

ЛАБОРАТОРНА РОБОТА № 6

Тема: Асинхронні запити на сервер та їх обробка

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

Завдання 1: Система керування користувачами

Опис завдання: 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 (фронтенд):

- 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 (бекенд):

- 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 (фронтенд):

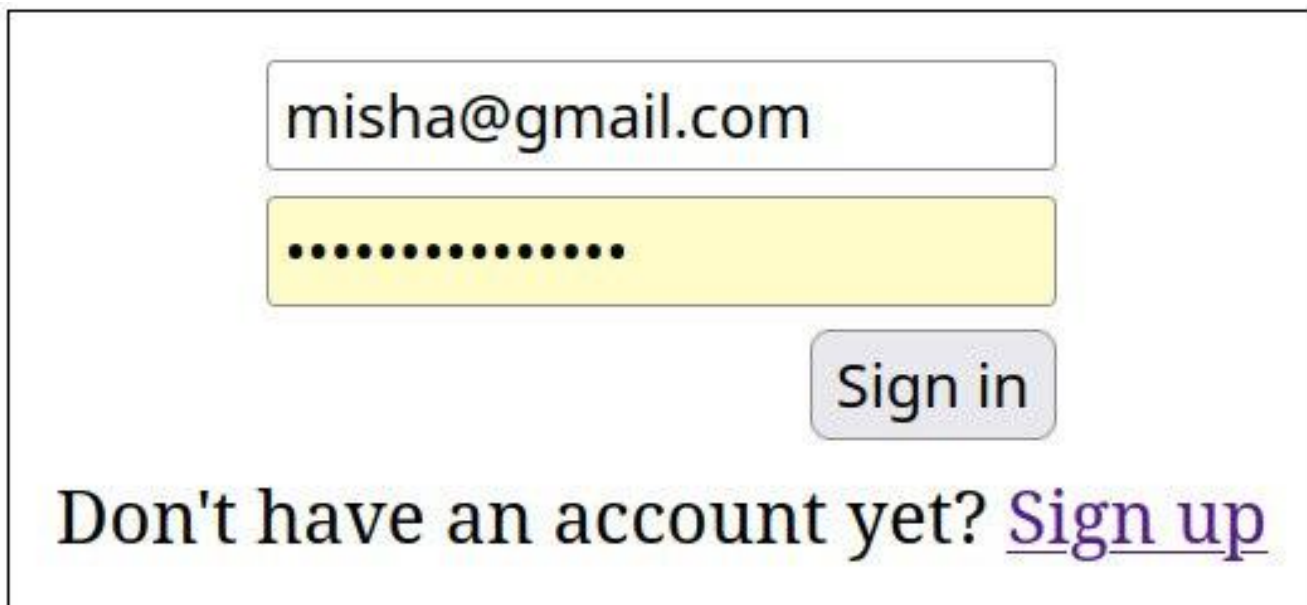
- Створіть кнопку або посилання для відображення списку користувачів.
- Використовуйте JavaScript для відправки запиту на сервер для отримання списку користувачів. Виведіть отриманий список користувачів на веб-сторінці.

4. Отримання списку користувачів (бекенд):

- 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.

Результат:

					ДУ«Житомирська політехніка».25.121.27.000 – Лр6			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Семенчук О.А.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Дмитренко І.А.						Аркушів
Керівник								1
Н. контр.								13
Зав. каф.							ФІКТ Гр. ІПЗ-23-1[2]	



