Weather Wizard

Technical Documentation

Author: Senali Perera

Version Number: v1.0

Date: 08/12/2023

Introduction

Project Overview

Weather Wizard is a software solution aimed at unveiling real-time weather information for the users. The software will fetch weather conditions, temperatures, wind speeds. Application is designed to get current weather information, daily weather information as well as it helps the users to keep track of hourly temperature fluctuations over week.

Purpose of the document

Purpose of this technical documentation is to give an in-depth idea of how the software solution is designed, implemented and tested in all the ends.

Objectives

The objectives of this project are to display following real-time information.

- Current weather information.
- Daily weather information.
- Hourly weather information.

Scope

In Scope

- The timezone should be fetched automatically and no input should be recorded from the user.
- Users should be able to switch between Fahrenheit and Celsius.
- Users should be able to reload the application with a button.

The following information must be displayed in the application

Current weather conditions

- Displaying current temperature
- Displaying current status of weather
- Displaying an icon of current weather status (Day/Night conditions should be considered)
- Displaying current wind speed
- Displaying last updated date and time

Daily weather conditions

- Displaying daily weather state with an icon

- Displaying daily minimum temperature
- Displaying daily maximum temperature

Hourly weather conditions

- Displaying a temperature fluctuation chart indicating temperatures of every hour for the week

Out Scope

- The application should be automatically updated once weather conditions change.
- Push notifications should be sent in severe weather conditions.
- Any other metrics like humidity, air pressure, etc. will be displayed.
- The application allows users to search for locations and view their weather conditions.

Assumptions

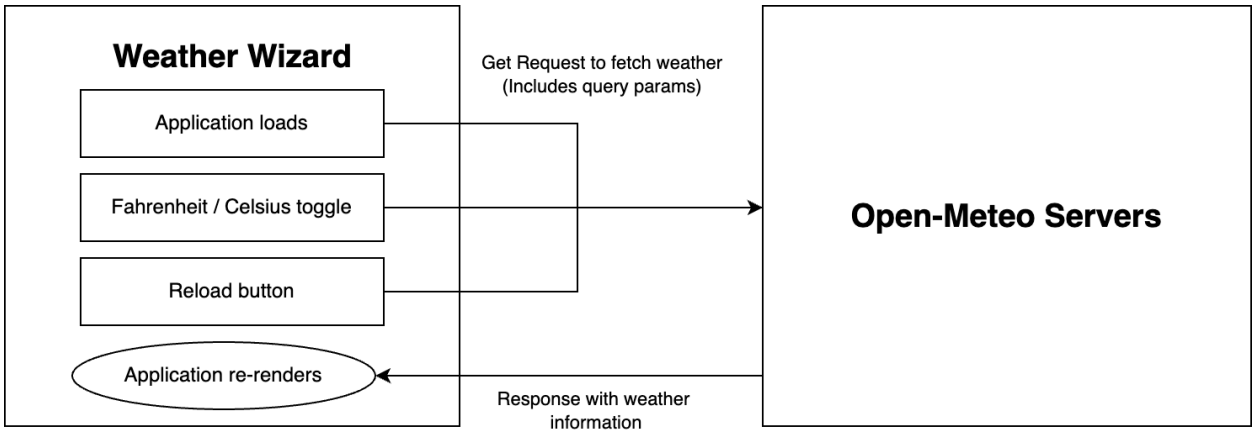
- The application uses open-meteo API for fetching weather information. The application assumes that the information is updated, reliable and correct.

Risks

- The learning curve of React.js and related testing is quite big. It might impact the delivery times.

Product Overview

High level architecture



Requirement Analysis

Functional Requirements

FR-01: Displaying current weather information

Requirement	The user should be able to view current weather information.

Acceptance Criteria	- The location information should be fetched in the first application launch. - Users should be able to view the current temperature. - Users should be able to view current wind speed. - Users should be able to view current weather status (Cloudy, Rain, etc.) - The current weather container should display an appropriate icon for weather status. - The displaying icon should change considering the day or nighttime.

FR-02: Displaying daily weather information

Requirement	The user should be able to view daily weather information for the whole week.

