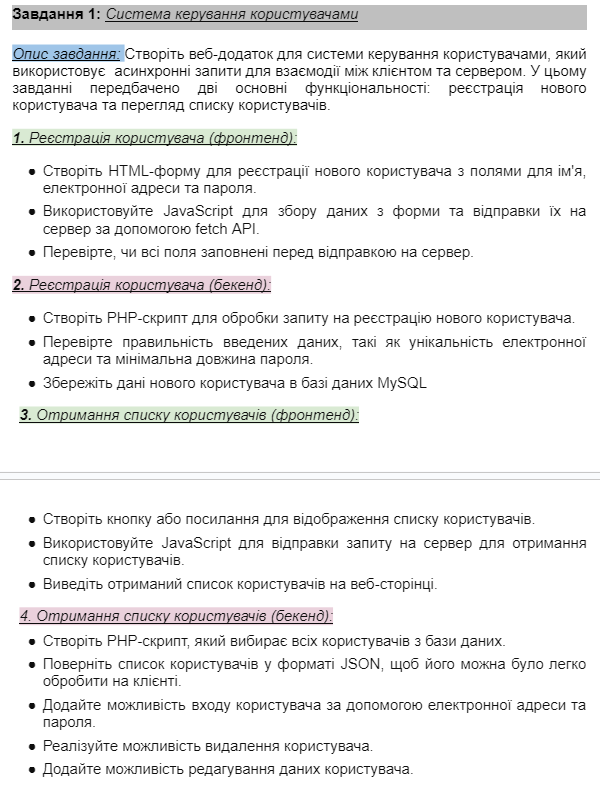
**Лабораторна робота №6**

**Тема: Асинхронні запити на сервер та їх обробка**

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

**Завдання на лабораторну роботу**



Лістинг програми(index.html)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>User Registration</title>

</head>

<body>

<h2>Registration Form</h2>

<form id="registrationForm">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required><br><br>

<button type="submit">Register</button>

</form>

<div id="message"></div>

<script>

document.getElementById('registrationForm').addEventListener('submit', function(event) {

event.preventDefault();

let formData = new FormData(this);

fetch('register.php', {

method: 'POST',

body: formData

})

.then(response => response.json())

.then(data => {

document.getElementById('message').textContent = data.message;

if (data.success) {

this.reset();

}

})

.catch(error => console.error('Error:', error));

});

</script>

<h2>Login Form</h2>

<form id="loginForm">

<label for="loginEmail">Email:</label>

<input type="email" id="loginEmail" name="email" required><br><br>

<label for="loginPassword">Password:</label>

<input type="password" id="loginPassword" name="password" required><br><br>

<button type="submit">Login</button>

</form>

<div id="loginMessage"></div>

<script>

document.getElementById('loginForm').addEventListener('submit', function(event) {

event.preventDefault();

let formData = new FormData(this);

fetch('login.php', {

method: 'POST',

body: formData

})

.then(response => response.json())

.then(data => {

document.getElementById('loginMessage').textContent = data.message;

if (data.success) {

window.location.href = 'user\_list.html';

}

})

.catch(error => console.error('Error:', error));

});

</script>

</body>

</html>

Лістинг програми(user\_list.html)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>User List</title>

</head>

<body>

<h2>User List</h2>

<button id="getUserList">Get User List</button>

<ul id="userList"></ul>

<h2>Edit User</h2>

<form id="editUserForm">

<label for="editEmail">Email:</label>

<input type="email" id="editEmail" name="email" required><br><br>

<label for="editName">New Name:</label>

<input type="text" id="editName" name="name" required><br><br>

<label for="editPassword">New Password:</label>

<input type="password" id="editPassword" name="password" required><br><br>

<button type="submit">Edit User</button>

</form>

<div id="editMessage"></div>

<h2>Delete User</h2>

<form id="deleteUserForm">

<label for="deleteEmail">Email:</label>

<input type="email" id="deleteEmail" name="email" required><br><br>

<button type="submit">Delete User</button>

</form>

<div id="deleteMessage"></div>

<script>

document.getElementById('getUserList').addEventListener('click', function() {

fetch('get\_users.php')

.then(response => {

if (!response.ok) {

throw new Error('Network response was not ok');

}

return response.json();

})

.then(data => {

let userList = document.getElementById('userList');

userList.innerHTML = '';

data.forEach(user => {

let li = document.createElement('li');

li.textContent = `${user.name} - ${user.email}`;

userList.appendChild(li);

});

})

.catch(error => console.error('Error fetching data:', error));

