

Weather Forecast Application Documentation

GitHub repository : <https://github.com/NIHAL3403/weather-app>

Overview

The **Weather Forecast Application** is a responsive web application that allows users to search for weather forecasts by city name or detect their current location to display the weather for that location. The application fetches weather data and a 5-day extended forecast using the **OpenWeatherMap API**. The user interface is styled using **Tailwind CSS** and includes animations for an enhanced user experience.

Features

1. **Search by City Name:** Users can search for the weather of any city by entering its name in the input field.
2. **Detect Current Location:** Users can click the "Get Current Location" button to fetch weather data for their live location.
3. **Display Current Weather:** The application shows the current weather, including temperature, humidity, and weather conditions.
4. **5-Day Extended Forecast:** The application displays a 5-day weather forecast with temperature, humidity, and an icon representing the weather condition.
5. **Responsive Design:** The app is responsive, ensuring a smooth experience across devices like desktops, tablets, and phones.
6. **Animations:** Smooth animations have been added to improve the user experience.

Technologies Used

- **HTML:** Structure of the web application.
- **JavaScript:** Logic for fetching and displaying weather data.
- **Tailwind CSS:** For styling the interface with utility-first CSS.
- **Custom CSS:** For additional animations and styles.
- **OpenWeatherMap API:** Fetches weather data based on city name or coordinates.
- **Geolocation API:** Detects the user's current location to fetch weather data for that location.

Prerequisites

- **OpenWeatherMap API Key:** You must have an API key from OpenWeatherMap. You can get one by signing up at OpenWeatherMap.
- **Browser with Geolocation support:** To detect the user's current location, the browser must support the **Geolocation API**.

Project Structure

```
/weather-app
└── dist
    └── tailwind.css      # Compiled Tailwind CSS file
    └── index.html        # Main HTML file
    └── script.js          # JavaScript file for fetching
and displaying weather data
└── styles.css           # Custom CSS for animations and
    └── tailwind.css      # Tailwind configuration file
    └── README.md          # Documentation
```

File Descriptions

1. `index.html`

This is the main HTML file that structures the weather forecast application. It includes:

- A header for the app title.
- A form where users can input a city name and a button to search for the city's weather.
- A "Get Current Location" button to fetch the weather based on the user's live location.
- Sections to display the current weather and the 5-day extended forecast.

2. `script.js`

This JavaScript file handles the application's logic, including:

- **Fetching weather data** from the OpenWeatherMap API using city names or geographical coordinates.
- **Displaying current weather** and **5-day forecasts** based on user input or live location.
- **Error handling** for both API requests and geolocation errors.
- The use of `fetch()` to make API requests.

3. `styles.css`

This file contains custom CSS rules to complement Tailwind CSS for animations and visual effects, such as:

- **Animations:** Used for fading in elements, sliding in forms, and scaling forecast cards on hover.
- **Custom styles:** Includes input field focus effects and button hover transitions.

4. tailwind.css

This file contains the basic Tailwind CSS setup and is compiled into the `dist/tailwind.css` file using Tailwind's CLI.

Key Functions in `script.js`

Geolocation Detection

```
navigator.geolocation.getCurrentPosition(showPosition,  
showError);
```

- **showPosition:** This function is triggered when the user's location is successfully obtained. It fetches the weather data using the latitude and longitude.
- **showError:** This function handles errors in getting the user's location (e.g., if the user denies the permission).

Fetching Weather Data

The function `fetchWeatherDataByCoords(lat, lon)` is called when the user's coordinates are available. It sends a request to OpenWeatherMap's API to fetch weather data based on the user's latitude and longitude.

```
async function fetchWeatherDataByCoords(lat, lon) {  
  const response = await fetch(` ${API_URL}?lat=${lat}&lon=${lon}&appid=${API_KEY}&units=metric`);  
  const data = await response.json();  
  displayWeatherData(data); // Displays the fetched weather  
  data on the webpage.  
}
```

Fetching Extended Forecast

The function `fetchExtendedForecastByCoords(lat, lon)` retrieves a 5-day weather forecast for the detected location or searched city.

```
async function fetchExtendedForecastByCoords(lat, lon) {  
  const response = await fetch(` ${FORECAST_URL}?lat=${lat}  
&lon=${lon}&appid=${API_KEY}&units=metric`);  
  const data = await response.json();  
  displayExtendedForecast(data); // Displays the 5-day  
  forecast.  
}
```

