Project Requirements Document(PRD)

Title: Weather App

# Overview:

This Weather App is designed to provide users with current weather conditions for any city they search for. The app fetches real-time data from the OpenWeatherMap API and displays the temperature, weather icon, description, and the date. Additionally, it uses the Google Custom Search API to fetch a dynamic background image of the city entered by the user. Built with Django as the backend and styled using CSS and Google Fonts, the application ensures a visually appealing and responsive user experience.

# Goals & Objectives:

* Enhance the user interface with relevant city images.
* Ensure responsiveness and usability across devices.
* Provide error handling for invalid or unrecognized cities.
* Integrate OpenWeatherMap API for real-time data.
* Use Google Custom Search for city image backgrounds.
* Maintain simple and clear user interface.

# Features:

| **Feature** | **Description** | **Priority** |
| --- | --- | --- |
| View Weather | Display current weather of a searched city including temperature and condition | High |
| |  | | --- | | City Search |  |  | | --- | |  | | Allow users to input and search any city name | High |
| |  | | --- | | City Image Display |  |  | | --- | |  | | Users can view and manage only their own posts | Medium |
| Error Handling | Display alerts and default data if city is not found or API fails | High |
| Responsive UI | Ensure weather display is mobile-friendly and visually clean | Medium |

# Functional Requirements

# **View Events** Users can search and view current weather conditions of any city. The app displays relevant information such as temperature, weather description, date, and a weather icon. Additionally, a city-related background image is fetched using the Google Custom Search API. All weather data is retrieved from the OpenWeatherMap API in real time.

# **Reminder System** A basic alert system notifies users when weather data for a searched city is unavailable or invalid. The user receives an in-app alert with a friendly error message, helping them understand the issue and encouraging correction. This system enhances user experience by providing timely and clear feedback.

# **Filter System** To enhance usability, the Weather App includes a city search feature allowing users to input any city name and view real-time weather data. The interface supports keyword-based city filtering, enabling quick access to desired locations. If the input is invalid or the data is unavailable, a fallback alert ensures a smooth user experience.

# UI/UX Requirements

# Clean, mobile-first responsive design for seamless user experience across all screen sizes. • Three main interface screens:  o **Home:** Displays real-time weather information including temperature, description, and icon based on the selected city.  o **City Search & Result:** Allows users to input a city name and view the corresponding weather data.  o **Error/Alert Display:** Notifies users with a styled message and alert when an invalid city is searched or data is unavailable.

# **Tech Stack**

# **Frontend:** Built using HTML, CSS, and basic JavaScript to handle user interactions and dynamic UI updates.

# **Backend:** Powered by Django, which handles routing and logic, and integrates with external APIs.

# Timeline

| Task | Deadline |
| --- | --- |
| Requirement Finalization | Day 1 |
| UI Wireframes | Day 2 |
| Build Event List View | Day 3 |
| Add Filter & Reminder | Day 4 |
| Testing | Day 5 |
| Final Touches & Report | Day 6 |

# Success Criteria

* App should successfully load and display weather data for at least 5 different cities.
* User can search for any city and view real-time weather information.
* Mobile-responsive layout with visually appealing and user-friendly design.
* No major bugs or crashes encountered during testing across common browsers and devices.

Deliverables

* Fully working blog app (deployed via PythonAnywhere or similar).
* Source code hosted on GitHub.
* Short demo video (2–3 minutes showing key features).
* Final project report (PDF including screenshots, features, and learnings).