});

document.getElementById('editUserForm').addEventListener('submit', function(event) {

event.preventDefault();

let formData = {

email: document.getElementById('editEmail').value,

name: document.getElementById('editName').value,

password: document.getElementById('editPassword').value

};

fetch('edit\_user.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(formData)

})

.then(response => response.json())

.then(data => {

document.getElementById('editMessage').textContent = data.message;

if (data.success) {

document.getElementById('editUserForm').reset();

document.getElementById('getUserList').click();

}

})

.catch(error => console.error('Error:', error));

});

document.getElementById('deleteUserForm').addEventListener('submit', function(event) {

event.preventDefault();

let formData = {

email: document.getElementById('deleteEmail').value

};

fetch('delete\_user.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(formData)

})

.then(response => response.json())

.then(data => {

document.getElementById('deleteMessage').textContent = data.message;

if (data.success) {

document.getElementById('deleteUserForm').reset();

document.getElementById('getUserList').click();

}

})

.catch(error => console.error('Error:', error));

});

</script>

<button id="logout">Logout</button>

<script>

document.getElementById('logout').addEventListener('click', function() {

fetch('logout.php')

.then(response => response.json())

.then(data => {

alert(data.message);

window.location.href = 'index.html';

})

.catch(error => console.error('Error:', error));

});

</script>

</body>

</html>

Лістинг програми(login.php)

<?php

session\_start();

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "user\_management";

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

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$email = $\_POST['email'];

$password = $\_POST['password'];

$sql = "SELECT \* FROM users WHERE email='$email'";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

$row = $result->fetch\_assoc();

if (password\_verify($password, $row['password'])) {

$\_SESSION['loggedin'] = true;

$\_SESSION['email'] = $email;

$response = array("success" => true, "message" => "Login successful");

} else {

$response = array("success" => false, "message" => "Invalid password");

}

} else {

$response = array("success" => false, "message" => "Email not found");

}

echo json\_encode($response);

$conn->close();

Лістинг програми(register.php)

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "user\_management";

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

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$name = $\_POST['name'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$sql = "SELECT \* FROM users WHERE email='$email'";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

$response = array("success" => false, "message" => "Email already exists");

echo json\_encode($response);

exit();

}

if (strlen($password) < 6) {

$response = array("success" => false, "message" => "Password should be at least 6 characters long");

echo json\_encode($response);

exit();

}

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

$sql = "INSERT INTO users (name, email, password) VALUES ('$name', '$email', '$hashed\_password')";

if ($conn->query($sql) === TRUE) {

$response = array("success" => true, "message" => "User registered successfully");

} else {

$response = array("success" => false, "message" => "Error: " . $sql . "<br>" . $conn->error);

}

echo json\_encode($response);

$conn->close();

Лістинг програми(logout.php)

<?php

session\_start();

session\_unset();

session\_destroy();

$response = array("success" => true, "message" => "Logout successful");

echo json\_encode($response);

Лістинг програми(delete\_user.php)

<?php

session\_start();

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "user\_management";

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

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if (!isset($\_SESSION['loggedin']) || !$\_SESSION['loggedin']) {

header('HTTP/1.1 403 Forbidden');

echo json\_encode(array("message" => "Unauthorized"));

exit();

}

$data = json\_decode(file\_get\_contents("php://input"), true);

$email = $data['email'];

$sql = "DELETE FROM users WHERE email='$email'";

if ($conn->query($sql) === TRUE) {

$response = array("success" => true, "message" => "User deleted successfully");

} else {

$response = array("success" => false, "message" => "Error: " . $sql . "<br>" . $conn->error);

}

echo json\_encode($response);

$conn->close();

Лістинг програми(edit\_user.php)

<?php

session\_start();

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "user\_management";

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

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if (!isset($\_SESSION['loggedin']) || !$\_SESSION['loggedin']) {

header('HTTP/1.1 403 Forbidden');

echo json\_encode(array("message" => "Unauthorized"));

exit();

}

$data = json\_decode(file\_get\_contents("php://input"), true);

$email = $data['email'];

$newName = $data['name'];

$newPassword = $data['password'];

$hashed\_password = password\_hash($newPassword, PASSWORD\_DEFAULT);

$sql = "UPDATE users SET name='$newName', password='$hashed\_password' WHERE email='$email'";

if ($conn->query($sql) === TRUE) {

$response = array("success" => true, "message" => "User updated successfully");

} else {

$response = array("success" => false, "message" => "Error: " . $sql . "<br>" . $conn->error);

}

echo json\_encode($response);

$conn->close();

Лістинг програми(get\_users.php)

<?php

session\_start();

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "user\_management";

