**Name : Sharvayu Zade**

**Sec: B(B2)**

**Prn: 23070521135**

**Practical No.: 10 JSON Handling and Mini API Mock**

**Aim:** To parse and display JSON data using both **fetch()** and **$.getJSON() (jQuery AJAX)** methods by loading data from a local JSON file or mock API, and dynamically render it in an HTML table.

**Requirements:**

**Software/Tools:**

● Web Browser (Chrome/Edge/Firefox)

● Code Editor (VS Code / Sublime / Notepad++)

● Internet connection (for jQuery CDN)

**Files:**

● index.html

● script.js

● data.json (mock JSON data file)

**Theory:**

**What is JSON?**

● **JSON (JavaScript Object Notation)** is a lightweight data-interchange format. ● It is easy for humans to read and write and easy for machines to parse and generate.

● Used for sending and receiving structured data between client and server.

**Example JSON:**

[

{ "id": 1, "name": "Alice", "marks": 85 },

{ "id": 2, "name": "Bob", "marks": 92 }

]

**Fetching JSON Data :**JavaScript can retrieve data from files or APIs using:

**Fetch API (modern method)**

fetch('data.json')

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

.then(data => console.log(data));

**jQuery AJAX ($.getJSON()) (traditional method)**

$.getJSON('data.json', function(data) {

console.log(data);

});

Both methods are used to load and display data dynamically without reloading the webpage.

**Steps:**

1. **Create the project folder** named Practical10\_JSON\_AJAX. 2. **Create the following files:**

○ index.html

○ script.js

○ data.json

3. **Include jQuery in your HTML file** using CDN.

4. **Write the HTML structure** to contain an empty table.

5. **Write JavaScript code** to:

○ Fetch JSON data using fetch().

○ Fetch JSON data using $.getJSON().

○ Display the results in a table dynamically.

6. **Run the HTML file** in your browser.

7. **Observe and compare** how both methods display the data.

**5. Example Code**

**data.json**

[

{ "id": 1, "name": "Alice", "marks": 85 },

{ "id": 2, "name": "Bob", "marks": 92 },

{ "id": 3, "name": "Charlie", "marks": 78 }

]

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>JSON Handling with Fetch and jQuery</title>

<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script> </head>

<body>

<h2>Student Records (Using Fetch and jQuery AJAX)</h2>

<h3>Data from Fetch()</h3>

<table id="fetchTable" border="1" cellpadding="5"></table>

<h3>Data from $.getJSON()</h3>

<table id="jqueryTable" border="1" cellpadding="5"></table>

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

</body>

</html>

**script.js**

// ---------- Fetch API Example ----------

fetch('data.json')

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

.then(data => {

let table = document.getElementById('fetchTable');

table.innerHTML = "<tr><th>ID</th><th>Name</th><th>Marks</th></tr>"; data.forEach(student => {

table.innerHTML += `<tr>

<td>${student.id}</td>

<td>${student.name}</td>

<td>${student.marks}</td>

</tr>`;

});

})

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

// ---------- jQuery AJAX Example ----------

$.getJSON('data.json', function(data) {

let table = $('#jqueryTable');

table.append("<tr><th>ID</th><th>Name</th><th>Marks</th></tr>"); $.each(data, function(index, student) {

table.append(`<tr>

<td>${student.id}</td>

<td>${student.name}</td>

<td>${student.marks}</td>

</tr>`);

});

}).fail(function() {

console.log("Error loading JSON data with jQuery.");

});

**Output:**

When the page is opened, two tables appear:

● One populated via **Fetch API**

● Another populated via **jQuery AJAX**

Both show the same student data from data.json.

**Sample Output:**

Student Records (Using Fetch and jQuery AJAX)

| **ID** | **Name** | **Marks** |
| --- | --- | --- |
| 1 | Alice | 85 |
| 2 | Bob | 92 |
| 3 | Charlie | 78 |

**Simple Task for Students:**

1. Add a new key "grade" (A/B/C) to each student in the JSON file and display it in the table.

2. Highlight rows where marks > 90 using a background color.

3. Modify the fetch call to simulate a delay using setTimeout() to observe asynchronous loading.