

NORAIZ RANA

Task #4

Topic

Weather App using
HTML, CSS & JavaScript

Date:

1 July 2025

SUBMITTED TO:

APPVERSE TECHNOLOGIES



Responsive Weather Application using JavaScript ES6+

☰ Objective:

The goal of this project is to build a responsive and interactive weather web application using HTML, CSS, and modern JavaScript (ES6+). The app fetches real-time weather data from a public API and provides a dynamic user experience with theme toggling, location-based weather, and search history.

⚙️ Technologies Used:

- **HTML5** – for structure and semantic layout
 - **CSS3** – for styling, responsiveness, and animations
 - **JavaScript ES6+** – for logic and interactivity (using features like `let`, `const`, arrow functions, destructuring, and template literals)
 - **OpenWeatherMap API** – to retrieve real-time weather data
 - **Geolocation API** – to fetch weather based on the user's location
 - **Unsplash Source API** – for dynamic weather-specific background images
-

🎯 Key Features:

1. City-Based Weather Search

Users can enter any city name and receive current temperature, condition, and an animated weather icon.

2. Geolocation Support

One-click access to current weather based on the user's actual location using the browser's built-in geolocation API.

3. Dark/Light Theme Toggle

A simple toggle switch allows users to change between light and dark mode dynamically.

4. Animated Icons & Weather Backgrounds

Weather icons float gently using CSS animation, and the background image updates to match the current weather (e.g., clear, clouds, rain, snow).

5. Search History

Previously searched cities are saved and displayed as clickable items for easy re-checking.

6. Responsive Design

The app is mobile-friendly and adjusts its layout based on screen size for a better user experience.

ES6+ Concepts Demonstrated:

- `const` and `let` for proper scope and variable control
 - Arrow functions for concise callbacks
 - Template literals for building dynamic strings
 - Object destructuring for cleaner access to data
 - Asynchronous `fetch` handling with `.then()` chaining
 - DOM manipulation for real-time content updates
-

Project Structure:

- `index.html` – markup structure
 - `style.css` – custom styles and animations
 - `script.js` – all JavaScript logic and API integration
-

The project successfully demonstrates how to build a functional and visually appealing web application using real-time data and modern web technologies. It integrates external APIs, uses modern JS features, and enhances user experience through interactivity, animations, and responsive design.