# Лабораторна робота 3: Розробка UI з використанням HTML та JS

### **Мета роботи:**

Навчитися створювати простий веб-інтерфейс для взаємодії з RESTful сервером. Освоїти базові принципи роботи з HTTP запитами в JavaScript.

### **Завдання:**

1. Створити HTML-сторінку з формами для введення даних, які будуть відправлятися на сервер.

Приклад:

Поле 1: [\_\_\_\_\_\_\_\_\_\_]

Поле 2: [\_\_\_\_\_\_\_\_\_\_]

[Save]

List:

Element 1 [Edit] [Delete]

Element 2 [Edit] [Delete]

1. Використовуючи JavaScript (зокрема Fetch API), реалізувати функції для відправлення POST, PUT, GET, DELETE запитів до RESTful сервера.
2. Обробляти відповіді від сервера та відображати результати на сторінці.

### **План роботи:**

#### **1. Підготовка HTML сторінки:**

* Створити основну структуру сторінки з формами для введення даних (наприклад, ім'я, вік, ID і т.д.).
* Додати список з елементами які будуть отримані з сервера

#### **2. Додати тестові данні на сервер:**

* У масив серверу додати дані що будуть повертатись на клієнт

#### **3. Реалізація GET запиту:**

* Використати Fetch API для відправлення GET HTTP запиту за всіма елементами.
* Оновити список в HTML документі використовуючи отримані дані

#### **4. Реалізація POST запиту:**

* Додати необхідні поля для відправки даних на сервер.
* Додати кнопку “Save”
* Додати функцію що буде викликатись по натисненню на кнопку.
* Використати Fetch API для відправлення POST HTTP запиту.   
  Приклад:

async function postData() {

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

const age = document.getElementById('age').value;

const response = await fetch(apiUrl, {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ name, age })

});

const result = await response.json();

// update HTML with result

}

* Додати створений елемент у список на сторінці

#### **5. Реалізація PUT запиту:**

* Додати кнопку “Edit” біля кожного елементу списку
* Після натиснення кнопку - відповідний елемент повинен заповнити поля для введення даних біля кнопки “Save”
* Використати Fetch API для відправлення PUT HTTP запиту після натиснення на кнопку Save

#### **6. Реалізація Delete запиту:**

* Додати кнопку “Delete” біля кожного елементу списку
* Використати Fetch API для відправлення Delete HTTP запиту.
* Якщо запит успішний - видалити елеммент з HTML

#### **7. Тестування роботи:**

* Перевірити роботу кожної функції, відправляючи різні запити до RESTful сервера.
* Переконатися, що відповіді сервера коректно обробляються та відображаються на сторінці.