**Front-End UI/UX Mini Project**

**TITLE PAGE**

* **Project Title:** Weather DashBoard
* **Submitted By:**
  + **Team Members:** Anshika Sharon Sonwani, Freida B. Rodrigues, Evan Devanand P A
  + **Roll Numbers: 2460334, 2460367, 2460365**
  + **College Email ID:**
  + **[anshika.sharon@btech.christuniversity.in](mailto:anshika.sharon@btech.christuniversity.in  )**

[**freida.b@btech.christuniversity.in**](mailto:freida.b@btech.christuniversity.in)

[**evan.devanand@btech.christuniversity.in**](mailto:evan.devanand@btech.christuniversity.in)

* **Course:** *UI/UX Design Fundamentals*
* **Instructor Name:** Mr. Narendra
* **Institution:** *Christ University*
* **Date of Submission:** 26/09/2025

**Abstract**

This project involves designing and developing a responsive weather dashboard that allows users to input a location and view **current weather conditions** along with a **5-day forecast**. The key goal is to provide a clean, intuitive, and interactive interface that fetches real-time weather data from an external API (OpenWeatherMap) and presents it effectively. Built using HTML, CSS, JavaScript, Bootstrap, and jQuery, the dashboard emphasizes responsive design, usability, and accessibility across multiple devices. The interface includes sections for current weather details, such as temperature, humidity, and general conditions, and a 5-day forecast with visual indicators. The final outcome is a functional, visually appealing front-end application that demonstrates proficiency in API integration, DOM manipulation, and responsive design principles.

**Objectives**

* Understand how to interact with external APIs to fetch real-time data.
* Design a clean, modern, and visually appealing interface for weather information.
* Display **current weather** details including temperature, humidity, and weather conditions.
* Implement a **5-day weather forecast** with day-wise temperature and conditions.
* Ensure **full responsiveness** using Bootstrap for desktops, tablets, and mobile devices.
* Use JavaScript and jQuery for dynamic content rendering and API interaction.
* Enhance user experience with intuitive layout, clear icons, and accessibility considerations.

**Scope of the Project**

* Front-end development using HTML, CSS, JavaScript, and Bootstrap.
* Integrates OpenWeatherMap API to fetch real-time weather data.
* Provides **current weather** and **5-day forecast** for any user-specified location.
* Fully responsive design suitable for desktop, tablet, and mobile devices

**Tools & Technologies Used**

* **HTML5** – Content structure and semantic layout
* **CSS3** – Styling, responsive layout, and visual enhancements
* **JavaScript** – Dynamic content rendering and API integration
* **jQuery** – DOM manipulation and event handling
* **Bootstrap** – Responsive grid system and UI components
* **OpenWeatherMap API** – Real-time weather data
* **VS Code** – Code editing

**HTML, CSS, and JS Structure Overview**

**HTML Structure**

* **Landing Page (index.html)**
  + Contains a **search form** for users to input the city.
  + Header with logo and brand name.
  + Main section with search input and submit button.
  + Linked to style.css for styling and script.js for interactivity.
* **Dashboard Page (dashboard.html)**
  + Header with logo, brand name, search input, and unit selector (°C, °F, K).
  + Main section divided into:
    - **Current Weather Card**: Displays date, time, temperature, weather condition, humidity, location, and weather icon.
    - **Weather Info Section**: Includes description/advisory text, a forecast button, and a **5-day forecast grid**.
  + Uses Bootstrap grid classes (row, col-\*) for responsive layout.
  + Weather icons and forecast cards dynamically populated using JavaScript.
  + Linked to style.css and script.js.
* **Semantic Elements Used**
  + <header>: For site branding and navigation.
  + <main>: Wraps all main content.
  + <section>: Separates current weather and forecast areas.
  + <div>: Containers for cards, grid layout, and dynamic content.
  + <form>: Search input and submit functionality.

**CSS Structure (style.css)**

* **Global Styles**
  + Resets for body and html, setting full height and font-family.
  + Base styles for background images on landing and dashboard pages.
* **Layout**
  + .dashboard-page and .index-page with background images.
  + .content-wrapper to wrap the main dashboard content.
  + Bootstrap container and grid classes for responsive layout.
* **Header**
  + Flexbox used to align logo, brand, search input, and unit selector.
  + Responsive adjustments to hide brand on smaller screens.
* **Search Form**
  + Rounded input field with shadow effect.
  + Button styling with hover effect.
* **Weather Card**
  + .weather-card: Rounded card with shadow, padding, and column layout.
  + .weather-card-header: Flexbox to align date and time.
  + .weather-main: Contains weather icon, temperature, location, and condition text.
  + Forecast cards with hover scaling effects for interactivity.
* **Responsive Design**
  + Media queries for tablet (max-width: 991.98px) and mobile (max-width: 767.98px & 575.98px) to adjust padding, font sizes, and layout.

**JavaScript Structure (script.js)**

* **API Integration**
  + Uses OpenWeatherMap API (fetch) to retrieve current weather and 5-day forecast.
  + Supports units: Metric (°C), Imperial (°F), Standard (K).
* **DOM Manipulation**
  + Elements cached in els object for efficient access.
  + Dynamic rendering of:
    - Current weather (renderCurrent)
    - Forecast cards (renderForecast)
  + Weather icon and background images updated based on conditions.
* **Utility Functions**
  + formatDateTimeIST for displaying date/time in Indian Standard Time.
  + unitSymbol for temperature units.
  + groupForecastDailyIST to group forecast by day.
* **Event Handling**
  + Search input: Enter key triggers weather fetch.
  + Unit selector: Updates temperature units and reloads weather.
  + Forecast button: Scrolls to forecast grid using jQuery animation.
* **Error Handling**
  + Displays user-friendly messages if city not found or API fails.
  + Defaults to fallback icons and backgrounds if condition not mapped.
* **Initialization**
  + Checks if on dashboard page and loads weather if city provided in query string.
  + Preloads images for smooth background transitions.

**CSS Styling Strategy**

* Bootstrap grid for responsive layout
* Flexbox for content alignment and distribution
* Media queries for adaptive design on mobile, tablet, and desktop
* Clear typography and contrasting colors for readability
* Weather icons and hover effects for enhanced user engagement

**Key Features**

* Input-based search to fetch weather for any location.
* Real-time display of **current temperature, humidity, and conditions.**
* 5-day weather forecast with day-wise temperature and condition icons.
* Responsive design that adapts to all device screen sizes.
* Clean, modern interface with clear visual hierarchy.
* Gives Tips according to the Weather Condition.

**Challenges Faced & Solutions**

* **Challenge:** Fetching real-time data and handling asynchronous API calls.  
  **Solution:** Used JavaScript fetch with proper error handling and promise chaining.
* **Challenge:** Displaying weather forecast dynamically based on API response.  
  **Solution:** Manipulated the DOM using jQuery to render forecast cards programmatically.
* **Challenge:** Ensuring full responsiveness across multiple devices.  
  **Solution:** Utilized Bootstrap grid, Flexbox, and media queries for adaptive layouts.
* **Challenge:** Handling invalid or misspelled location input.  
  **Solution:** Added validation and user-friendly error messages when API returns errors.

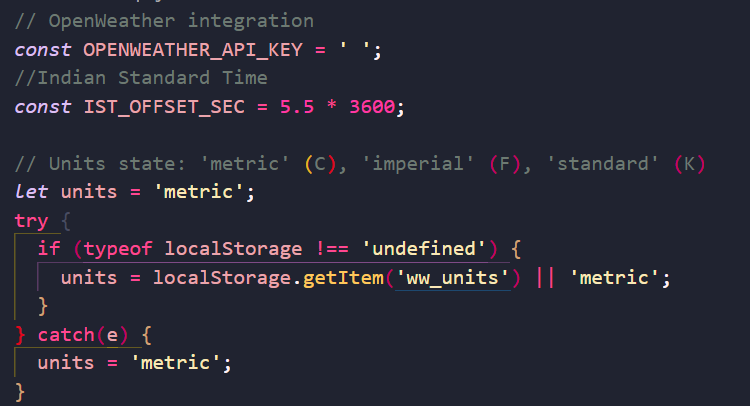
**Outcome**

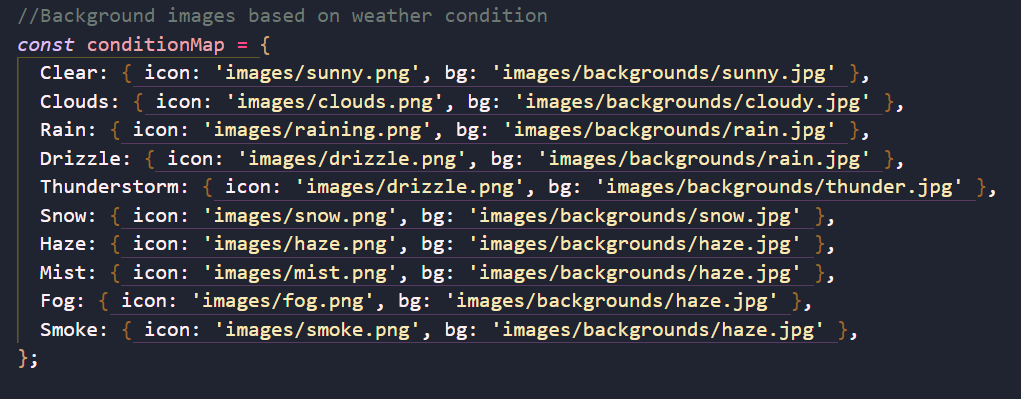
The project resulted in a fully functional, responsive weather dashboard that allows users to view **current weather** and a **5-day forecast** for any location. The application effectively integrates with the OpenWeatherMap API, dynamically renders weather data, and presents it in an aesthetically appealing format. The dashboard works seamlessly across desktops, tablets, and mobile devices, offering a practical, interactive tool for weather updates while demonstrating skills in API integration, JavaScript programming, jQuery, and responsive front-end design.

