**PRACTICAL: 2**

**AIM:**

Create a simple weather application that displays a hardcoded temperature for a given city. This simple weather application demonstrates basic HTML structure for user input, CSS styling for layout and appearance, and JavaScript functionality to handle user interactions and display dynamic content. It provides a foundational example of building an interactive web application using essential front-end technologies.

Technologies Used: HTML, CSS, JavaScript (ES6)

Notes:

* weatherData: This object holds hardcoded weather information for specific cities. In a real-world scenario, this data would typically come from an API.
* addEventListener: Listens for a click event on the Get Weather button.
* Fetching Weather: When the button is clicked, it retrieves the city entered by the user, checks if weather data exists in weatherData, and displays the corresponding weather information or a message if the city is not found.

**THEORY:**

This simple weather application displays hardcoded temperature data for specific cities using basic web technologies. It demonstrates how user input can be processed and dynamically rendered using JavaScript. HTML is used for the layout and input fields, CSS styles the UI, and JavaScript handles logic and interaction. The weather data is stored in a static object for demonstration. This project builds foundational understanding of interactive front-end development.

**CODE:**

|  |
| --- |
| **HTML:**  <!DOCTYPE *html*>  <html *lang***=**"en">  <head>    <meta *charset***=**"UTF-8">    <title>Weather App</title>    <link *rel***=**"stylesheet" *href***=**"style.css">  </head>  <body>    <div *class***=**"container">      <h1>Weather App</h1>      <input *type***=**"text" ***id*=**"cityInput" *placeholder***=**"Ahmedabad">      <button ***id*=**"getWeatherBtn">Click here to Get Weather</button>      <div ***id*=**"weatherResult"></div>    </div>    <script *src***=**"script.js"></script>  </body>  </html>  **CSS:**  body {    margin**:** 0**;**    padding**:** 0**;**    background-color**:** **#c0392b;** */\* Red background \*/*    font-family**:** Arial**,** sans-serif**;**    height**:** 100vh**;**    display**:** flex**;**    justify-content**:** center**;**    align-items**:** center**;**  }  *.container* {    text-align**:** center**;**  }  h1 {    color**:** black**;**    font-weight**:** bold**;**  }  input {    display**:** block**;**    margin**:** 10px auto**;**    padding**:** 10px**;**    width**:** 250px**;**    font-size**:** 16px**;**  }  button {    display**:** block**;**    margin**:** 10px auto**;**    padding**:** 10px**;**    width**:** 250px**;**    font-size**:** 16px**;**    background-color**:** white**;**    color**:** black**;**    border**:** none**;**    cursor**:** pointer**;**  }  *#weatherResult* {    margin-top**:** 15px**;**    color**:** black**;**    font-size**:** 16px**;**  }  **JS:**  const weatherData **=** {    "Ahmedabad"**:** { temperature: "40°C" },    "Delhi"**:** { temperature**:** "38°C" }**,**    "Mumbai"**:** { temperature**:** "34°C" }**,**    "Bangalore"**:** { temperature**:** "30°C" }  }**;**  document**.**getElementById("getWeatherBtn")**.**addEventListener("click"**,** () **=>** {    const city **=** document**.**getElementById("cityInput")**.***value***.**trim()**;**    const resultDiv **=** document**.**getElementById("weatherResult")**;**    if (weatherData[city]) {      resultDiv.textContent = `The weather in ${city} is ${weatherData[city].temperature}`;    } else {      resultDiv.textContent = "Weather data not available for the entered city.";    }  });  // document.getElementById("getWeatherBtn").addEventListener("click", () => {  //     const city = document.getElementById("cityInput").value.trim();  //     const resultDiv = document.getElementById("weatherResult");    //     fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=YOUR\_API\_KEY&units=metric`)  //         .then(response => {  //         if (!response.ok) {  //             throw new Error("City not found");  //         }  //         return response.json();  //         })  //         .then(data => {  //         const temperature = data.main.temp;  //         resultDiv.textContent = `The weather in ${city} is ${temperature}°C`;  //         })  //         .catch(error => {  //         resultDiv.textContent = error.message;  //         });  //     }); |

**OUTPUT:**

****

**LATEST APPLICATIONS:**

* **Real-Time Weather Apps** – Platforms like AccuWeather and Weather.com use APIs to show live data with responsive design.
* **Smart Home Integration** – Weather modules in smart home systems display local forecasts on smart displays.
* **Mobile Weather Widgets** – Android and iOS use lightweight weather apps/widgets for real-time updates.
* **Flight & Travel Apps** – Apps like Google Travel show weather info for destinations in planning tools.
* **Agricultural Advisory Apps** – Farmers use weather apps to plan irrigation and harvesting.

**LEARNING OUTCOME:**

* Learned how to capture and process user input using JavaScript.
* Understood basic DOM manipulation and event handling (addEventListener).
* Gained experience using conditional logic to display relevant data.
* Applied basic styling using CSS for a better UI experience.
* Built a functional mini-project combining HTML, CSS, and JS.

**REFERENCES:**

1. W3Schools – JavaScript Tutorial – <https://www.w3schools.com/js/>
2. MDN Web Docs – addEventListener() – <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>
3. CSS Layout Techniques – <https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout>