**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.

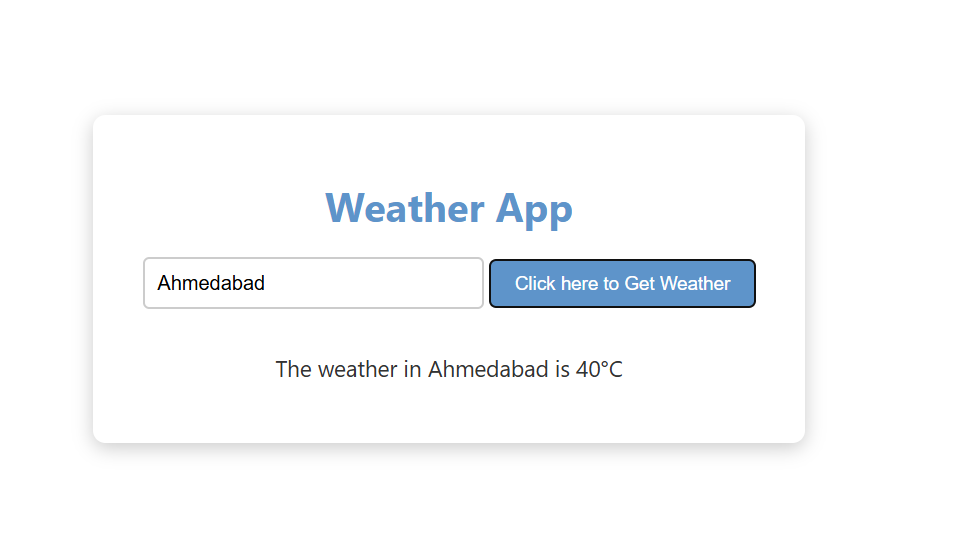
**THEORY:**

A simple weather application is a beginner-level web project that uses HTML, CSS, and JavaScript. HTML creates the structure with an input box and button, CSS styles the layout to make it visually appealing, and JavaScript handles user interaction by displaying a hardcoded temperature based on the entered city name. This project helps in understanding how basic front-end technologies work together to create interactive web pages.

**CODE:**

|  |
| --- |
| **HTML CODE :**  **CSS CODE:**  body {  margin: auto;  font-family: 'Segoe UI', sans-serif;  background-image: url(Weather\_3.jpg);  /\* background-size: contain; \*/  background-repeat: no-repeat;  background-position: center;  justify-content: center;  align-items: center;  height: 100vh;  display: flex;  justify-content: center;  align-items: center;  background-size: cover;    }  .weather-container {  background: white;  padding: 30px 40px;  border-radius: 10px;  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);  text-align: center;  }  .weather-container h1 {  margin-bottom: 20px;  color: #5e94ca;  }  input[type="text"] {  width: 250px;  padding: 10px;  border: 2px solid #ccc;  border-radius: 5px;  margin-bottom: 15px;  font-size: 16px;  }  button {  padding: 10px 20px;  background-color: #5e94ca;  color: #fff;  border: none;  border-radius: 5px;  font-size: 15px;  cursor: pointer;  transition: background-color 0.3s;  }  button:hover {  background-color:#0da0ea;  }  #weatherResult {  margin-top: 20px;  font-size: 18px;  color: #333;  }  **JAVASCRIPT CODE:**  const weatherData = {  Ahmedabad: "40°C",  Delhi: "38°C",  Mumbai: "35°C",  Jaipur: "42°C",  Bangalore: "30°C",  Surat: "37°C",  Rajkot: "39°C"  };  document.getElementById("getWeatherBtn").addEventListener("click", () => {  const city = document.getElementById("cityInput").value.trim();  const result = document.getElementById("weatherResult");  if (weatherData[city]) {  result.textContent = `The weather in ${city} is ${weatherData[city]}`;  } else {  result.textContent = `Sorry, weather data for "${city}" is not available.`;  }  }); |

**OUTPUT :**

****

When a user enters a known city like "Ahmedabad" and clicks the button, the application shows a message such as "The current temperature in Ahmedabad is 32°C." If the city is not in the predefined list, it shows a message like "City not found." This output behavior demonstrates dynamic content update and basic condition checking using JavaScript.

**LATEST APPLICATIONS:**

* AI Forecasting – Accurate predictions using AI & ML.
* Real-Time Alerts – Instant weather warnings.
* Wearable Integration – Weather on smartwatches.
* AR Visuals – Augmented Reality weather views.
* Air Quality Index – Shows AQI and health alerts.
* Voice Assistant Support – Works with Siri, Alexa, Google.
* Travel Planning – Weather-based trip suggestions.
* Smart Home Control – Automates devices based on weather.
* User Reports – Crowdsourced weather updates.
* Live Radar Maps – Minute-level local forecasts.

**LEARNING OUTCOME:**

* Understood how to build interactive web applications using HTML, CSS, and JavaScript.
* Learned to update DOM elements dynamically based on user actions.
* Gained hands-on experience with JavaScript functions, events, and objects.
* Simulated real-time updates using setInterval to mimic live voting.
* Improved UI/UX design through basic CSS styling and layout techniques.

**REFERENCES:**

* W3Schools – JavaScript Introduction
* [MDN Web Docs – JavaScript DOM Manipulation](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model)
* CSS Basics – W3Schools
* [JavaScript setInterval() – MDN](https://developer.mozilla.org/en-US/docs/Web/API/setInterval)
* HTML Buttons – W3Schools
* How to Create a Voting App – GeeksforGeeks