Day 6 → JavaScript APIs and AJAX:

JavaScript APIs and AJAX:
JavaScript APIs (Application Programming Interfaces) allow you to interact with external services, libraries, and frameworks to extend the functionality of your web applications. AJAX (Asynchronous JavaScript and XML) is a technique that allows you to update parts of a webpage without reloading the entire page.

1. Fetch API:
The Fetch API is a modern JavaScript interface for making asynchronous HTTP requests. It provides a more powerful and flexible way to make API requests compared to older techniques like XMLHttpRequest. The Fetch API returns a Promise that resolves to the response from the server.

Example of Fetch API:

```javascript
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => {
    // Process the retrieved data
    console.log(data);
  })
  .catch(error => {
    console.error('Error:', error);
  });
```

In this example, we use the Fetch API to make a GET request to the URL 'https://api.example.com/data'. The response is then parsed as JSON using the `response.json()` method, and the resulting data is logged to the console.

2. Working with JSON data:
JSON (JavaScript Object Notation) is a lightweight data interchange format. It is commonly used for transferring data between a server and a

web application. JavaScript provides methods to parse JSON strings into JavaScript objects and stringify JavaScript objects into JSON strings.

Parsing JSON:

```javascript
const jsonString = '{"name": "John", "age": 30}';
const obj = JSON.parse(jsonString);
console.log(obj.name); // Output: John
```

Stringifying JavaScript Object:

```javascript
const obj = { name: "John", age: 30 };
const jsonString = JSON.stringify(obj);
console.log(jsonString); // Output: {"name":"John","age":30}
```

3. Implementing AJAX:

AJAX allows you to load and send data asynchronously without requiring a page reload. It enables you to update specific parts of a webpage dynamically.

Example of AJAX using Fetch API:

```javascript
function fetchData() {
  fetch('https://api.example.com/data')
    .then(response => response.json())
    .then(data => {
      // Update the webpage with the retrieved data
      document.getElementById('result').innerHTML = data.message;
    })
    .catch(error => {
      console.error('Error:', error);
    });
}
```

In this example, we define a `fetchData` function that uses the Fetch API to make a GET request to an API endpoint. The retrieved data is then used to update the content of an HTML element with the id 'result'.

Hands-on Project:

```
<!DOCTYPE html>
<html>

<head>
    <title>Weather App</title>
    <style>
        .loader {
            display: none;
            position: fixed;
            top: 50%;
            left: 50%;
            transform: translate(-50%, -50%);
            font-size: 20px;
            border: 8px solid #f3f3f3;
            border-top: 8px solid #3498db;
            border-radius: 50%;
            width: 60px;
            height: 60px;
            animation: spin 2s linear infinite;
        }

        @keyframes spin {
            0% {
                transform: rotate(0deg);
            }

            100% {
                transform: rotate(360deg);
            }
        }

        .popup-bg {
            display: none;
            position: fixed;
```

```
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            background-color: rgba(230, 19, 19, 0.5);
            z-index: 999;
        }

        .popup {
            display: none;
            position: fixed;
            top: 50%;
            left: 50%;
            transform: translate(-50%, -50%);
            padding: 20px;
            background-color: white;
            border: 1px solid black;
            box-shadow: 0 2px 8px rgba(0, 0, 0, 0.15);
            font-size: 18px;
            z-index: 1000;
        }

        .close-icon {
            position: absolute;
            top: 10px;
            right: 10px;
            font-size: 24px;
            cursor: pointer;
        }
    </style>
</head>

<body>
    <h1>Weather App</h1>

    <input type="text" id="locationInput" placeholder="Enter location">
    <button onclick="getWeatherData()">Get Weather</button>

    <div id="loader" class="loader"></div>
```

```
<div id="popupBg" class="popup-bg"></div>

<div id="popup" class="popup">
    <span class="close-icon" onclick="closePopup()">&times;</span>
    <h2 id="locationName"></h2>
    <p id="temperature"></p>
    <p id="humidity"></p>
</div>

<script>
    function showLoader() {
        const loader = document.getElementById('loader');
        loader.style.display = 'block';
    }

    function hideLoader() {
        const loader = document.getElementById('loader');
        loader.style.display = 'none';
    }

    function showWeatherPopup(location, temperature, humidity) {
        const popupBg = document.getElementById('popupBg');
        const popup = document.getElementById('popup');
        const locationName = document.getElementById('locationName');
        const temp = document.getElementById('temperature');
        const humid = document.getElementById('humidity');

        locationName.textContent = location;
        temp.textContent = `Temperature: ${temperature}°C`;
        humid.textContent = `Humidity: ${humidity}%`;

        popupBg.style.display = 'block';
        popup.style.display = 'block';
    }

    function closePopup() {
        const popupBg = document.getElementById('popupBg');
        const popup = document.getElementById('popup');

        popupBg.style.display = 'none';
```

```
            popup.style.display = 'none';
        }

        function getWeatherData() {
            const locationInput =
document.getElementById('locationInput');
            const location = locationInput.value;

            const apiKey = 'c7a9299a3d8da1d910da08bcffb48a3b';
            const apiUrl =
`http://api.openweathermap.org/data/2.5/weather?q=${location}&appid=${apiK
ey}`;

            showLoader();

            setTimeout(() => {
                fetch(apiUrl)
                .then(response => response.json())
                .then(data => {
                    hideLoader();
                    const temperature = Math.round(data.main.temp -
273.15);
                    const humidity = data.main.humidity;
                    showWeatherPopup(location, temperature, humidity);
                })
                .catch(error => {
                    hideLoader();
                    console.log('An error occurred:', error);
                });
            }, 3000);
        }

        window.addEventListener('click', function (event) {
            const popupBg = document.getElementById('popupBg');
            const popup = document.getElementById('popup');

            if (event.target === popupBg) {
                closePopup();
            }
        });
```

```
    </script>
</body>

</html>
```