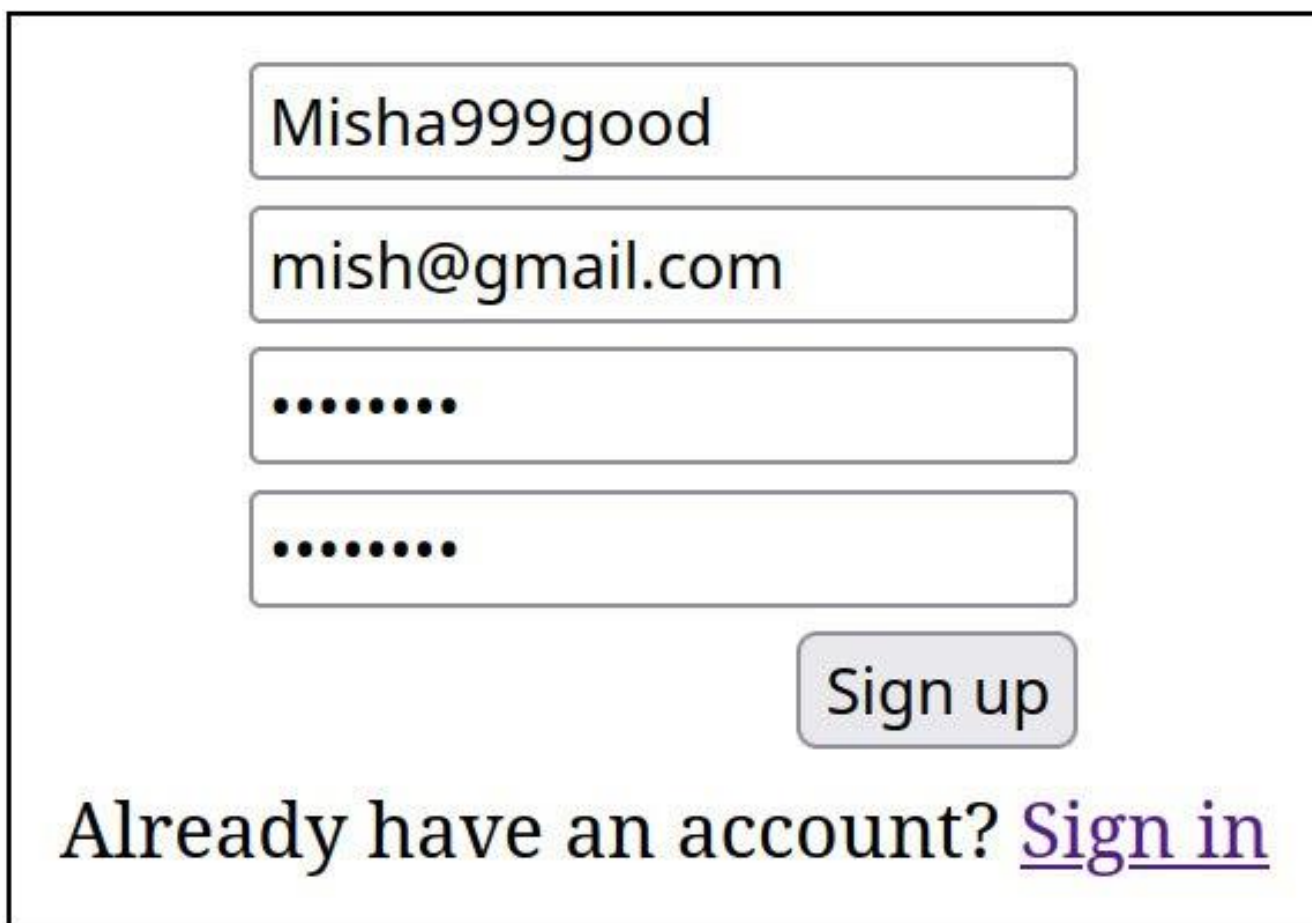
misha@gmail.com

.....

Sign in

Don't have an account yet? [Sign up](#)

Рис. 1 Сторінка входу



Misha999good

mish@gmail.com

.....

.....

