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)

THEORY:

1. HTML (HyperText Markup Language)

HTML creates the basic structure of the weather app. It includes:

- An input field where the user types a city name
- A button to check the weather
- A display area where the hardcoded temperature is shown

These elements are organized using <input>, <button>, and <div> tags.

2. CSS (Cascading Style Sheets)

CSS styles the application to make it clean and appealing.
It is used to:

- Center elements on the screen
- Set background colors, font styles, and spacing
- Improve user experience with a neat layout and readable design

3. JavaScript (ES6)

JavaScript makes the app interactive. It:

- Listens for the button click (event handling)
- Reads the city name from the input field
- Displays a hardcoded temperature in the output section (DOM manipulation)
- Uses a function like showWeather() to handle logic
- May use conditionals to show different temperatures for different cities (optional)

CODE:

//index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <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="Enter City Name" />
    <button id="getWeatherBtn">Click here to Get Weather</button>
    <p id="weatherResult"></p>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

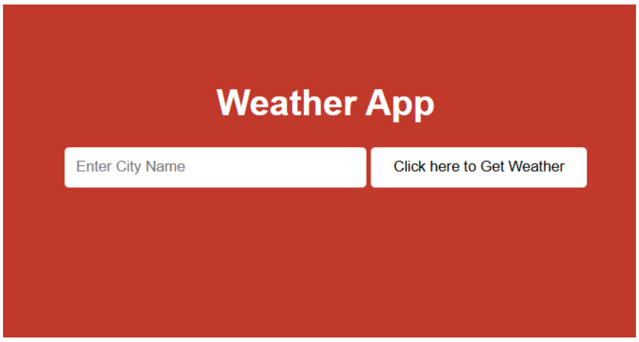
//script.js

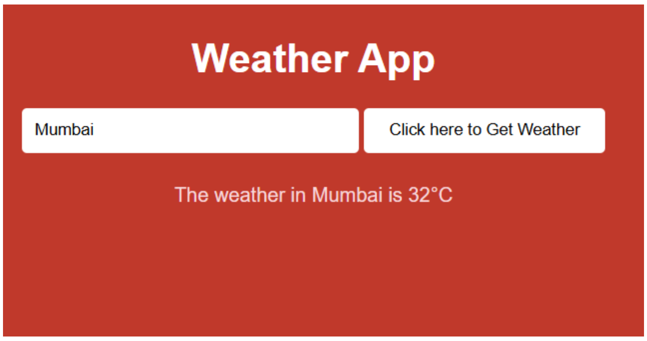
```
// Hardcoded weather data
const weatherData = {
  Ahmedabad: '40°C',
  Mumbai: '32°C',
  Delhi: '38°C',
  Bengaluru: '28°C',
  Chennai: '34°C'
};

// Event listener for the button
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 ${weatherData[city]}`;
  } else {
    result.textContent = `Sorry, no weather data available for "${city}"`;
  }
});
```

//style.css

```
body {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
  background-color: #c0392b;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  background-color: #c0392b;
  text-align: center;
  padding: 40px;
  border-radius: 8px;
  color: white;
}

input[type="text"] {
  padding: 10px;
  width: 250px;
  margin-bottom: 10px;
  border: none;
  border-radius: 4px;
}

button {
  padding: 10px 20px;
  border: none;
  background-color: white;
  color: black;
  cursor: pointer;
  border-radius: 4px;
}

#weatherResult {
  margin-top: 15px;
  color: #f8d7da;
}
```

OUTPUT:





Mumbai

The weather in Mumbai is 32°C

LATEST APPLICATIONS:

In todays time html css and javascript are great stepping stones to get into the world of learning web development, a lot of simple websites use this tech stack to reduce complexity and cost of hostings.

LEARNING OUTCOME:

By performing this practical II got a thorough understanding of basic web technologies and terms. And how the frontend and backend connect in a real website. I also learnt about form inputs and outputs

REFERENCES:

1. Html: https://www.freecodecamp.org/learn/2022/responsive-web-design/learn-html-by-building-a-cat-photo-app/step-66
2. Css: https://www.w3schools.com/css/