Acceptance Criteria	- There should be 7 day view cards starting from “Today”. - Each day card should contain minimum and maximum temperatures for that day. - Each day card should contain weather status for that day along with the icon.

FR-03: Displaying hourly weather information

Requirement	The user should be able to view hourly weather information.

Acceptance Criteria	- A chart should be displayed to the user with temperature variations. - The chart should be plotted for every hour starting from today for 7 days ahead.

FR-04: Switching temperature units

Requirement	The user should be able to switch between Celsius and Fahrenheit.

Acceptance Criteria	- A toggle should be there for the user to switch between C and F. - All the temperature metrics should switch units once the button is clicked.

FR-05: Refresh button functionality

Requirement	The user should be able to refresh the page on button click.

Non-Functional Requirements

NFR-01: Responsive design

Requirement	- The user interface should be responsive.

Acceptance Criteria	- The UI adapts to the viewer's screen size. - The font sizes should adapt to screen height. - Images should adapt to screen size. - The UI should support mobile views. - The daily view should be horizontally scrollable in mobile view.

NFR-02: Usability

Requirement	The application should be user friendly so that people can use the application with a small amount of computer literacy.

Acceptance Criteria	- The required information should be clearly displayed. - The error messages should be descriptive. - The UI should be simple and not overcomplicated. - Proper loading screens must be displayed on time taking processes.

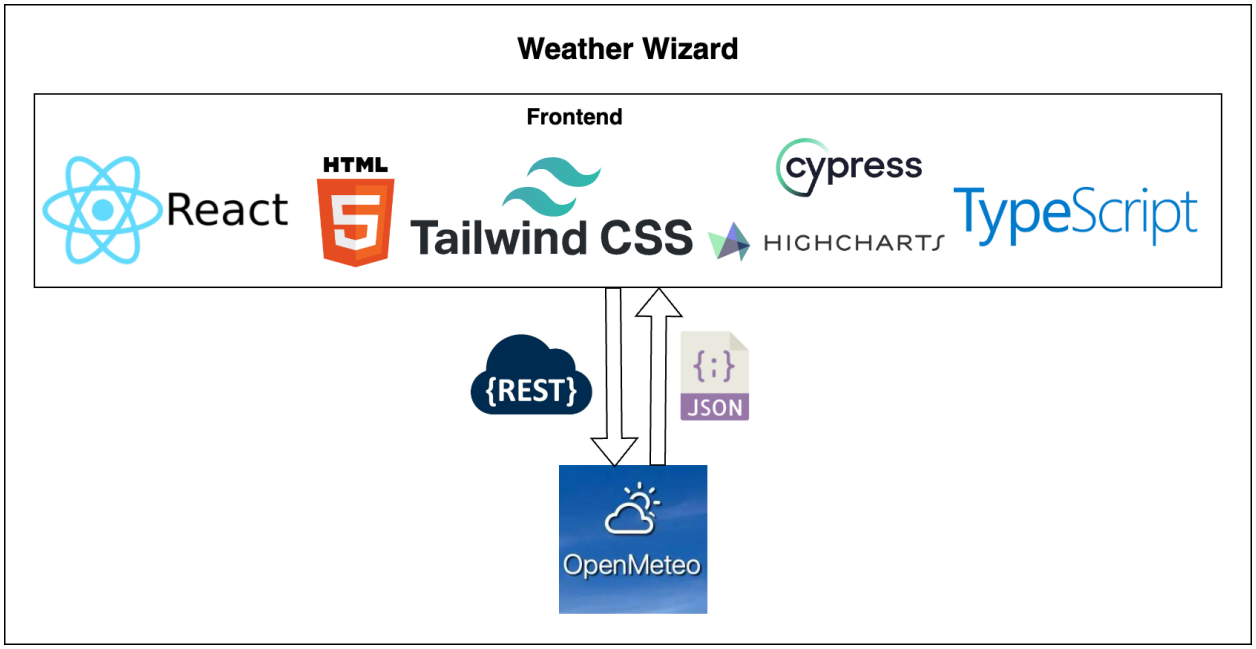
NFR-03: Availability

Requirement	The application should always be available.