Sign up

Already have an account? [Sign in](#)

Рис. 2 Сторінка реєстрації

Рис. 3 Сторінка профілю

Лістинг файлу login.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Registration</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="form-container">
    <div class="inputs">
      <div>
        <input id="email-address" required type="text"
name="email-address" placeholder="Email address">
      </div>

      <div>
        <input id="password" required type="password"
minlength="8" maxlength="32" name="password"
placeholder="Password">
      </div>

      <div style="width: 100%; display: flex; justify-content:
flex-end">
        <button onclick="login()" style="margin-left:
auto">Sign up</button>
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        </div>
    </div>

    <p class="prompt">Don't have an account yet? <a
href="register.html">Sign up</a></p>
    </div>
</body>
<script>
    const email_address_input = document.querySelector("#email-
address");
    const password_input = document.querySelector("#password");

    async function login()
    {
        const email_address = email_address_input.value;
        const password = password_input.value;

        const response = await fetch("http://localhost/login", {
            method: "POST",
            headers: { "Content-Type": "application/json", },
            body: JSON.stringify({ email_address, password }),
        });

        const user_id = await response.text();

        if(response.status == 200)
        {
            sessionStorage.setItem("user_id", user_id);
            window.location = "profile.html";
        }
    }
</script>
</html>

```

Лістинг файлу register.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Registration</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <form>
        <div class="form-container">
            <div class="inputs">
                <div>
                    <input id="user-name" required minlength="4"
maxlength="16" type="text" name="user-name"
placeholder="Username">
                </div>

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

        <div>
            <input id="email-address" required type="email"
name="email-address" placeholder="Email Address">
        </div>

        <div>
            <input id="password" required type="password"
minlength="8" maxlength="32" name="password"
            placeholder="Password">
        </div>

        <div>
            <input required type="password" minlength="8"
maxlength="32" name="confirm-password"
            placeholder="Confirm Password">
        </div>

        <div id="confirm-password" style="width: 100%;
display: flex; justify-content: flex-end">
            <input style="margin-left: auto" type="submit"
onclick="registerUser(event)" value="Sign up">
        </div>
    </div>

    <p class="prompt">Already have an account? <a
href="login.html">Sign in</a></p>
</div>
</form>
</body>
<script defer>
    async function registerUser(event)
    {
        event.preventDefault();

        const user_name = document.querySelector("#user-
name").value;
        const email_address = document.querySelector("#email-
address").value;
        const password = document.querySelector("#password").value;
        const confirm_password = document.querySelector("#confirm-
password").value;

        if (confirm_password !== password)
        {
            alert("Passwords do not match!");
            return;
        }

        const data =
        {
            user_name,
            email_address,
            password,

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    const response = await fetch("http://localhost/users",
    {
        method: "POST",
        headers: {"Content-Type": "application/json", },
        body: JSON.stringify(data),
    });
}
</script>
</html>

    Лістинг файлу profile.html:
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title></title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="form-container">
        <div>
            <input id="user-name" type="text" required minlength="4"
maxlength="16" placeholder="Username">
        </div>

        <div>
            <input id="email-address" type="text" required
placeholder="Email address">
        </div>

        <div style="width: 100%; display: flex; justify-content:
flex-end">
            <button onclick="saveAccount()">Save</button>
        </div>

        <div style="width: 100%; display: flex; justify-content:
flex-end">
            <button onclick="deleteAccount()">Delete
account</button>
        </div>
    </div>
</body>
<script defer>
    const user_id = sessionStorage.getItem("user_id");

    if (!user_id)
    {
        window.location = "login.html";
    }

    const user_name_input = document.querySelector("#user-name");

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

    const email_address_input = document.querySelector("#email-
address");

    (async() =>
    {
        const response = await
fetch(`http://localhost/users/${user_id}`);
        const user = await response.json();

        user_name_input.value = user.user_name;
        email_address_input.value = user.email_address;
    })();

    async function deleteAccount()
    {
        let confirmation = confirm("Are you sure you want to delete
your account?");

        if (!confirmation)
        {
            return;
        }

        await fetch(`http://localhost/users/${user_id}`, { method:
"DELETE" });
        sessionStorage.setItem("user_id", "");
        window.location = "login.html";
    }

    async function saveAccount()
    {
        data =
        {
            user_name: user_name_input.value,
            email_address: email_address_input.value,
        }

        const response = await
fetch(`http://localhost/users/${user_id}`,
        {
            method: "PATCH",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify(data),
        });
    }
}
</script>
</html>

```

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

Опис завдання: 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 сер-

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

вері. Взаємодія між клієнтом та сервером повинна здійснюватися за допомогою асинхронних запитів та PHP на серверній стороні.

