

# Project report

## Introduction to Web Programming

**Student:** Martin Bárta  
**ID:** 002525033

### Weather app

This app was created with most of the features listed in the project work instructions. Only one feature - Favourites tab was not implemented. There are two HTML files, two JavaScript files, one CSS file and an assets directory with images used as image backgrounds on different weather events, current temperatures etc.

Both HTML files are almost identical with basic head declarations and links. The body is divided into 3 main parts. The first part is just the text with a header and paragraph to instruct the user of what the application is about. Second part is a weather form, where the user inputs the location he is looking for. There is of course a submit button to submit the choice and also other options underneath - button for using user's current location and the unit selection, where the user can switch between different temperature units. The last part of the body is the container with more dividers, that will be populated with fetched data after user confirms the location. This container is initially hidden using CSS so it doesn't bother the user when there are no data at first. There is also a navigation divider on the bottom of the file which shows the user where the data he is looking at are from and he can also switch to a different API provider if he wants. The secondapi.html page also contains a chart, where can be seen how can the temperatures from different providers differ.

The CSS file is used of course by both HTML files and there are also media queries for different screen sizes at the bottom. The main thing set up there is the grid size for different screen widths. Since the page shows 24 items for the upcoming forecast, the grid is divided into 3 rows if the screen is wide enough, 4 rows after, 6 rows after that and then 12 rows for mobile devices (2 items per row). Other necessary CSS rules have been applied one by one to make the page look somewhat user friendly and good looking.

The Javascript files are both similar as well, because they use mostly the same logic. The first JS file (index.js) is using two (actually three, but the weather data are just from two) APIs for fetching necessary weather data. The first API used is [OpenWeatherMap.org](https://openweathermap.org/) which is used for providing the current weather at given location. This API uses API key which needed to be obtained from the site and is limited in terms of forecasts and forecast frequency (in free version). The second API had to be used because of that, namely Open-Meteo.com. This API doesn't require API keys for API calls and provides 24h, hourly forecast as well as 7-day forecast. The third API used was WeatherAPI.com just for getting the local time of given location, because even though these previous APIs provide some values, I wasn't able to get the correct time. WeatherAPI on the other hand has a dedicated value for local time and date and was easier to work with. In the JS file there are EventListeners for buttons used, which call on other functions. There is also a function to convert the coordinates from my current position to a location so other functions can be called in the same manner as when location is given by the input field. The main function is the getWeather function which calls the other necessary functions to display the fetched and transformed data to the screen. The displayWeather function is used to show the current weather and add the HTML code to the page. The function updateTheme is used to modify the page looks for different defined circumstances like rain, temperature (low, medium, high), or night. The

function `getForecasts` is getting data from OpenMeteo API and calling functions to transform and display data to the user. The two following functions are used to display those data. Last two functions are used as tools, first one for mapping the weather codes to according icons and the second for getting the local time, like described earlier. The `secondapi.js` file is very similar, because it is doing basically the same job, just working with different data and sometimes different logic, due to those differences. The [WeatherAPI.com](https://weatherapi.com) has been used with API key that had to be obtained from the site, but with minimal limitations. In this file the chart is displayed to show the temperature differences of the two providers. For this reason, modified function from `index.js` has been used to push hourly data to an array. This array with the second array of hourly temperatures from WeatherAPI is used as datasets and `Frappe` is then used to create the chart itself.

## AI Declaration

No AI tool was used in writing this project report and in case the AI declaration is regarding the coding itself - only one AI system, namely ChatGPT, was used in creation of the weather app. It has been used mainly as a documentation substitute, a quick way to find out what methods, functionalities and other syntaxes exist and how can they be used with examples. It has been also used as a consultation tool when planning and improving the app - for example what might be some issues I haven't thought of or what would be some recommendations for the layout, structure etc. It has also been used to consult, check or come up with different parts of the app logic. Lastly it has been also used as a debugging tool, because it provides quick bug and typo check, which could sometimes save a lot of time.

## Points

For this project I would like to be awarded 38 points, since I believe I have fulfilled all the point requirements but one feature. There are also only 39 points maximum if my math is correct, but I suppose that is just a typo. I also think my app has some minor features which weren't necessary (like manipulating local time) that could make up for some minor deficiencies.

- ✓ Well written PDF Report (3 pts)
- ✓ Application is responsive and can be used on both desktop and mobile environment (4 pts)
- ✓ Application works on Firefox, Safari, Edge and Chrome (3 pts)
- ✓ The application has clear directory structure and everything is organised well (2pts)
- ✓ User can search for locations (1 pt)
- ✓ User can use his/her location GPS-coordinates (Geolocation API) (2 pts)
- ✓ At least two data/forecast providers are used (3 pts)
- ✓ At least three data/forecast providers are used (2 pts)
- ✓ User sees the current weather at a specific location (1 pt)
- ✓ User sees the forecast for the next 24 hour, hourly based (3 pts)
- ✓ User sees the forecast for the next 7 days (3 pts)
- ✓ All the weather forecast elements uses icons for e.g. sunny and cloudy weathers (3 pts)
- ✓ The look and feel of the application reflects the current weather (2 pts)
- ✓ User sees simultaneously two forecast in a graph (3 pts)
- ✗ User has the option to tag some locations as her favorites (2 pts)
- ✓ User has an option to switch between celsius and fahrenheit degrees and kelvins (2 pts)