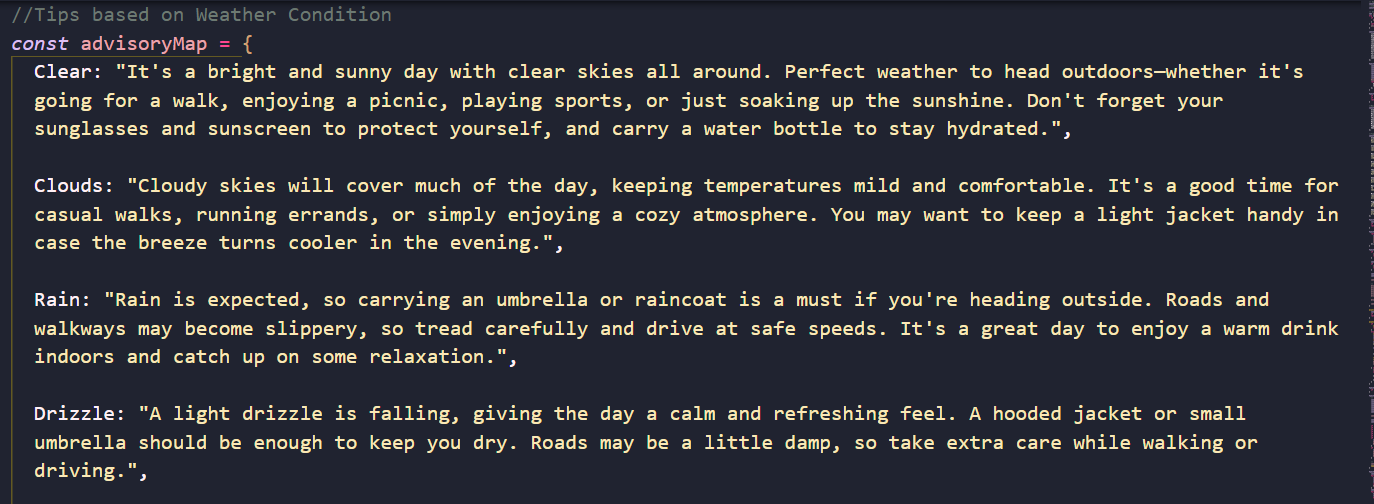
**Future Enhancements**

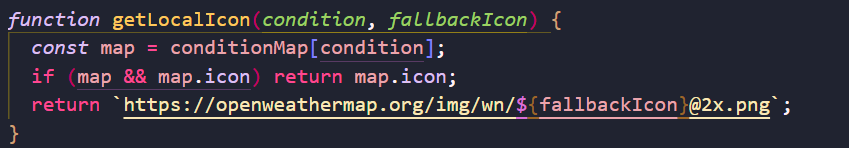
* Add **geolocation** feature to automatically detect the user’s location.
* Include **hourly weather forecast** and additional weather details such as wind speed, pressure, and UV index.
* Implement **search history** for previously searched locations.
* Optimize API calls and implement **caching** to improve performance.
* Add **dark/light mode toggle** for better user experience.

**Sample Code**

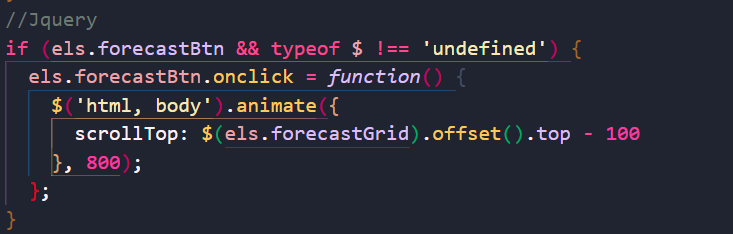
****

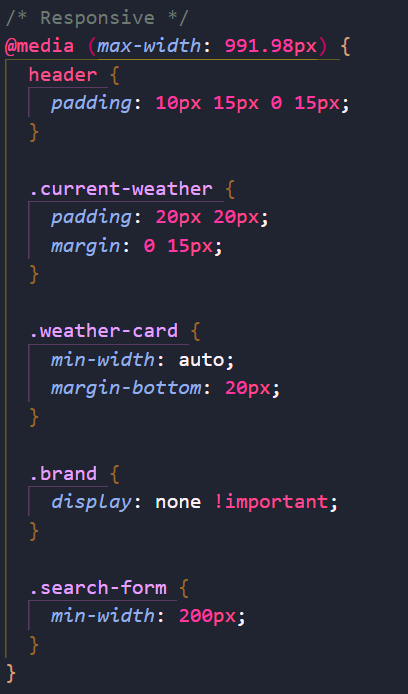
****

****

****

****

****

****