Ключові вимоги:

- 1. Додавання заміток: Користувач повинен мати можливість додавати нові замітки, вказуючи заголовок та текст, та надсилати їх на сервер за допомогою асинхронних запитів.
- 2. Перегляд заміток: Усі замітки, збережені на сервері, повинні відображатися на головній сторінці.
- 3. Оновлення заміток: Користувач повинен мати можливість оновлювати замітки, змінюючи їх заголовок або текст. Зміни повинні відображатися на сервері після натискання відповідної кнопки або події.
- 4. Видалення заміток: Користувач повинен мати можливість видаляти замітки за допомогою відповідної опції або кнопки. Видалення заміток повинно бути асинхронним, без перезавантаження сторінки.
- 5. Валідація даних на клієнті та сервері: Перед надсиланням даних на сервер, вони повинні бути перевірені на клієнті (заголовок та текст мають бути заповнені).
- 6. 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.
- 7. 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.
- 8. Асинхронність та безперервність додатку: Усі операції (додавання, оновлення, видалення заміток) повинні відбуватися без перезавантаження сторінки, з використанням асинхронних запитів.

Результат:

Meeting with Team Discuss project updates, deadlines, and next steps. Prepare presentation slides.	Edit Delete 2025-03-28 09:43:03
Grocery List Milk, eggs, bread, coffee, pasta, and vegetables. Check for missing spices.	Edit Delete 2025-03-28 09:43:03
title Monday: Cardio. Tuesday: Strength. Wednesday: Yoga. Stay consistent!	Edit Delete 2025-03-28 09:43:03
Book Recommendations Read "Atomic Habits" and "Deep Work." Look for more self-improvement books.	Edit Delete 2025-03-28 09:43:03
Weekend Plans Visit the museum, have dinner with friends, and go for a nature walk.	Edit Delete 2025-03-28 09:43:03
Add Note	

Рис. 4 Результат завдання 2

Лістинг файлу index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
  <link rel="stylesheet" href="styles.css">
```



```

</head>
<body>
  <div class="container"></div>
</body>
<script defer>
  const container = document.querySelector(".container");

  load_notes();

  async function load_notes()
  {
    const response = await fetch("http://localhost/notes");
    const notes = await response.json();

    notes.sort((a, b) => new Date(b.created_at) - new
Date(a.created_at));

    for(let i = 0; i < notes.length; i++)
    {
      item = createItem(notes[i]);

      container.appendChild(item);
    }

    const add_button = document.createElement("div");

    add_button.addEventListener("click", open_add_note_window);

    add_button.classList.add("item");
    add_button.classList.add("add-button");

    add_button.innerHTML = "Add Note";

    container.appendChild(add_button);
  }

  function createItem(note)
  {
    const item = document.createElement("div");
    item.classList.add("item");
    item.setAttribute("data-note-id", note.note_id);

    item.innerHTML = `
      <div class="title-container">
        <h2 class="title">${note.title}</h2>

        <div class="buttons">
          <button onclick="open_edit_note_window(this,
${note.note_id})">Edit</button>
          <button onclick="delete_note(this,
${note.note_id})">Delete</button>
        </div>
    `
  }

