WeatherEase

ACCESSIBLE WEATHER APP FOR BLIND AND VISUALLY IMPAIRED USERS

Group Members:

- Tajra Sinanagic,
- Ehab Mohamed,
- Mike A. Khadeida.

Supervisor:

- Basak Taylan (CUNY Brooklyn College)

Course: CISC 4900

Date: Sept 2025

ORGANIZATION CHART

TAJRA – Team Lead
sinanagict@gmail.com

CODING
- Repo setup
- API
- Speech features

LEAD
- Creating Weekly Plans
- Coordinating tasks
- Reviewing progress

EHAB – Individual Contributor
ehabm7986@gmail.com

CODING
- UI structure
- Error handling
- Command mapping
- Testing & Debugging
- Mutual collaboration in code with Tajra

MIKE – Individual Contributor
mike.khadeida@gmail.com

RESEARCH
- Accessibility guidelines
- User testing plans
- Reporting findings
- Supporting UI/UX improvements

BASAK TAYLAN – Supervisor
basak.taylan@brooklyn.cuny.edu

Team ClearPath

Easy Breezy Forecasts

(elevator pitch)

WeatherEase is designed for blind and visually impaired users, making weather updates simple, fast, and inclusive.

Our mission is clear: ***to ensure everyone, rain or shine, can access the forecast with ease.***

By replacing visuals with spoken updates and intuitive voice controls, WeatherEase transforms weather checking into an effortless and empowering experience.

TOOLS & TECHNOLOGIES

LANGUAGES
- HTML5
- CSS3
- JavaScript

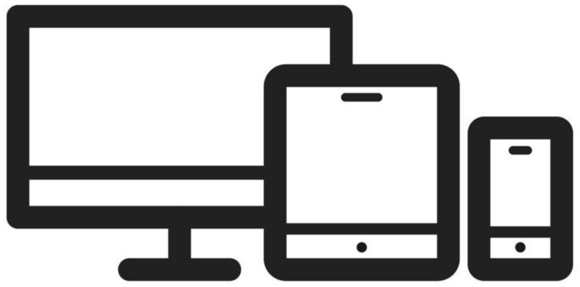
API
- OpenWeatherMap
- Web Speech API
- SpeechSynthesis API

PWA COMPONENTS
- manifest.json (app metadata & install behavior)
- Service Worker (offline caching)

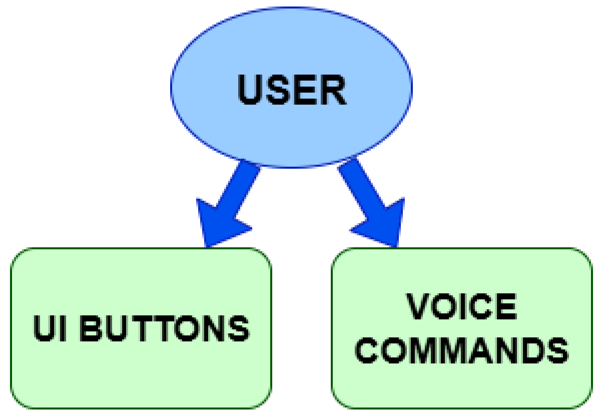
HOSTING
- GitHub
- GitHub Pages

PROJECT MANAGEMENT
- Notion
- Google Docs

DEVICE TESTING
- Chrome
- Microsoft Edge
- Safari (iOS)
- Android browsers (Google Play devices)

