

1. Project Overview

App Name: Weather Today

Description:

The app is created which can be used to view live weather of entire cities of the world in real time. It fetches weather data from OpenWeatherMap API and displays it in an appealing manner. It has functionalities such as searching weather of any city, handling errors and providing responsive design across various screen sizes.

2. Functionality

- **Fetches Data from Public API:**
 - API: [OpenWeatherMap](#)
 - The app fetches weather data based on the user's current location or a manually searched city.
- **Displays Data:**
 - The app displays the current temperature, weather condition, and animations corresponding to weather conditions (e.g., sunny, rainy, cloudy).
- **User Interaction:**
 - Search functionality to find weather data for a specific city.
 - Displays a message if the city is not found.

2.2 Code Requirements

- **Clean and Well-Structured Code:**
 - Follow the best practices for Flutter development, including meaningful variable names, proper indentation, and modular code.
- **Flutter Best Practices:**
 - Use of Stateless and Stateful widgets appropriately.
 - Separation of business logic from UI components.
- **Responsive Design:**
 - Use of Flex and Expanded to ensure responsiveness across different devices and screen sizes.

- **Error Handling and Data Validation:**
 - Handling of API errors (e.g., network issues, incorrect city name).
 - User-friendly error messages.
- **State Management:**
 - Use of basic state management with **useState**.

3. Implementation Details

3.1 Fetching Weather Data

- **API Integration:**
 - Integrated the OpenWeatherMap API using the http package.
 - Utilized methods to fetch weather data for the current city and for user-searched cities.

3.2 Display of Data

- **UI Components:**
 - **WeatherPage:** Main screen displaying the weather data.
 - **Search Bar:** Allows the user to search for different cities.
 - **Lottie Animations:** Displays relevant animations based on the current weather condition (sunny, cloudy, rainy).
 - **App Name and Splash Screen:** Displays a loading animation using Lottie before the app loads.

3.3 User Interaction

- **Search Feature:**
 - Implemented a search bar that allows users to enter a city name.
 - Fetches and displays the weather data for the entered city.
 - Displays an error message if the city is not found.

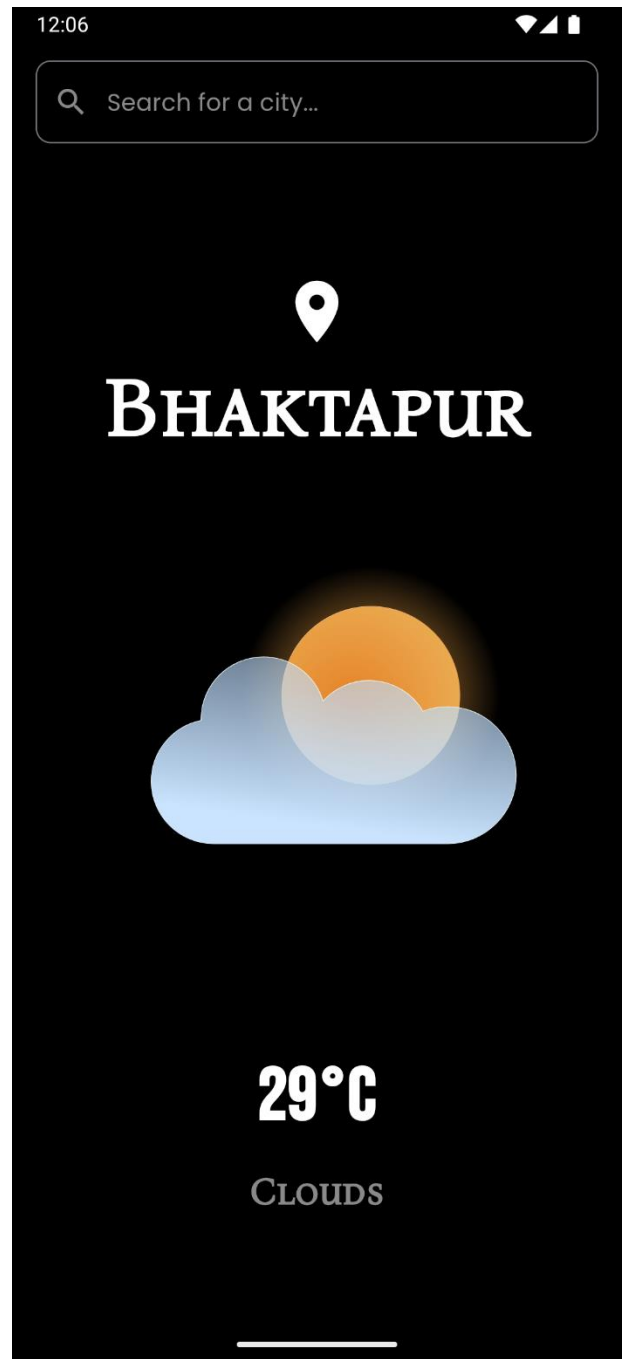
3.4 Error Handling

- Added error handling to manage network issues, incorrect city names, and API errors.

