JAVASCRIPT FETCH

LAB 1

JavaScript's fetch() method is a powerful tool for working with web APIs. Here's a tutorial that demonstrates how to use the fetch() method to access a public API and retrieve data. We will use the JSONPlaceholder API (https://jsonplaceholder.typicode.com/) as an example.

1) Create an HTML file called "fetch.html" and add the following code:

```html
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Fetch</title>
</head>
<body>
    <button id="get-data-button">Get Data</button>
    <div id="data-container"></div>
    <script src="fetch.js"></script>
</body>
</html>
```

This creates a simple HTML page with a button that says "Get Data!" and a div element that will be used to display the retrieved data. We also include a link to a JavaScript file called "fetch.js" which will contain our JavaScript code.

2) Create a file called "fetch.js" in the same directory as "fetch.html" and add the following code:

```javascript
const getDataButton = document.getElementById("get-data-button");
const dataContainer = document.getElementById("data-container");

getDataButton.addEventListener("click", () => {
    fetch('https://jsonplaceholder.typicode.com/posts')
    .then(response => response.json())
    .then(data => {
        data.forEach( post => {
            let postData = document.createElement('div');
            postData.innerHTML = `<h2>${post.title}</h2> <p>${post.body}</p>`;
            dataContainer.appendChild(postData);
        });
    })
```

JAVASCRIPT FETCH

```
    .catch(err => {
        console.log(err);
    });
});
```

This code uses the getElementById() method to select the getDataButton and dataContainer elements. Then, it uses the addEventListener() method and arrow function to attach a "click" event listener to the button. When the button is clicked, the anonymous function passed to addEventListener() is executed, which will call the fetch() method to retrieve data from the JSONPlaceholder API. The fetch() method returns a promise, so we use the .then() method to handle the response and retrieve the json data. It then loops through the posts and creates new div element for each post which is then added to dataContainer.

3) Open "fetch.html" in a web browser.

4) Click the button "Get Data" and you should see the list of posts along with their titles and bodies being displayed on the page

This example demonstrates a basic use of fetch() method, you can also pass additional options such as headers, and also use different HTTP methods like POST, PUT, DELETE using fetch(). you can also use await instead of .then() to make the code more readable. Also JSONPlaceholder API has many more endpoints like /comments, /albums etc you can use as per your requirement.

LAB 2

Using the OpenWeatherMap API (https://openweathermap.org/api), create a JavaScript program that retrieves the current weather data for a given city and displays the information on a web page. You will need to obtain an API key from the site and read the API documentation for this to work correctly.

• Use the fetch() method to retrieve the data from the API.

• The information should be displayed in a format of your choice, with the temperature, humidity and wind speed being displayed on the page

• You should have a button that user can enter a city name and on button click it makes the fetch call and displays the weather information

Deliverable:

CODE EXERCISE
JAVASCRIPT FETCH

- A single HTML file called weather.html

- A single JavaScript file called weather.js