Installation and Setup

1. Install Tailwind CSS (Optional)

If you want to use Tailwind CSS locally and customize it, follow these steps:

1. Initialize your project with npm:

```
npm init -y
```

2. Install Tailwind CSS via npm:

```
npm install -D tailwindcss
```

3. Create a configuration file:

```
npx tailwindcss init
```

4. Add Tailwind to your CSS:

```
@tailwind base;
```

5. @tailwind components;

6. @tailwind utilities;

7. Build your CSS:

```
npx tailwindcss -i ./tailwind.css -o ./dist/tailwind.css  
--watch
```

2. Add OpenWeatherMap API Key

Open the `script.js` file and replace the placeholder '`YOUR_API_KEY`' with your

```
const API_KEY = 'YOUR_API_KEY'; // Replace with your actual  
API key
```

3. Running the Application

After setting up the project, simply open the `index.html` file in your browser to see the application in action.

Usage

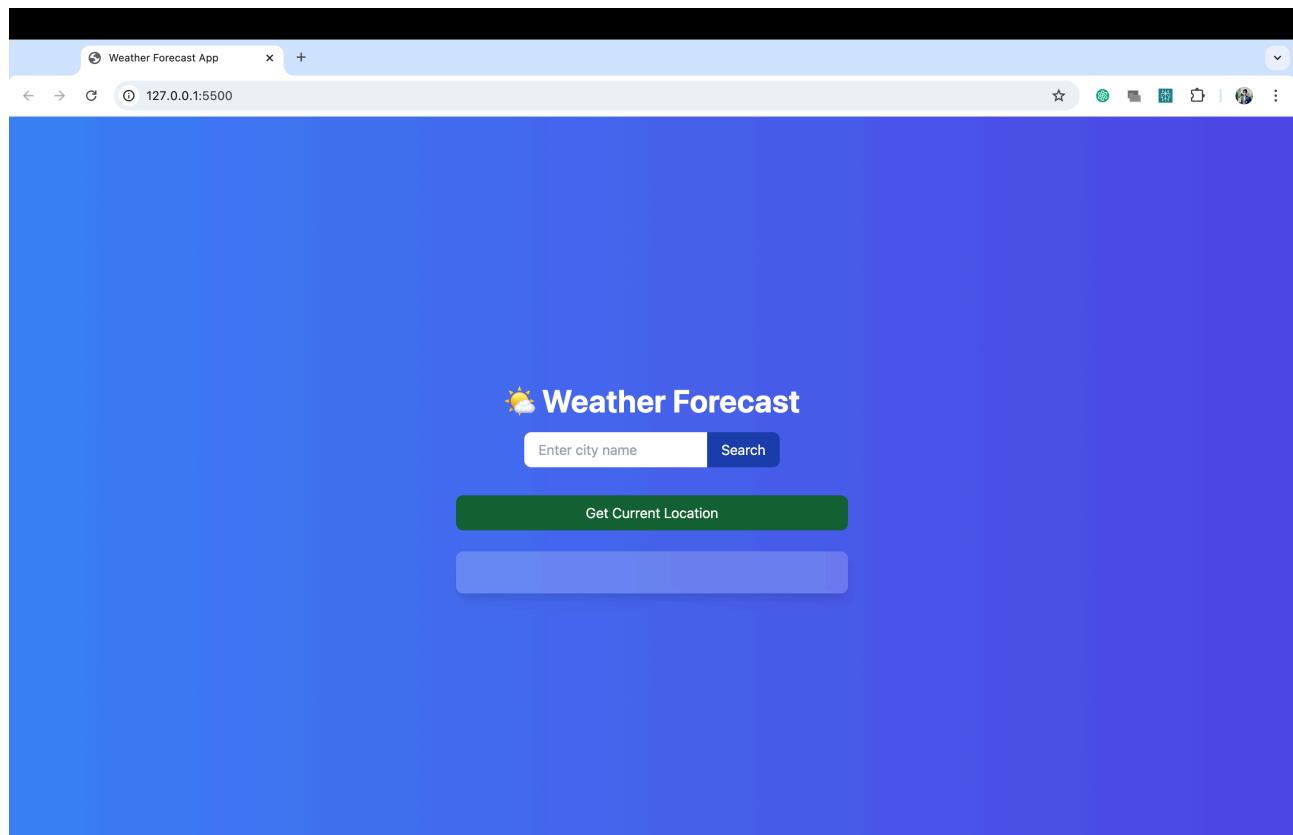
- **Search for a city:** Enter the name of the city in the search bar and click the "Search" button to display the current weather and 5-day forecast.
- **Get current location weather:** Click the "Get Current Location" button to display the weather for your live location. Ensure that your browser allows location access.