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

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if (!isset($\_SESSION['loggedin']) || !$\_SESSION['loggedin']) {

header('HTTP/1.1 403 Forbidden');

echo json\_encode(array("message" => "Unauthorized"));

exit();

}

$sql = "SELECT name, email FROM users";

$result = $conn->query($sql);

$userList = array();

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

$userList[] = array("name" => $row["name"], "email" => $row["email"]);

}

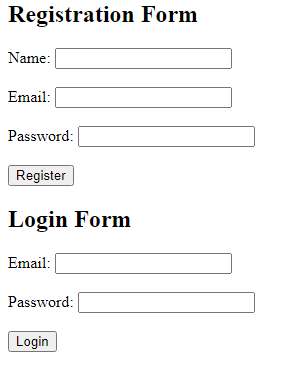
}

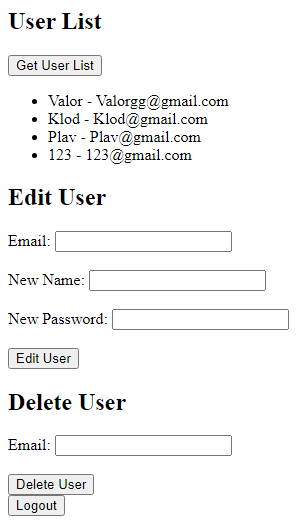
header('Content-Type: application/json');

echo json\_encode($userList);

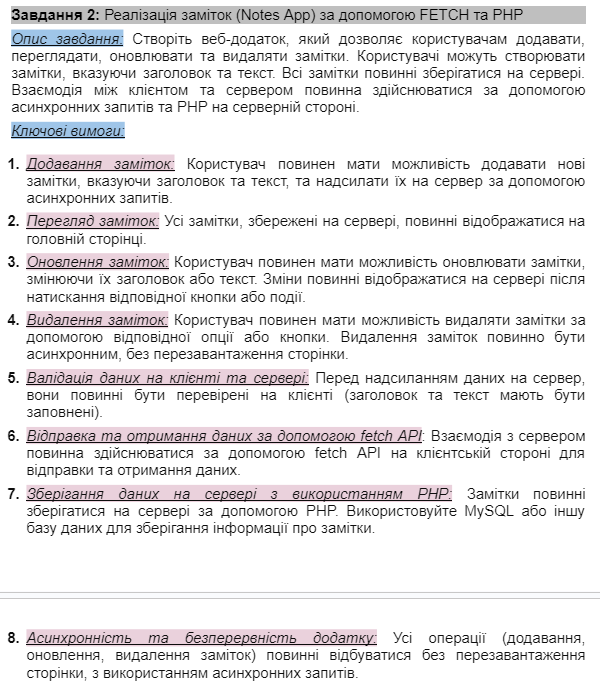
$conn->close();

Результат виконання програми:









Лістинг програми(index.html):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Notes App</title>

</head>

<body>

<h1>Notes App</h1>

<form id="noteForm">

<input type="text" id="noteTitle" placeholder="Заголовок" required>

<textarea id="noteContent" placeholder="Текст замітки" required></textarea>

<button type="submit">Додати замітку</button>

</form>

<div id="notesList"></div>

<script>

const noteForm = document.getElementById('noteForm');

const notesList = document.getElementById('notesList');

noteForm.addEventListener('submit', function(e) {

e.preventDefault();

const title = document.getElementById('noteTitle').value;

const content = document.getElementById('noteContent').value;

fetch('add\_note.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({ title, content }),

})

.then(response => response.json())

.then(data => {

if (data.status === 'success') {

alert('Замітка додана успішно');

noteForm.reset();

fetchNotes();

} else {

alert(data.message);

}

})

.catch(error => console.error('Помилка:', error));

});

function fetchNotes() {

fetch('get\_notes.php')

.then(response => response.json())

.then(notes => {

notesList.innerHTML = '';

notes.forEach(note => {

const noteElement = document.createElement('div');

noteElement.innerHTML = `

<h3>${note.title}</h3>

<p>${note.content}</p>

<button onclick="updateNote(${note.id})">Оновити</button>

<button onclick="deleteNote(${note.id})">Видалити</button>

`;

notesList.appendChild(noteElement);

});

})

.catch(error => console.error('Помилка:', error));

}

function updateNote(id) {

const newTitle = prompt('Введіть новий заголовок:');

const newContent = prompt('Введіть новий текст:');

fetch('update\_note.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({ id, title: newTitle, content: newContent }),

})

.then(response => response.json())

.then(data => {

if (data.status === 'success') {

alert('Замітка оновлена успішно');

fetchNotes();

