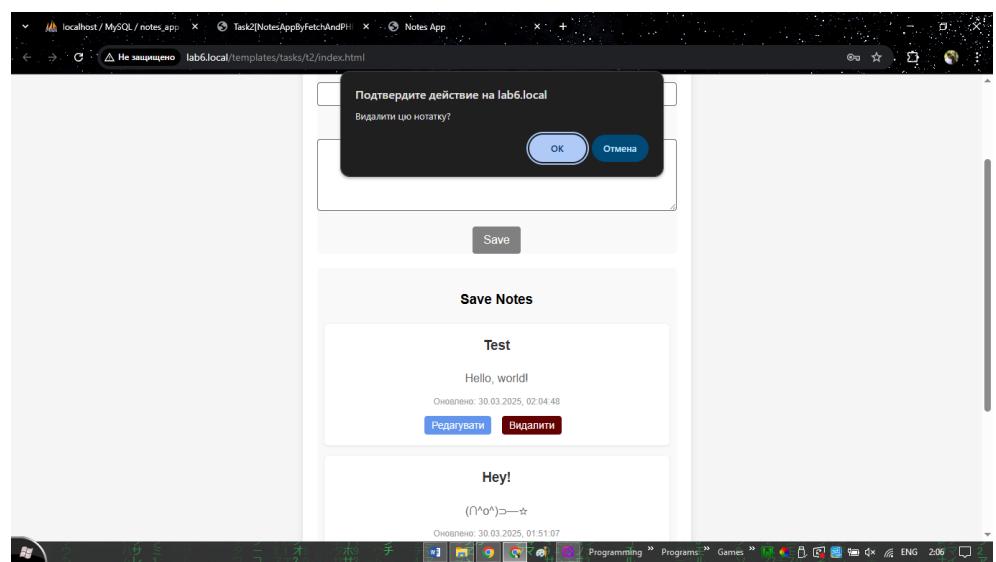


Рис.2.21 Видалення нотатки

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

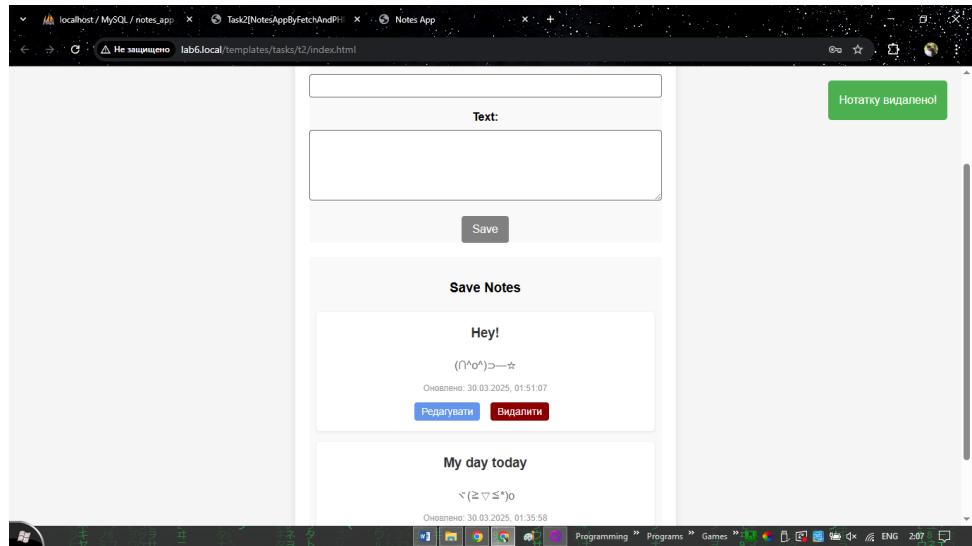


Рис.2.22 Повідомлення про успіх видалення

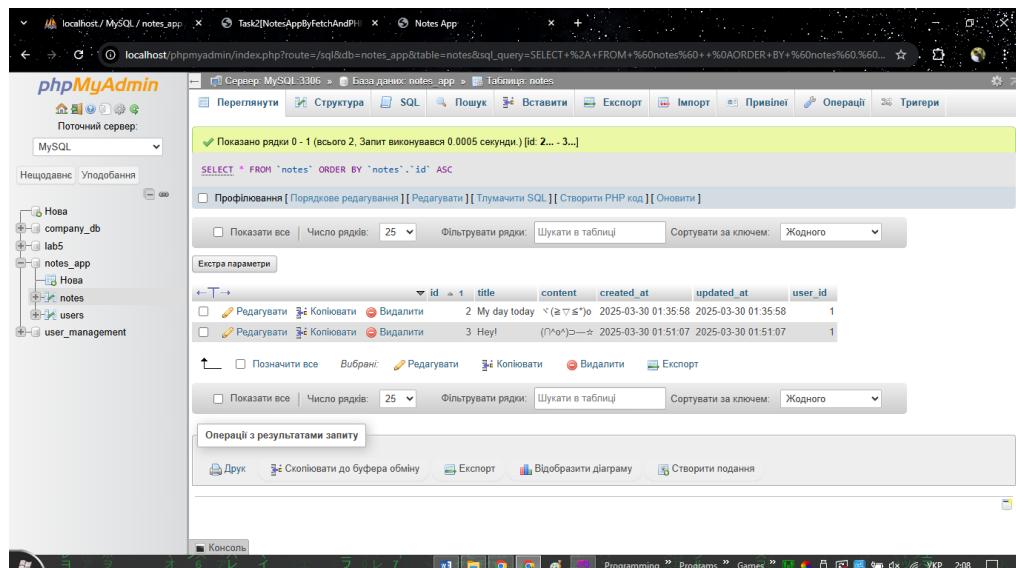


Рис.2.23 Видалені дані з БД

**Висновки:** ознайомилися з асинхронними запитами на сервер та їх обробкою.

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр6	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		37