Weather API App Documentation

Overview

The Weather API App is a React-based web application that provides real-time weather information and a 5-day forecast for a specified city. Built using TypeScript, Tailwind CSS, and Framer Motion, the app fetches data from the OpenWeatherMap API and features a responsive design with dark mode, unit switching (Celsius/Fahrenheit), and an interactive interface. The project is part of a #100DaysOfCode challenge and showcases learning in React, API integration, and state management.

- **Author**: tkntsh
- **Start Date**: October 17, 2025
- **Version**: 0.1.0
- **Last Updated**: October 19, 2025
- **Repository**: [https://github.com/tkntsh/weather-app]
- **Deployment**: [https://weather-app-heroku-4245cfd7ec00.herokuapp.com/]

Features

- **Current Weather**: Displays temperature, weather description, city name, humidity, wind speed, sunrise, and sunset times.
- **5-Day Forecast**: Shows daily temperature, weather description, and icons for the next five days.
- **Unit Switching**: Toggle between metric (°C) and imperial (°F) units.
- **Dark Mode**: Switch between light and dark themes.
- **Search Functionality**: Enter a city name to fetch weather data.
- **Animations**: Smooth transitions and icon animations using Framer Motion.
- **Responsive Design**: Adapts to mobile, tablet, and desktop screens.

Prerequisites

- **Node.js**: Version 14.x or higher.
- **npm**: Version 6.x or higher (included with Node.js).
- **Text Editor**: Visual Studio Code (recommended) with TypeScript and Tailwind CSS extensions.
- **Internet Connection**: Required for API calls to OpenWeatherMap.

Installation and Setup

1. Clone the Repository

If hosted on GitHub, clone the repository:

`bash`

```
git clone <repository-url>
cd weather-app
```

If not using Git, download the project files manually.

2. Install Dependencies

Run the following command to install all required packages:

`bash`

```
npm install
```

3. Configure Environment Variables

- Create a `.env` file in the root directory (`C:\Users\Ntokozo\weather-app\.env`).
- Add your OpenWeatherMap API key:
- REACT_APP_WEATHER_API_KEY=your_api_key_here
- Obtain an API key from OpenWeatherMap if you don't have one.

4. Project Structure

```
C:\Users\Ntokozo\weather-app\
├── node_modules/          # Managed by npm, stores all installed packages
├── public/                # Static files
│   ├── index.html
│   ├── favicon.ico
│   └── manifest.json
├── src/                   # Source code
│   ├── components/        # React components
│   │   ├── Search.tsx
│   │   ├── Weather.tsx
│   ├── types.d.ts         # Custom TypeScript definitions
│   ├── App.tsx            # Main app component
│   ├── index.tsx          # Entry point
│   └── styles/            # CSS or Tailwind config
│       └── index.css
├── .env                   # Environment variables
├── package.json           # Project dependencies and scripts
├── tsconfig.json          # TypeScript configuration
├── craco.config.js        # CRACO configuration for Tailwind
├── README.md              # Project documentation
```

```
└── build/              # Output after `npm run build` (generated)
```

5. Run the Application

- Start the development server:
- `bash`
- npm start
- Open `http://localhost:3000` in your browser to view the app.

6. Build for Production

- Create a production build:
- `bash`
- npm run build
- The `build/` folder will contain the optimized files for deployment.

Usage

Interacting with the App

- **Search**: Enter a city name in the search bar and press Enter or click the search icon to update the weather data.
- **Unit Toggle**: Click the °C/°F button to switch temperature units.
- **Dark Mode**: Click the "Light"/"Dark" button to toggle the theme.
- **Details**: Click "Show Details"/"Hide Details" to expand/collapse additional weather information (humidity, wind speed, sunrise, sunset).
- **5-Day Forecast**: View the daily forecast below the current weather, with icons, temperatures, and descriptions.

Example Output

- For Abuja:
  - Current: 24.4°C, Clouds
  - Forecast: Daily temperatures with icons (e.g., WiDaySunny, WiRain).

Dependencies

- **Core Libraries**:
  - `react`: ^18.2.0
  - `react-dom`: ^18.2.0

    - `typescript`: ^4.9.5
    - `react-scripts`: ^5.0.1
- **API and Data**:
    - `axios`: ^1.6.0
    - `dotenv`: ^16.3.0
- **Styling and Animation**:
    - `tailwindcss`: ^3.0.0
    - `framer-motion`: ^11.0.0
    - `react-icons`: ^4.12.0
- **Development Tools**:
    - `@craco/craco`: ^7.0.0
    - `@testing-library/*`: For testing
    - `webpack`: ^5.88.0 (with polyfills)

Troubleshooting

Common Issues

- **API Key Error**: Ensure `REACT_APP_WEATHER_API_KEY` is set in `.env` and matches your OpenWeatherMap key. Check the console for "Failed to fetch weather data".
- **TypeScript Errors (e.g., TS2305)**:
    - Add or update `src/types.d.ts` with custom declarations for `react-icons/wi`:
    - `typescript`

```
declare module 'react-icons/wi' {
  import { ComponentType } from 'react';
  export const WiCloudy: ComponentType<any>;
  export const WiRain: ComponentType<any>;
  export const WiSnow: ComponentType<any>;
  export const WiDaySunny: ComponentType<any>;
```

    - }
- - Restart the TypeScript server in VS Code (Command Palette: "TypeScript: Restart TS Server").
- **Icons Not Displaying**: Verify `react-icons` is installed (`npm list react-icons`). If issues persist, consider switching to `weather-icons-react`:
- `bash`
- npm install weather-icons-react
    - Update `Weather.tsx` imports to `import { WiDayCloudy, WiRain, WiSnow, WiDaySunny } from 'weather-icons-react';`.
- **Squished Layout**: If the 5-day forecast looks cramped, ensure `App.tsx` uses `max-w-4xl` and adjust `Weather.tsx` grid gaps or padding.
- **Build Failures**: Clear the cache and reinstall dependencies:

- `bash`

```
rmdir /s /q node_modules/.cache
npm install
```

- npm start

Logging

- Check the browser’s Developer Tools (F12) Console tab for API errors or JavaScript issues.
- Use `console.log` in `Weather.tsx` (e.g., within `useEffect`) to debug data fetching.

Deployment

- Host the `build/` folder on a static file server (e.g., Netlify, Vercel).
- Update the API key in the environment variables on the hosting platform.

Future Improvements

- **Add Error Handling UI**: Display user-friendly messages for API failures.
- **Enhance Icons**: Integrate animated icons with `weather-icons-react` or custom SVGs.
- **Mobile Optimization**: Add touch gestures for navigation.
- **Unit Tests**: Implement tests with `@testing-library/react`.

Contributing

- Fork the repository, create a branch, and submit pull requests.
- Report issues or suggestions via [GitHub Issues] (if applicable).

License

- [Specify license, e.g., MIT] - Add details if applicable.

Notes

- **Customization**: Adjust the `README.md` content in the project root to mirror this documentation for public use.

- **Testing**: After updating files, run `npm start` and verify all features work. Test with different cities and screen sizes.
- **Feedback**: Let me know if you’d like to expand any section or add specific details (e.g., API endpoints, code snippets)!