} else {

alert('Помилка при оновленні замітки');

}

})

.catch(error => console.error('Помилка:', error));

}

function deleteNote(id) {

if (confirm('Ви впевнені, що хочете видалити цю замітку?')) {

fetch('delete\_note.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({ id }),

})

.then(response => response.json())

.then(data => {

if (data.status === 'success') {

alert('Замітка видалена успішно');

fetchNotes();

} else {

alert('Помилка при видаленні замітки');

}

})

.catch(error => console.error('Помилка:', error));

}

}

fetchNotes();

</script>

</body>

</html>

Лістинг програми(get\_notes.php):

<?php

$host = 'localhost';

$dbname = 'notes\_app';

$username = 'root';

$password = '';

try {

$db = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);

} catch (PDOException $e) {

die('Помилка підключення до бази даних: ' . $e->getMessage());

}

$stmt = $db->query("SELECT \* FROM notes");

$notes = $stmt->fetchAll(PDO::FETCH\_ASSOC);

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

echo json\_encode($notes);

Лістинг програми(update\_note.php):

<?php

$host = 'localhost';

$dbname = 'notes\_app';

$username = 'root';

$password = '';

try {

$db = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);

} catch (PDOException $e) {

die('Помилка підключення до бази даних: ' . $e->getMessage());

}

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$data = json\_decode(file\_get\_contents('php://input'), true);

$id = $data['id'];

$newTitle = $data['title'];

$newContent = $data['content'];

if (!empty($id) && !empty($newTitle) && !empty($newContent)) {

$stmt = $db->prepare("UPDATE notes SET title = ?, content = ? WHERE id = ?");

$stmt->execute([$newTitle, $newContent, $id]);

echo json\_encode(['status' => 'success']);

} else {

echo json\_encode(['status' => 'error', 'message' => 'Помилка: невірні дані для оновлення замітки']);

}

}

Лістинг програми(delete\_note.php):

<?php

$host = 'localhost';

$dbname = 'notes\_app';

$username = 'root';

$password = '';

try {

$db = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);

} catch (PDOException $e) {

die('Помилка підключення до бази даних: ' . $e->getMessage());

}

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$data = json\_decode(file\_get\_contents('php://input'), true);

$id = $data['id'];

if (!empty($id)) {

$stmt = $db->prepare("DELETE FROM notes WHERE id = ?");

$stmt->execute([$id]);

echo json\_encode(['status' => 'success']);

} else {

echo json\_encode(['status' => 'error', 'message' => 'Помилка: невірні дані для видалення замітки']);

}

}

?>

Лістинг програми(add\_note.php):

<?php

$host = 'localhost';

$dbname = 'notes\_app';

$username = 'root';

$password = '';

try {

$db = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);

} catch (PDOException $e) {

die('Помилка підключення до бази даних: ' . $e->getMessage());

}

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$data = json\_decode(file\_get\_contents('php://input'), true);

$title = $data['title'];

$content = $data['content'];

if (!empty($title) && !empty($content)) {

$stmt = $db->prepare("INSERT INTO notes (title, content) VALUES (?, ?)");

$stmt->execute([$title, $content]);

echo json\_encode(['status' => 'success']);

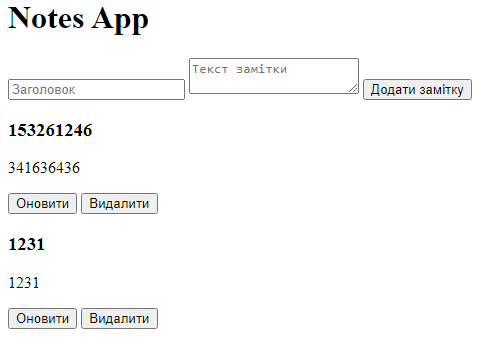
} else {

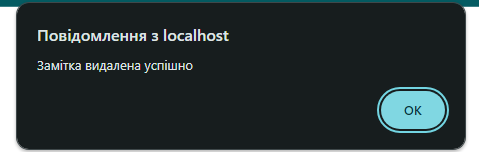
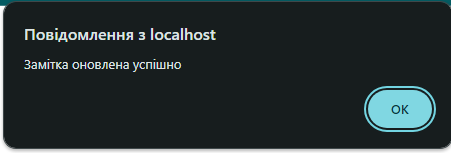
echo json\_encode(['status' => 'error', 'message' => 'Помилка: заголовок та текст не можуть бути порожніми']);

}

}

Результат виконання програми:





Посилання на github: https://github.com/loorkky/Backend.git