Weather App Documentation

App description:

The app shows the current weather in the user's location and the forecast for 24 hours or 7 days. The user can search for weather in a different location by city name and save the city into favourites.

Here are the main features:
--- Get weather in another location with city name.
--- Choose between list and chart visualizations.
--- Switch between hourly and daily forecast.
--- Cycle between °C, °F, and °K temperature units.
--- Save the current city to favourites.
--- Remove the current city from favourites.

Design and logic:

When loaded, the app tries to get user location to fetch the local weather data. If it fails, it defaults to Paris, FR. The current weather shows the city name and country code (for added clarity), weather type as an icon, the date and time, a verbal description, the temperature and what it feels like, humidity percentage and the wind speed. There is also a persona icon that reacts to temperature with 5 different faces. The current weather is fetched from Open Weather Map's API, and the weather type icon is fetched from openweathermap.org based on the weather type code.

The hourly forecast in the list view displays the hour, the weather type as icon, the temperature and precipitation. The daily forecast shows the weekday, dominant weather icon, the temperature range and the sum of precipitation. The hourly forecast graph shows the temperature development and hourly precipitation more accurately, and the daily graph shows the min and max temperatures and total precipitation per weekday.

The forecast data is fetched from Open-Meteo's API. By default, the app renders the hourly forecast data in list view. The two sources provide weather type codes in different format, and so the weather icons used in the list view are loaded locally. They are the same as used in the Yr weather app and copied from https://github.com/metno/weathericons.git.

The location is changed by searching with a city name. The app queries Open weather map directly with the name and extracts the coordinates from the response to variables, which are then used to fetch the weather forecast data from Open Meteo. This is to ensure that both sources point to the same location. The app updates the current weather and the forecast, considering the selected visualization and timespan. To avoid constat API calls for the same location, the app caches the forecast data during the

session for max 1 hour. If the data in the cache is older than 1 hour, it is fetched again. However, the current weather data is intentionally never cached, as it may change rapidly.

The user can save the current location to favourites using the star button. Favourites are stored to the browser's localStorage. The app checks if the current coordinates already exist in the storage and stores them with the city name if not. The favourites are rendered in a dropdown menu in the search bar. A separate button removes the current city from favourites.

All temperatures are rendered using a function that converts them to the selected unit, by default °C.

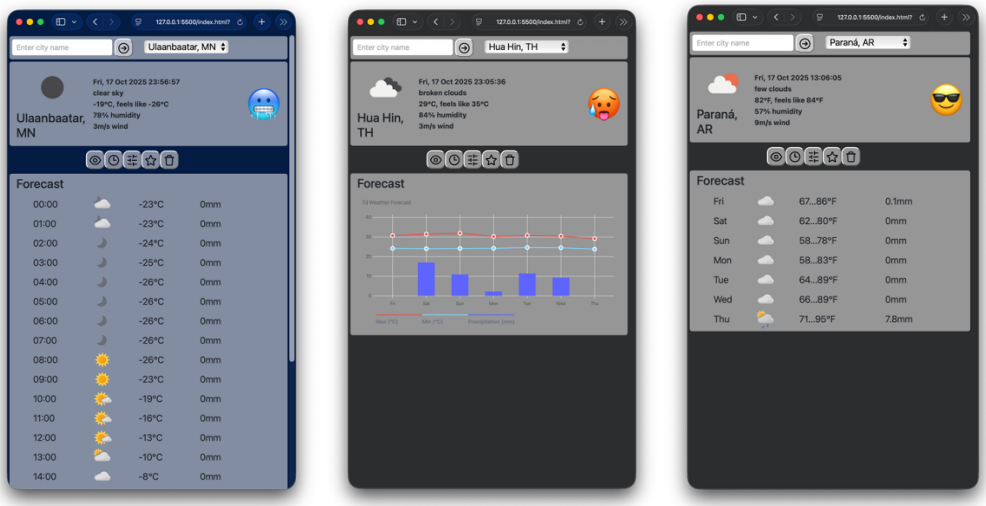
There is a basic implementation of rule-based styling, which changes the app background colour based on keywords in the data and the time of the day. The key-colour pairs are in a separate file (styleMap.js) for easier editing. For more sensitivity, the mapping could be based on the description instead of the one-word condition. Similarly, the persona faces are set in the style map, using emoji codes. The temperature limits are subjective and ideally could be set by the user in a later iteration.

Expected points for the project

The app is user-friendly and meets the minimum requirements apart from using three different data providers. Additionally, there is a caching logic which improves efficiency by avoiding too frequent API calls. I would expect to get 25 points.

Tools and declaration of AI usage

I've coded the experience in Visual Studio Code, tested it with live server on Safari and Chrome on desktop and iPhone and used Github for version control.
I have not used AI in coding or writing the report.



Screenshots of the app.