**2.b Develop an AJAX-based form to submit and retrieve data from a mock server**

**Aim:**  
To create a form that submits user input to a mock server using AJAX and retrieves stored data for display.

**Tools Required:**

* Web browser (Chrome/Firefox)
* Text editor (VS Code/Notepad++)
* JSONPlaceholder (online mock API) or json-server (local mock server, optional)

**Procedure:**

1. Create a folder named **ajax\_form**.
2. Inside the folder, create two files:
   * index.html
   * main.js
3. Design a simple HTML form with fields (Name, Email).
4. Use the fetch() API (AJAX) in main.js to submit form data (POST) to the server.
5. Retrieve the list of submitted records (GET) and display them in a <ul>.
6. Save and run the HTML file in a browser.
7. Verify that new entries are displayed dynamically after submission.

**Source Code:**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>AJAX Form</title>

</head>

<body>

<h1>AJAX Form Example</h1>

<form id="contactForm">

<label>Name: <input type="text" name="name" required></label><br><br>

<label>Email: <input type="email" name="email" required></label><br><br>

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

</form>

<h2>Saved Contacts</h2>

<ul id="contactList"></ul>

<script src="main.js"></script>

</body>

</html>

-------------------------------------------------

main.js

const API = "https://jsonplaceholder.typicode.com/users"; // Mock server

const form = document.getElementById("contactForm");

const list = document.getElementById("contactList");

// Initial fetch (load existing 10 dummy users)

async function fetchContacts() {

const res = await fetch(API);

const users = await res.json();

render(users);

}

// Handle form submission

form.addEventListener("submit", async (e) => {

e.preventDefault();

const data = Object.fromEntries(new FormData(form));

// Send POST request (fake save on JSONPlaceholder)

const res = await fetch(API, {

method: "POST",

headers: { "Content-Type": "application/json" },

body: JSON.stringify(data)

});

// Get the fake response (server returns data with an id)

const created = await res.json();

// ✅ Append new contact directly to the list

list.insertAdjacentHTML("beforeend", `<li>${created.name} — ${created.email}</li>`);

// Clear form

form.reset();

});

// Render function

function render(users) {

list.innerHTML = users.map(u => `<li>${u.name} — ${u.email}</li>`).join("");

}

**Expected Output:**

* User fills form and clicks Submit.
* Data is sent to the mock server via AJAX.
* Page displays all stored contacts below the form.

**Result:**  
The AJAX-based form was successfully created. Data submission and retrieval from the mock server were demonstrated using the fetch() API.