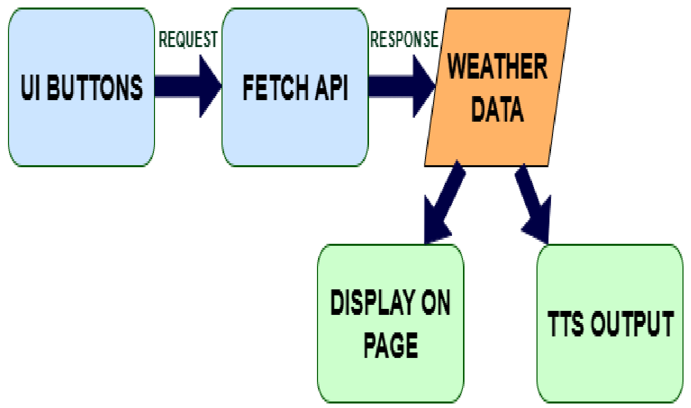


User Interaction Flow



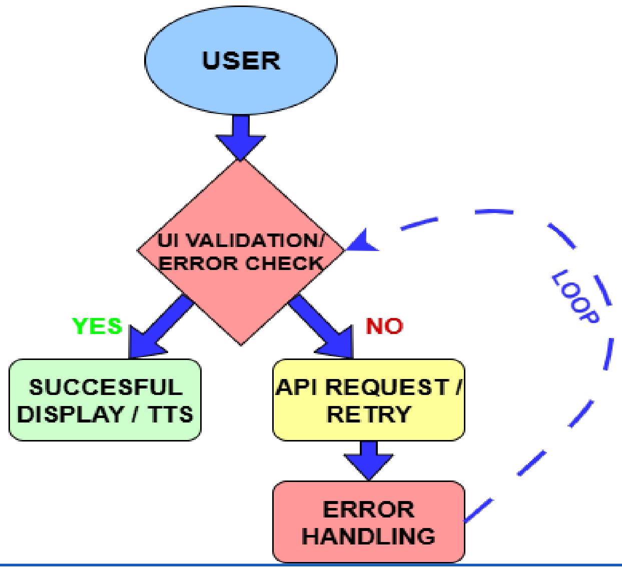
This diagram shows how the user interacts with the app, either by pressing buttons to get weather or listen, or by issuing voice commands for specific information.

Data Flow Diagram



This diagram shows how data flows through the app: the user interacts with UI buttons, which trigger API requests. The API retrieves weather data, which is then displayed on the page and read aloud via text-to-speech

Error Handling Flow



This diagram shows how the app handles errors: input is validated, and if successful, data is displayed and read aloud. If validation fails, the API request handles errors and loops back to re-check the input.

Tentative Schedule – Weeks 3–5 (Approx. 15 hrs/week)

Week	Tajra	Ehab	Mike
3: Sep 8 – 14	• repo set up • create starter files (index.html, style.css, script.js, manifest.json, service-worker.js) • push to GitHub • enable Pages • basic planning	• build HTML layout (<header>, <main>, <footer>) • add weather button • test keyboard nav • minor CSS • debug layout	• write accessibility guide for blind users • share with team • review HTML semantic usage
4: Sep 15 – 21	• register OpenWeather API key • fetch weather data handle responses • debug integration • update scripts	• display API data on page • style temperature/humidity/wind info • test keyboard access • minor UI fixes	• update accessibility checklist with ARIA attributes • note progressive enhancement • review dynamic content
5: Sep 22 – 28	• implement modular TTS function using SpeechSynthesis API • test with sample text • debug	• connect button to TTS • provide visual feedback while reading • test keyboard/mouse	• suggest concise TTS wording • test voice settings • guide on reading dynamic info clearly

This schedule shows planned tasks and estimated time per team member for Weeks 3–5, including coding, research, testing, and coordination.