Acceptance Criteria	- Proper exception handling must be done to avoid crashing.

Implementation

Technology Stack



Prerequisites

In order to run the development version of this software solution the following prerequisites are needed.

- Node version 18
- npm version 9.5 or above
- Google chrome

Implementation

Open-Meteo API

This application uses open-meteo weather API to fetch weather information.
API documentation: https://open-meteo.com/en/docs

Following parameters are passed from client side to the server from the GET request.

Parameter	Description
longitude	Longitude fetched from browser API is sent over to the server as a query parameter to determine the client's location.
latitude	Latitude fetched from browser API is sent over to the server as a query parameter to determine the client's location.
temperature_unit	Celsius or Fahrenheit is passed from the client side.

Following parameters are included in the response.

Current Weather

Parameter	Description
temperature_2m	Current temperature
weather_code	Current weather status code
wind_speed_10m	Current wind speed

Daily Weather

Parameter	Description
temperature_2m_max	Maximum possible temperature
temperature_2m_min	Minimum possible temperature
weather_code	Weather status code

Hourly Weather

Parameter	Description
temperature_2m	Temperature

CurrentWeather Component

Component Name	CurrentWeather
Props	currentWeather: currentWeatherType currentUnits: currentUnitsType
Implementation	- Formats dates. - Get weather information from the weather code. - Displays current weather information.

DailyWeather Component

Component Name	DailyWeather
Props	dailyWeather: dailyWeatherType dailyUnits: dailyUnitsType
Implementation	- Creates a new map with weather information for easy data access. - Iterates through the new map and renders a list of DayCards.

DayCard Component

Component Name	DayCard
Props	weather: DayWeatherType
Implementation	- Formats the date. - Gets weather information from the weather code. - Displays information of a day.

HourlyWeather Component

Component Name	HourlyWeather
Props	hourlyWeather: HourlyWeatherType hourlyUnits: HourlyUnitsType

Implementation	- This holds the WeatherChart component inside.

WeatherChart Component

Component Name	WeatherChart
Props	hourlyWeather: HourlyWeatherType hourlyUnits: HourlyUnitsType
Implementation	- Formats dates. - Feeds data to the highcharts component to draw the chart.

SwitchUnitsButton Component

Component Name	SwitchUnitsButton
Props	onToggle: Function
Implementation	- Styles a checkbox component to look like a toggle. - Executes onToggle function on click.

useComponentWidth Hook

Component Name	useComponentWidth
Props	onToggle: Function
Implementation	- This custom hook is used to measure width and height of a DOM element.

LoadingScreen Component

Component Name	LoadingScreen
Props	-
Implementation	- Displays a loading screen.