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

```

        </div>
        <div class="content-container">
            <div>
                <p class="content">${note.content}</p>
            </div>
            <div class="date-container">
                <p>${note.created_at}</p>
            </div>
        </div>
    `;

    return item;
}

function open_add_note_window()
{
    const contents = `
        <div>
            <input required id="title" type="text"
placeholder="Title">
        </div>
        <div>
            <textarea required id="content" rows="10" cols="50"
type="text" placeholder="Content"></textarea>
        </div>
        <div class="buttons-container">
            <button class="cancel-button">Cancel</button>
            <button onclick="add_note()" class="done-button
cancel-button">Done</button>
        </div>
    `;

    open_modal_window(contents);
}

async function add_note()
{
    const title = document.querySelector("#title").value;
    const content = document.querySelector("#content").value;

    const response = await fetch("http://localhost/notes",
    {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ title, content }),
    });

    const note =
    {
        title,
        content,
        created_at: get_formatted_datetime(),
    }

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        const item = createItem(note);
        container.prepend(item);
    }

    function open_edit_note_window(button, id)
    {
        const item = button.closest(".item");

        const original_title =
item.querySelector(".title").textContent;
        const original_content =
item.querySelector(".content").textContent;

        window.editing_item = item;

        const contents = `
            <div class="edit-window">
                <input id="title" type="text"
value="${original_title}" placeholder="Title">
            </div>
            <div>
                <textarea id="content" rows="10" cols="50"
placeholder="Content">${original_content}</textarea>
            </div>
            <div class="buttons-container">
                <button class="cancel-button">Cancel</button>
                <button onclick="edit_note(${id})" class="done-
button cancel-button">Save</button>
            </div>
        `;

        open_modal_window(contents);
    }

    async function edit_note(id)
    {
        const new_title = document.querySelector("#title").value;
        const new_content =
document.querySelector("#content").value;

        const response = await fetch(`http://localhost/notes/${id}`,
{
            method: "PATCH",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify({ title: new_title, content:
new_content }),
        });

        if (response.ok && window.editing_item)
        {
            window.editing_item.querySelector(".title").textContent
= new_title;

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

window.editing_item.querySelector(".content").textContent =
new_content;
    }
}

function get_formatted_datetime()
{
    const now = new Date();
    const year = now.getFullYear();
    const month = String(now.getMonth() + 1).padStart(2, '0');
    const day = String(now.getDate()).padStart(2, '0');
    const hours = String(now.getHours()).padStart(2, '0');
    const minutes = String(now.getMinutes()).padStart(2, '0');
    const seconds = String(now.getSeconds()).padStart(2, '0');

    return `${year}-${month}-${day}
    ${hours}:${minutes}:${seconds}`;
}

async function delete_note(button, id)
{
    const confirmation = confirm("Are you sure you want to
delete the note!");

    if (!confirmation)
    {
        return;
    }

    const response = await fetch(`http://localhost/notes/${id}`,
{ method: "DELETE" });

    await response.text();

    const item = button.closest(".item");

    if (item) item.remove();
}

function open_modal_window(contents)
{
    const modal_window_background =
document.createElement("div");
    modal_window_background.classList.add("modal-window-
background");

    const modal_window_container =
document.createElement("div");
    modal_window_container.classList.add("modal-window-
container");

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

modal_window_container.innerHTML = contents;

document.body.appendChild(modal_window_background);
document.body.appendChild(modal_window_container);

const cancel_buttons =
modal_window_container.querySelectorAll(".cancel-button");

for (let i = 0; i < cancel_buttons.length; i++)
{
    cancel_buttons[i].addEventListener("click", () =>
    {
        document.body.removeChild(modal_window_background);
        document.body.removeChild(modal_window_container);
    });
}
}
</script>
</html>

```

Git repository: <https://git.ztu.edu.ua/ipz/2023-2027/ipz-23-1/semenchuk-oleksiy/backend/-/tree/lab6>

Висновок: У ході виконання роботи було розглянуто основи роботи з базою даних MySQL за допомогою бібліотеки PDO у мові PHP. Було вивчено підключення до бази даних, виконання запитів, обробку результатів, а також використання підготовлених запитів для забезпечення безпеки та оптимізації роботи з даними.

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр6	Арк.
		Дмитренко І. А.				13
Змн.	Арк.	№ докум.	Підпис	Дата		