College code : 9607

College name : Immanuel arasar jj college of engineering

Department :BE(CSE)

Student NM-ID:aut2396070033

Roll no :20

Date :24-09-2025

Completed the project named as phase3:

B.sundhareswari, T.sivakalidevi, M .durgadevi

Technology project name :

Live Weather Dashboard

Submitted by,

Name : B.sundhareswari

Mobile no:8610176841

LIVE WEATHER DASHBOARD

Phase1:MVP Implementation:

Content:

Project setup

Core features implementation

Data storage (local state/database)

Testing core features

Version control (githup)

Project Setup: Live Weather Dashboard :

1.create project folder:

- On your PC, create a new folder and call it Weather Dashboard. All of your files will reside here.

2. Select Your Equipment:

There are two options available to you:

Easy: Just make use of JavaScript, HTML, and CSS.

Advanced: Make use of a contemporary framework, such as React, to make it appear more Polished.

3. Obtain meteorological information:

Register for a free account at open weather map.

They will provide you with an API key, which is a code that allows your app to communicate with their weather system.

4. Determine What You Wish to Present:

Your dashboard ought to exhibit elements such as.

Temperature

Humidity

Wind speed

Weather condition (Sunny, Rainy, Cloudy, etc).

5. Design Planning

Consider the dashboard as a control panel.

Utilize cards or boxes for each specific detail (for instance, one box for temperature and another for wind).

Ensure it appears clean and is easy to read (a blue or sky theme is quite effective).

6. Linking to Live Data:

Create a brief script that retrieves live weather information from OpenWeatherMap using your API key.

When a user inputs a city name, the dashboard will refresh automatically.

(Optional) Incorporate GPS location → allowing it to display your weather instantly.

7. Dashboard Testing:

Launch your project in a web browser.

Search for several cities (such as London, New York, or Chennai) → verify if it displays accurate weather information.

Modify the design to ensure compatibility with both desktop and mobile devices.

8. Additional Features (Optional, for enhancement):

Incorporate charts or graphs (to illustrate temperature trends).

Introduce a dark Mode.

Store recent searches.

Include a map view (similar to Google Maps style).

Core Features: Live Weather Dashboard :

1. Real-Time Weather Updates:

Display the current temperature, humidity, wind speed, and weather condition.

Automatically refresh every few minutes to ensure the data remains current.

2. City / Location Search:

Users can enter a city name (such as Chennai or New York).

The dashboard instantly provides the weather for that city.

There is also an option to use GPS for “current location weather.”

3. 5-Day / Weekly Forecast:

Show the weather forecast for the next 5–7 days.

Include minimum and maximum temperatures along with small weather icons.

4. Interactive Charts & Graphs:

Provide a temperature trend graph .

Include a rainfall or humidity graph for better comprehension.

5. Alerts & Notifications:

Display alerts for severe weather conditions (storm, heatwave, heavy rain).

Use red highlights or an alert icon to draw attention.

6. User-Friendly UI:

Present simple cards that display each weather detail.

Change the background based on the weather .

7. Extras (Nice-to-Have):

Include sunrise and sunset times.

Show the air quality index (AQI).

Display the feels-like temperature.

Provide an option to switch between °C and °F.

Data Storage : Live Weather Dashboard:

1. Local State (Temporary Storage):

Utilized within the app or browser for rapid, short-term data retention.

For instance: When you search for Chennai, the dashboard retains that weather information in memory for immediate display.

This data resets upon refreshing or closing the app.

User's temporary selections (°C / °F, dark mode, etc.)

2. Local Storage / Cache (Browser or App Memory):

Stores small amounts of data on the user's device to prevent reloading each time.

For example: The last searched city (Chennai) is remembered even if you access the dashboard again tomorrow.

Ideal for:

Recently searched cities.

User preferences (°C / °F, theme).

Last weather data for quick access.

3. Database (Permanent Storage):

Retains data on the server side for long-term accessibility.

For example: Storing historical weather patterns or user accounts.

Ideal for:

Saving user profiles (favorite cities, settings)

Collecting historical weather records (temperature trends over months)

Facilitating data sharing across multiple devices.

Testing Core Features – Live Weather Dashboard:

1. Real-Time Weather Updates:

Verify that the temperature, humidity, and weather conditions are updated accurately.

Refresh the page → the data should remain current.

Alter network speed (slow internet) → the dashboard should still load without any issues.

2. City / Location Search

Search for various cities .

The results should correspond to actual weather conditions.

Test GPS/“current location” → it should retrieve the correct city weather.

Incorrect city name → the dashboard should display a user-friendly error message, rather than crashing.

3. 5-Day / Weekly Forecast:

Review the forecast for 5–7 days.

Compare the displayed values with a reliable weather source.

Swipe or scroll → the forecast should be easy to navigate.

4. Interactive Charts & Graphs:

Access the hourly temperature chart → the values should align with the forecast data.

Tap/hover on graph points → it should reveal precise details (such as 30°C at 3 PM).

Resize the window → the chart should adjust appropriately.

5. Alerts & Notification:

Simulate severe weather conditions.

Verify if an alert banner or popup is displayed.

Ensure that alerts are clear but do not obstruct the entire dashboard.

6. User-Friendly UI:

Assess background changes (sunny = bright, rainy = cloudy).

Test dark mode against light mode.

Switch between °C and °F → all values should update accurately.

7. Performance & Reliability:

The dashboard should load within 2–3 seconds.

Open on mobile, tablet, and laptop → it should maintain a clean appearance across all devices.

Test without internet access → display a “No connection” message instead of failing.

Version Control : Live Weather Dashboard:

1. What is Version Control:

Consider it as a time machine for your project. It captures snapshots of your dashboard (design, data configuration, features) allowing you to:

Revert if you accidentally break something.

Identify who made which changes.

Collaborate with others without any confusion.

2. Why You Need It:

Safety net → If today’s update introduces errors, you can revert to yesterday’s functional version.

Teamwork → Each person works on their own copy, preventing anyone from overwriting another’s work.

History → You can monitor progress and comprehend the reasons behind changes made.

3. Popular Tools:

GitHub / GitLab / Bitbucket → Platforms for online storage and sharing of versions.

Git (operating in the background) → Executes the actual “time travel” for your files.

4. How It Works (Simple Steps):

1. Initialize → Begin tracking your dashboard files.

2. Commit → Save a snapshot accompanied by a brief note (such as “added temperature widget”).

3. Push → Upload it online (to GitHub, etc.) for backup.

4. Pull → Retrieve the latest version if you or someone else has made updates.