Testing and Quality Assurance

Unit testing

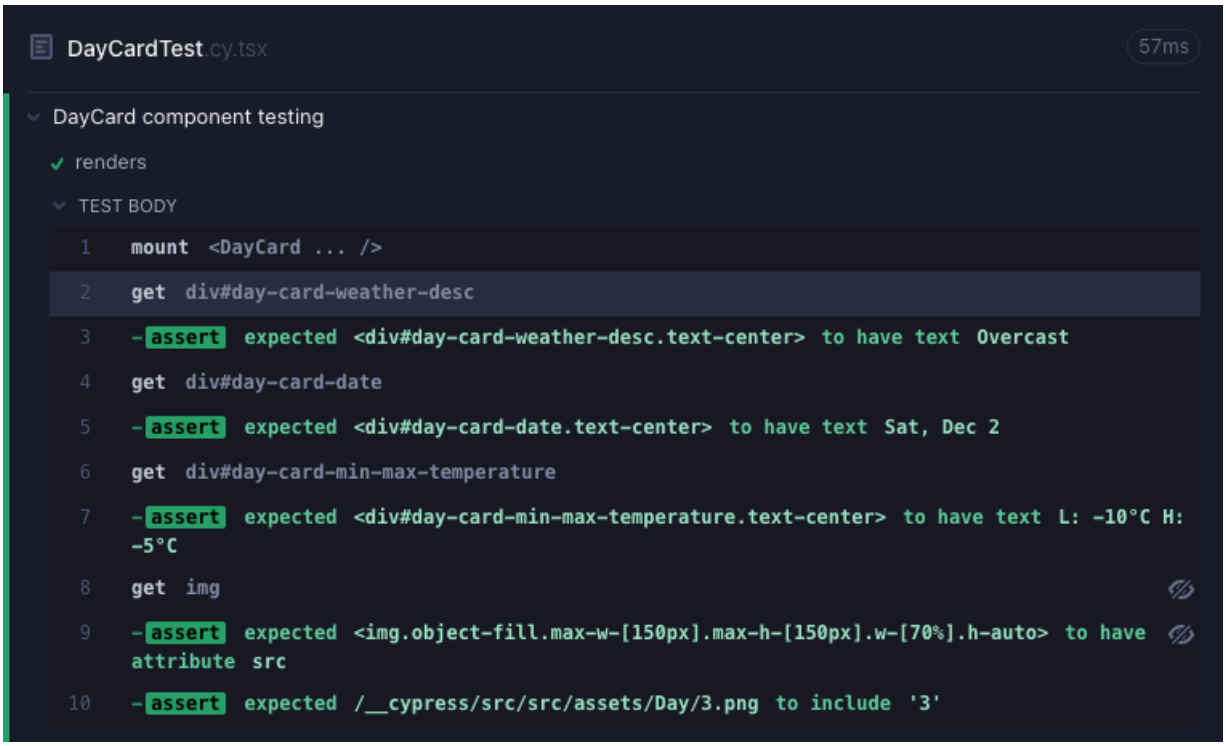
Unit testing has been done with cypress for the frontend. All the components are well tested and following are the test results.

Current Weather Test

Action	CurrentWeather component test
Expected	CurrentWeather Component to be rendered with weather information.
Result	CurrentWeather component rendered with temperature, weather status, image, wind speed and last updated time.
Status	**PASS**

Day Card Test

Action	Day Card component test
Expected	Day Card component to be rendered with weather information.
Result	Day Card component rendered with minimum temperature, maximum temperature, weather status, image and date information.
Status	**PASS**



Daily Weather Test

Action	DailyWeather component test
Expected	DailyWeather component to be rendered with 7 Day Cards.
Result	Day Card component rendered with 7 Day Cards with each day's weather information
Status	**PASS**