DATA SOURCE: OPENWEATHERMAP API

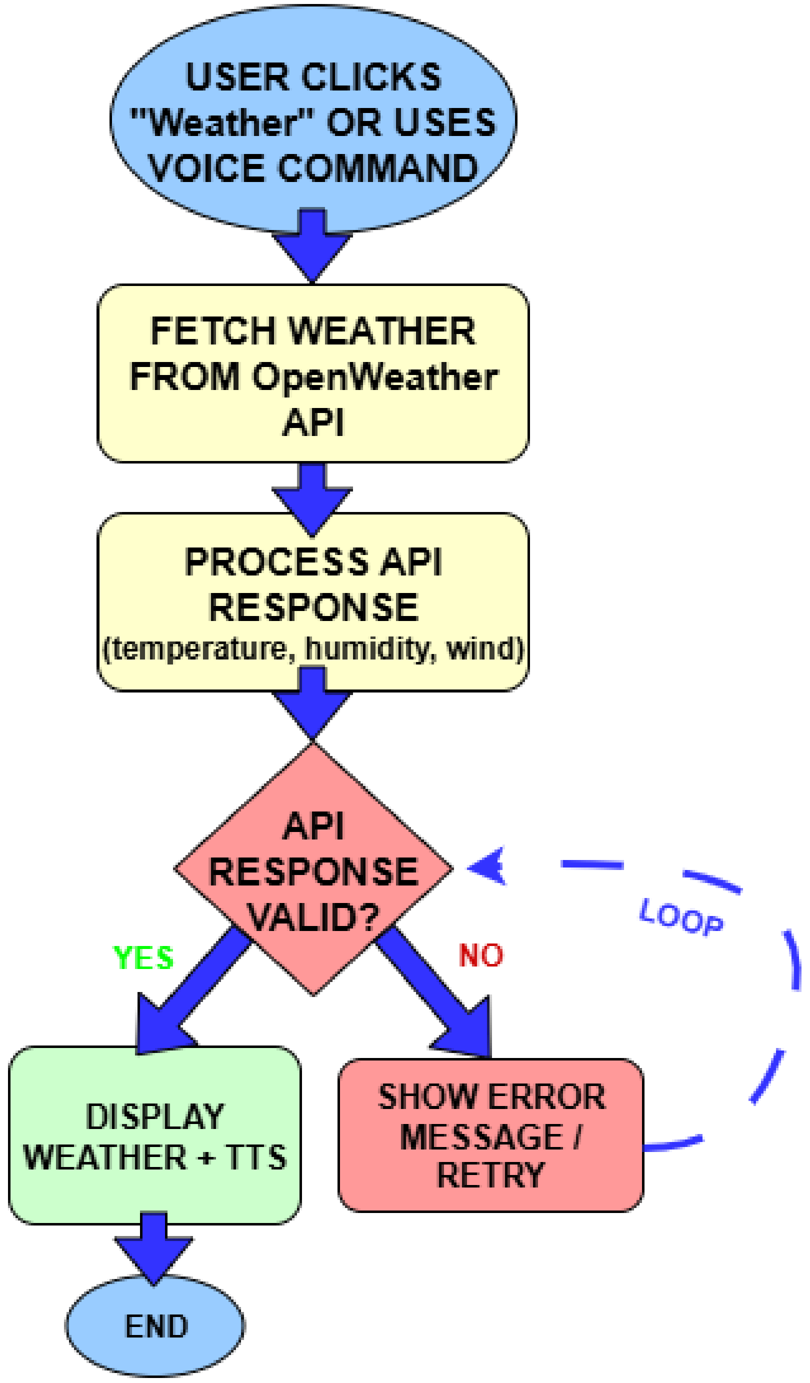
DATASET NATURE

- Provides **current weather** and **forecasts** (hourly up to 4 days, daily up to 16 days).
- Includes **historical data** and statistical weather parameters.
- Offers **weather maps** with multiple layers and air pollution data.
- Accessible via **geocoding API** for location-based queries.
- Responses are structured in **JSON format** and updated in real-time.
- API usage limits: 3,000 calls/minute, 100 million calls/month for current/forecast data; 50,000 calls/day for historical data.

