COLLEGE CODE : 9111

COLLEGE NAME: SRM MADURAI COLLEGE FOR ENGINEERING AND TECHNOLOGY

DEPARTMENT: B.E COMPUTER SCIENCE AND ENGINEERING

STUDENT NM-ID:

3DE3OEEADOADD16ABF843D795F7463AA
DE16A18153AEB4296F4DC6FC9A42259
31D7A43E9953565998B4B4D75D69E588
710F5CAB0746CA2B6104D9A9EF3C4CBC

ROLLNO:911123104007
911123104027
911123104037
911123104038

DATE:

Completed the project named as Phase 3

TECHNOLOGYPROJECT NAME : Live Weather Dashboard

SUBMITTED BY,NAME:
D.Devis Akalya Pushpam
T.D.B.Kiruthikha
R.S.Priyadharshini
S.B.Priyadharshini

LIVE WEATHER DASHBOARD – PHASE III

1. Project Setup

- Initialize theproject: Use a frontend framework like React (or Angular / Vue / Flutter).
- Set up project structure with folders for components, styles, and utilities.
- Install dependencies: React, Axios (for API calls), Chart.js / Recharts (for graphs), TailwindCSS
- / Bootstrap (optional).
- Set up version control: Initialize a Git repository. Create .gitignore file to exclude node_modules, build files, etc.
  Basic file scaffolding: Create main files App.js, index.js. Create UI components folder: e.g., WeatherCard.js, ForecastChart.js.

2. Core Features Implementation

- CurrentWeather Display: Fetchanddisplay temperature, humidity, wind speed, and condition.
- Show corresponding weather icon.
- Location-based Weather: Use Geolocation API to fetch weather for the user's current location.
- Forecast: Display next 5-day / hourly forecast with charts (line or bar).
  Search Functionality: Allow users to search weather by city name. Show error message if city not found.

3. Data Storage (Local State / Database)

- LocalState Management: Use React state / ContextAPI / Redux for managing weather data.
- Persistence (optional for MVP): Save user's last searched city in localStorage for reloads.
- Database Integration (for later versions): Use Firebase / MongoDB for storing user preferences. Enable real-time updates if needed.

4. Testing Core Features

- UnitTests: TestAPIintegrationworks (mock API calls). Verify weather data renders correctly in components.
- Integration Tests: Ensure searching a city updates weather data on the dashboard. Check forecast charts render correctly.
- Manual Testing: Test responsiveness across screen sizes. Verify invalid city names show an error.
- Automated UI tests (optional): Use Cypress / Selenium for automated end-to-end testing.

5. Version Control (GitHub)

- Initialize Git Repository: git init at project root.
- Commit frequently: After project setup, core features completion, and testing features.
- Branching Strategy: Use feature branches like feature/current-weather, feature/forecast-chart.
- Push to remote: Create repository on GitHub. git remote add origin , git push -u origin main.
- Pull Requests: Use PRs to review and merge features.
- Documentation: Maintain README.md with setup and usage instructions. Optionally include roadmap.

Coding

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 (HTML Only)</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script src="https://momentjs.com/downloads/moment.js"></script>
  <script src="script.js" defer></script>
</head>
<body>
  <div class="container">
    <h1>GeeksforGeeks</h1>
    <h3>Weather App</h3>

    <!-- Input -->
    <input id="city-input" type="text" placeholder="Enter city name">
    <button id="city-input-btn">Get Weather</button>