(All the images shown here are animation created via **Lottie animation**)

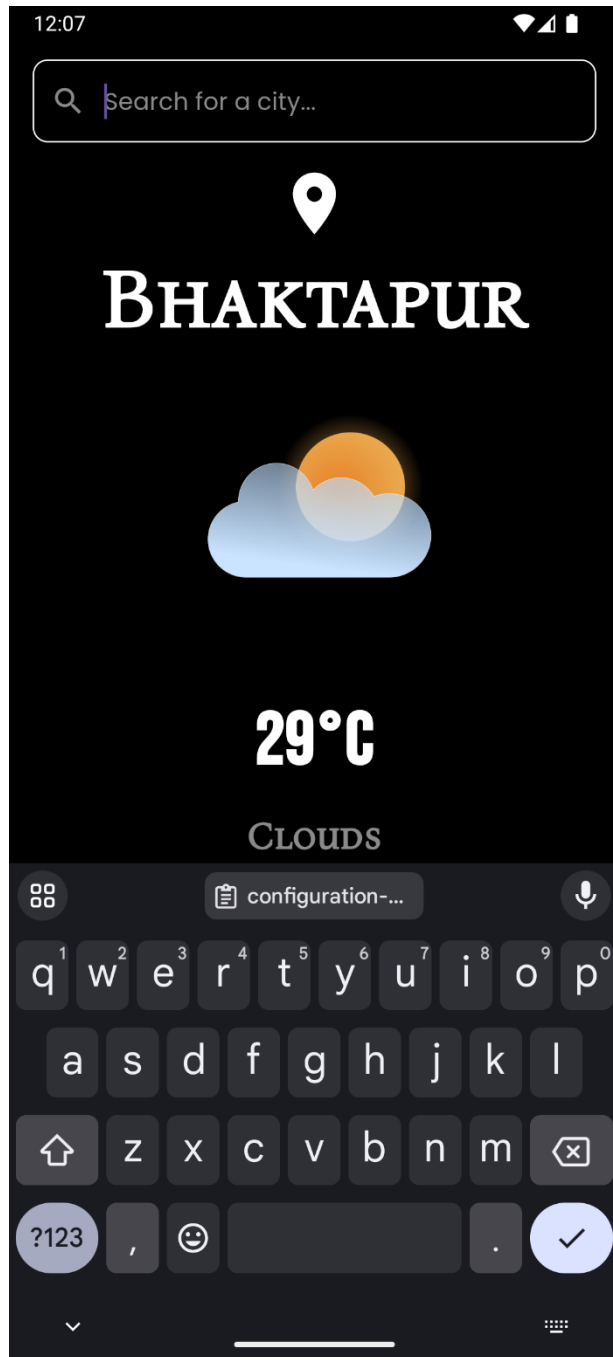


Splash Screen



Home Page

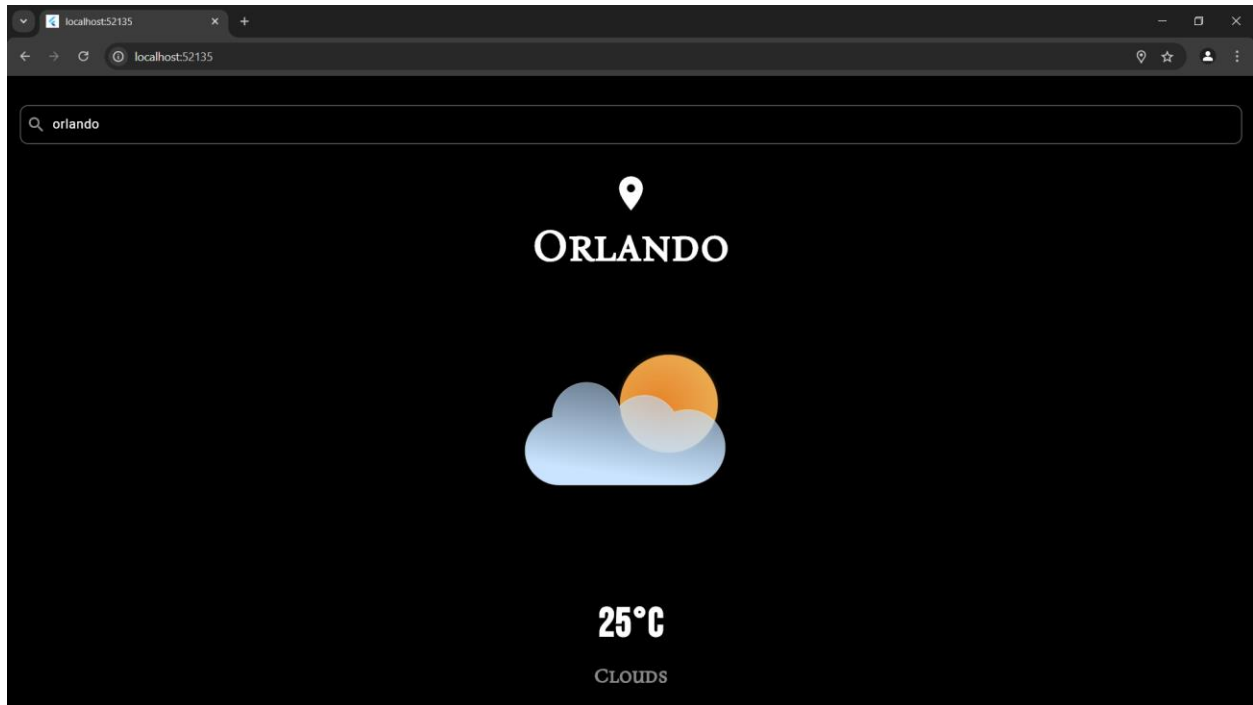
(All the images shown here are animated created via **Lottie animation**)



Search Field



New City



App demonstration on chrome browser