OpenWeather

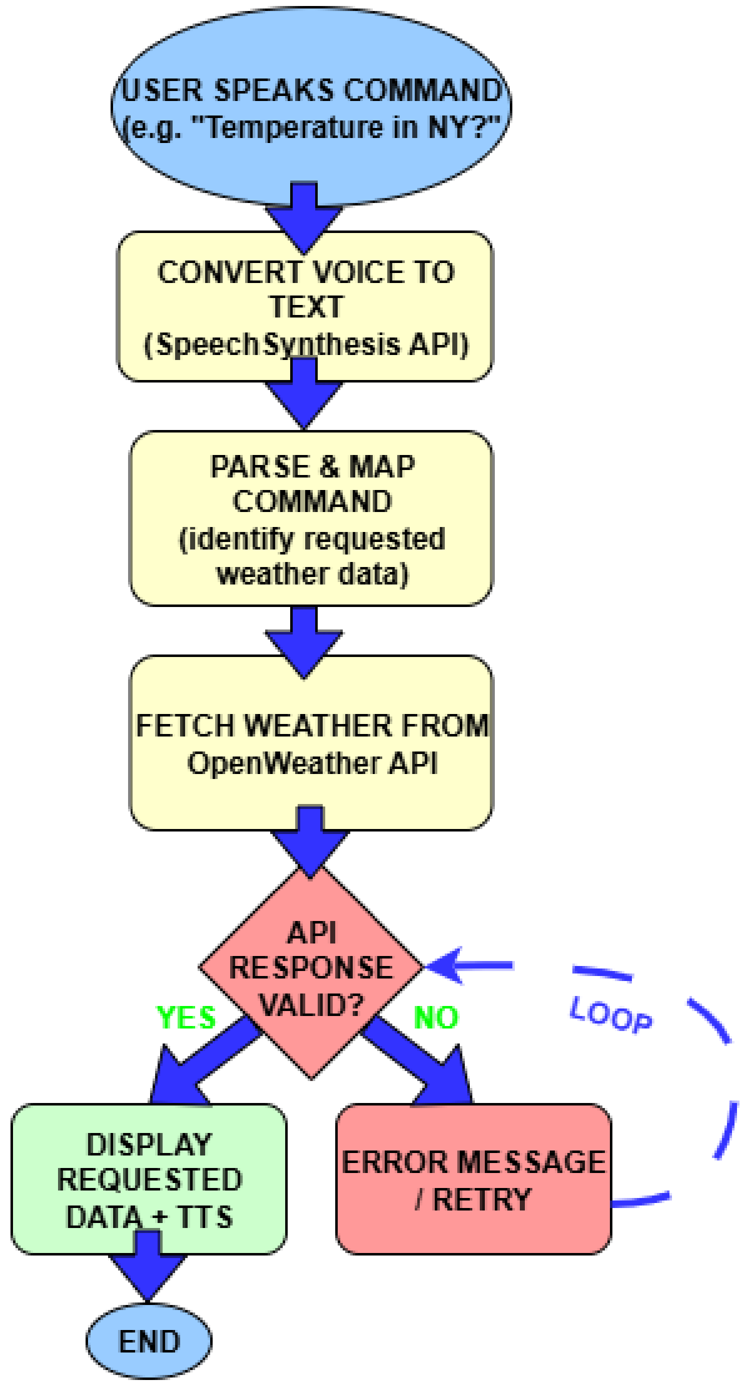
CASE 1: CHECK CURRENT WEATHER

- In this use case, the user checks the current weather by clicking the 'Get Weather' button or using a voice command.
- The app fetches data from the OpenWeatherMap API, processes the temperature, humidity, and wind speed, and displays the results on the screen.
- Optionally, the system reads the information aloud using text-to-speech.
- If the API response is invalid, an error message is shown, and the system automatically retries the request.


USER CLICKS "Weather" OR USES VOICE COMMAND
FETCH WEATHER FROM OpenWeather API
PROCESS API RESPONSE (temperature, humidity, wind)
API RESPONSE VALID?
YES
NO
LOOP
DISPLAY WEATHER + TTS
SHOW ERROR MESSAGE / RETRY
END