    <!-- Weather Info -->
    <div class="weather-card">
      <h3 id="city-name">City Name</h3>
      <p id="date">Date</p>
      <img id="weather-icon" src="" alt="Weather Icon">
      <p id="temperature">Temperature</p>
      <p id="description">Description</p>
      <p id="wind-speed">Wind Speed</p>
    </div>
  </div>
</body>
</html>
```

style.css (**Design**)

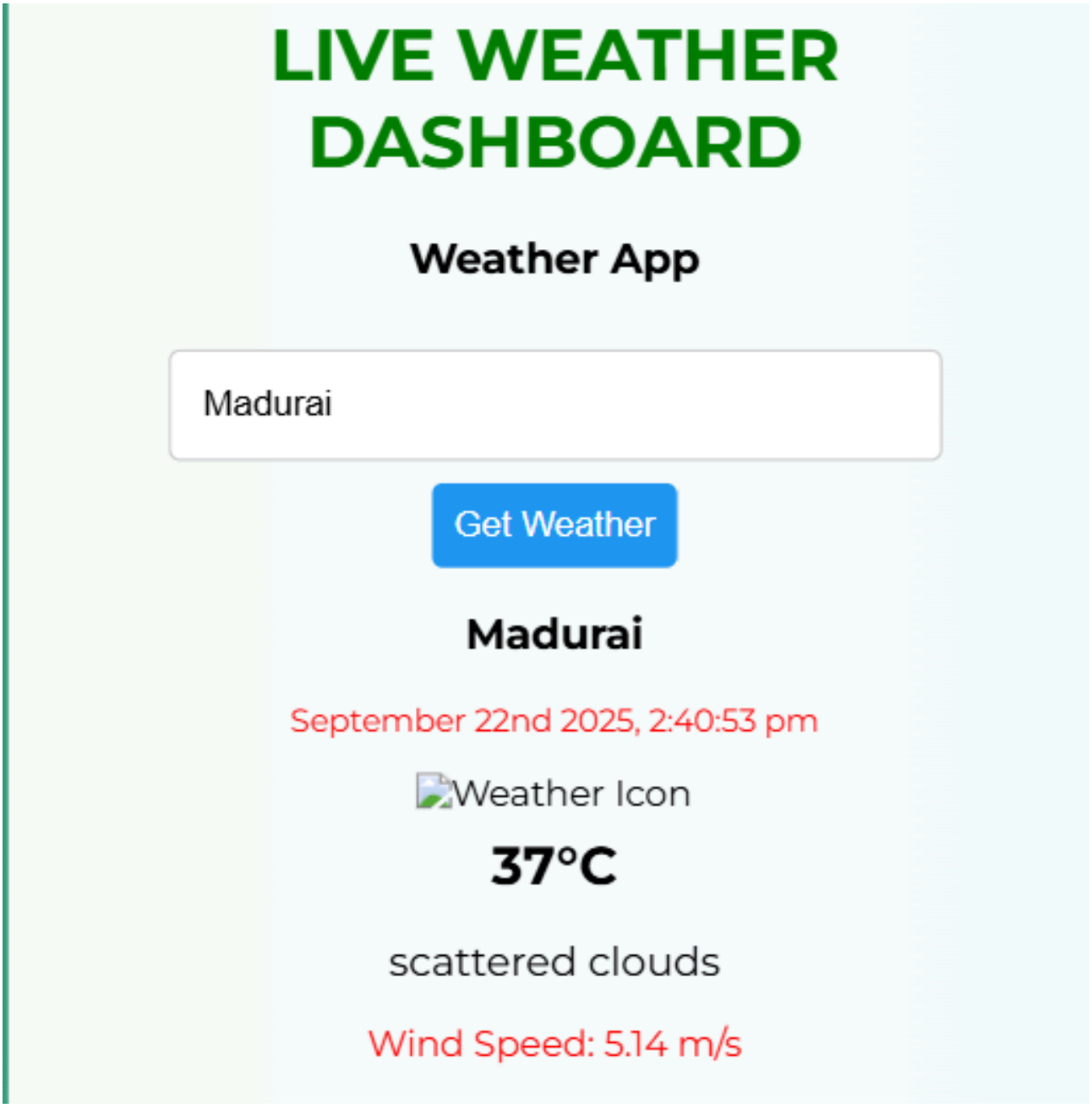
```
body {
    font-family: 'Montserrat', sans-serif;
    background: linear-gradient(to right, #4facfe, #00f2fe);
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
```

```
    margin: 0;
}

.container {
    text-align: center;
}

.weather-card {
    background: #fff;
    padding: 20px;
    border-radius: 15px;
    box-shadow: 0px 4px 15px rgba(0, 0, 0, 0.2);
    width: 320px;
    transition: transform 0.3s ease-in-out;
}

.weather-card:hover {
    transform: scale(1.05);
}

#city-input {
    padding: 10px;
    border-radius: 8px;
    border: 1px solid #ccc;
    margin: 10px 0;
    width: 80%;
}

#city-input-btn {
    background: green;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 8px;
    cursor: pointer;
    margin-top: 5px;
}

#city-input-btn:hover {
    background: darkgreen;
}

#weather-icon {
    width: 100px;
    margin: 10px auto;
}
```

script.js (**Functionality**)

```
const apiKey = "YOUR_API_KEY"; // 🔑 Replace with your OpenWeatherMap API Key

function weatherFn(city) {
    if (!city) {
        alert("Please enter a city name!");
        return;
    }

    const url =
`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric`;

    $.get(url, function(data) {
        $("#city-name").text(data.name + ", " + data.sys.country);
        $("#date").text(moment().format("MMMM Do YYYY, h:mm a"));
        $("#temperature").text(`🌡️ Temperature: ${data.main.temp}°C`);
        $("#description").text(`☁️ Condition: ${data.weather[0].description}`);
        $("#wind-speed").text(`💨 Wind Speed: ${data.wind.speed} m/s`);
        $("#weather-icon").attr("src",
`https://openweathermap.org/img/wn/${data.weather[0].icon}@2x.png`);
    }).fail(function() {
        alert("City not found. Please try again!");
    });
}

// Event listener
$("#city-input-btn").click(() => {
    const city = $("#city-input").val();
    weatherFn(city);
});
```

Output


LIVE WEATHER DASHBOARD
Weather App
Madurai
Get Weather
Madurai
September 22nd 2025, 2:40:53 pm
Weather Icon
37°C
scattered clouds
Wind Speed: 5.14 m/s