Troubleshooting

1. **Geolocation Not Working:** Make sure you have granted location permissions in your browser. Check if the Geolocation API is supported by your browser.
2. **Weather Data Not Displaying:** Ensure that you have entered a valid city name or that your API key is correctly configured.
3. **Console Errors:** Open the browser console (F12 or right-click → Inspect → Console) and check for any errors related to the API requests or geolocation.

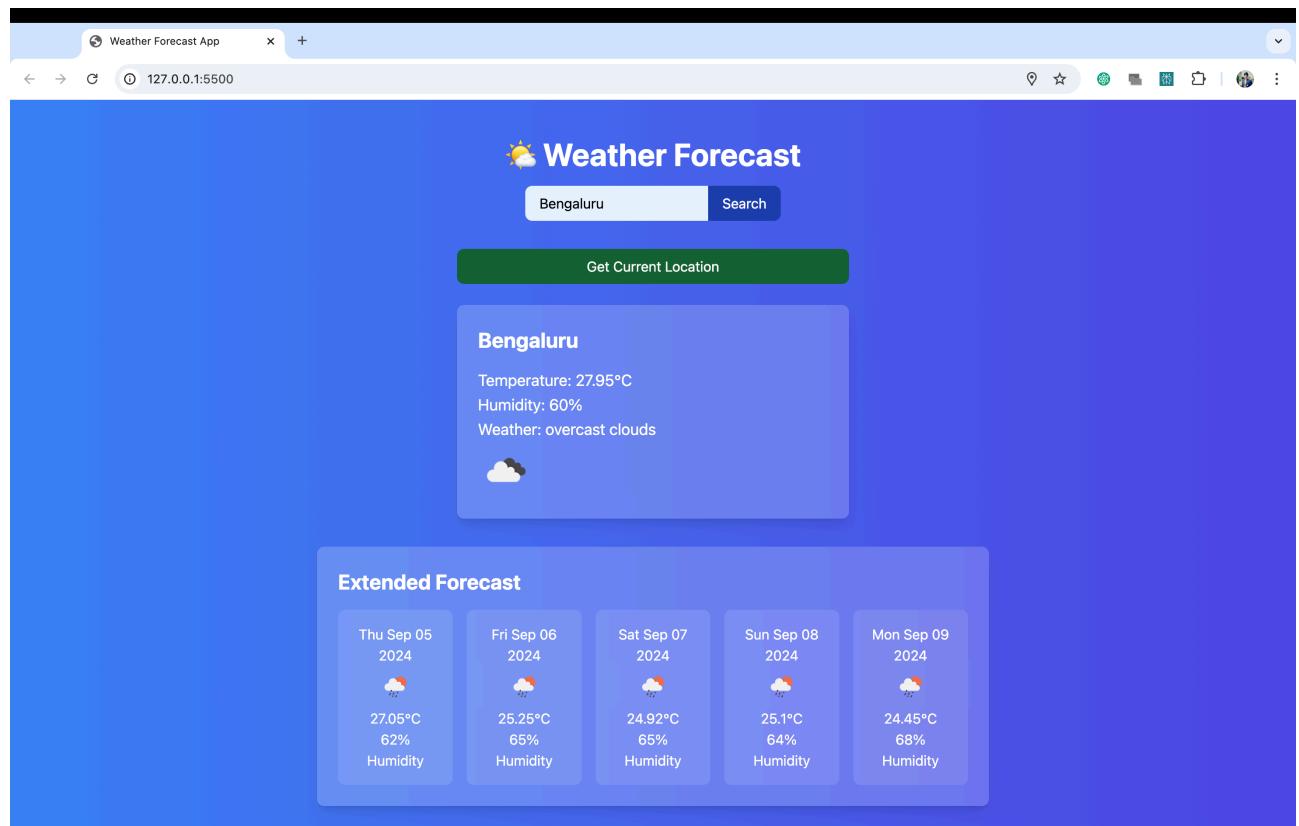
Screenshots

Initial View:



Weather After Search:

By Searching City :



By clicking on Current Location :