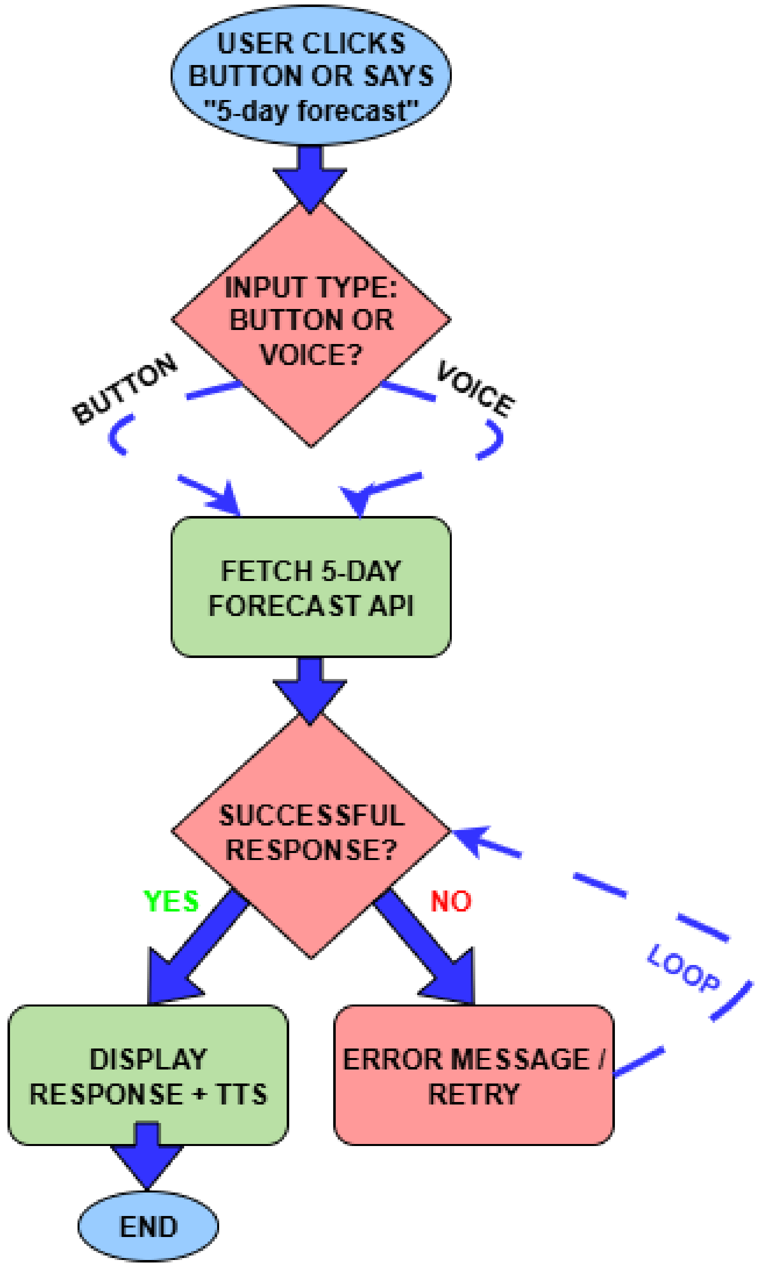
CASE II: VOICE QUERY FOR WEATHER DETAILS

- In this scenario, the user speaks a command, such as "What's the temperature in New York?".
- The system first uses the SpeechSynthesis API to convert the spoken input into text, then parses and maps the command to identify the requested weather data.
- A call is made to the OpenWeatherMap API, and if the response is valid, the result is displayed on the screen and also read aloud using text-to-speech for convenience.
- If the API request fails or the command cannot be understood, the system provides an error message and prompts the user to retry.


USER SPEAKS COMMAND (e.g. "Temperature in NY?")
CONVERT VOICE TO TEXT (SpeechSynthesis API)
PARSE & MAP COMMAND (identify requested weather data)
FETCH WEATHER FROM OpenWeather API
API RESPONSE VALID?
YES
NO
LOOP
DISPLAY REQUESTED DATA + TTS
ERROR MESSAGE / RETRY
END

CASE III: FETCHING A 5-DAY FORECAST

- In this scenario, the user requests a "5-day weather forecast."
- The system calls the OpenWeatherMap Forecast API and retrieves hourly and daily forecast data for the next five days.
- The information is displayed in a clean format on the screen (such as date, temperature, humidity, and conditions).
- At the same time, a summary of each day's weather is read aloud using text-to-speech for accessibility.
- If the API call fails, the user is shown an error message with the option to retry.


USER CLICKS BUTTON OR SAYS "5-day forecast"
INPUT TYPE: BUTTON OR VOICE?
BUTTON
VOICE
FETCH 5-DAY FORECAST API
SUCCESSFUL RESPONSE?
YES
NO
LOOP
DISPLAY RESPONSE + TTS
ERROR MESSAGE / RETRY
END