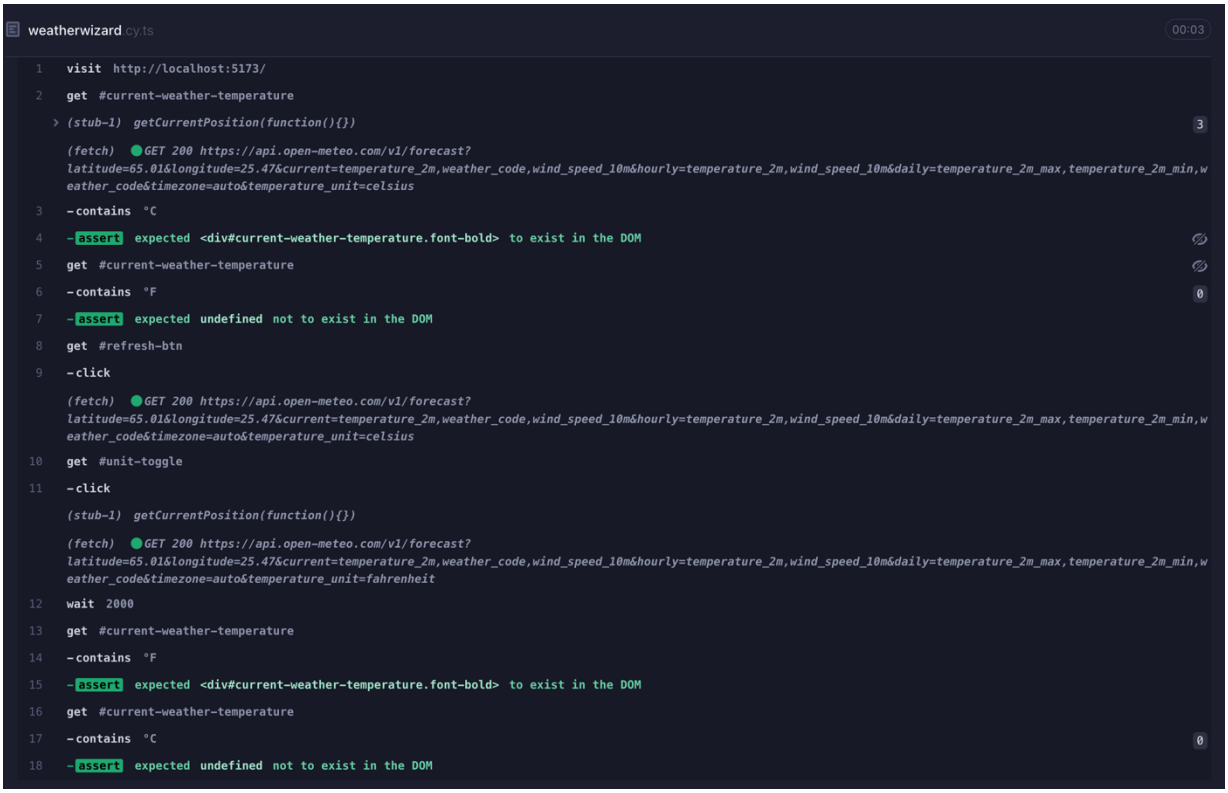
Hourly Weather Test

Action	HourlyWeather component test.
Expected	HourlyWeather component to be rendered with the chart.
Result	HourlyWeather component rendered with the chart.
Status	**PASS**

End to end testing

End to end testing of the whole flow of the application was tested with cypress. Following are the results.

Preconditions	-
Steps	- Visit application with url. - Grants permission for geolocation. - Clicks on the reload button. - Check if the unit is in Celsius. - Check if Fahrenheit is not included. - Click on change units toggle. - Wait for 2 seconds till the page rerenders. - Check if the units have changed to Fahrenheit from Celsius.
Expected Result	All the test cases to pass in one flow.
Result	**PASS**



Manual QA Testing

Manual testing has been done under following scenarios.

Scenario 1: Requesting location permission

Preconditions	-
Steps	- User navigates into Weather Wizard
Expected Result	The browser should request for the user's permission to fetch location data. A loading screen should be displayed till the data gets loaded.
Result	**PASS**

Scenario 2: UI testing

Preconditions	The user successfully grants location permission
Steps	- User navigates into Weather Wizard
Expected Result	The current weather should be displayed. The daily weather should be displayed for 7 days. The hourly weather chart should be displayed
Result	**PASS**

Scenario 3: Reload testing

Preconditions	The user successfully grants location permission
Steps	- User navigates into Weather Wizard - Current weather is displayed - User clicks on the reload button
Expected Result	The weather and last updated time should be updated to the latest.
Result	**PASS**

Scenario 4: Temperature unit change testing

Preconditions	The user successfully grants location permission
Steps	- User navigates into Weather Wizard - Current weather is displayed - User clicks on temperature unit toggle
Expected Result	The weather should be updated to °C / °F
Result	**PASS**

Challenges faced

A significant obstacle was faced during unit testing. The unit testing was intended to be done with Jest, but there were many troubles in Jest integration due the project being a TypeScript project and Vite based project.

After multiple tries, the technology stack for unit testing was changed to cypress allowing both the unit testing and end to end testing to be done with the same framework. This change was a remarkable change in the testing sector and worked really well for this application. The live view of the testing components and the flow of the program helped to keep track of the testing better than Jest.

Future implementation

- Support for a native mobile app.
- Internationalisation support.
- Support adding multiple locations to keep track of weather.
- Support for more metrics like pressure, humidity, etc….

Conclusion

The course of this project is to develop a web application to retrieve weather information from the open-meteo API and display it for the end users. The documentation covers all the details from the project planning to the implementation and testing. The application successfully displays weather information under 3 categories (current, daily, weekly) and provides accurate weather updates with the help of open-meteo API.

The future implementation section will be referred to and completed during the next revision of the Weather Wizard application.