

Documentation for CT30A2910 Introduction to Web Programming - Project Tomi Vilpponen 0614965

AI declaration: No AI was used in the making of this documentation. AI was only used for debugging purposes in the project if I couldn't find the bug myself. The AI tool used was ChatGPT.

I decided that the most interesting and fun project out of the 3 listed was the weather application. I wanted the application to be simple but to look good as well and provide enough information to the user. I started by looking at different weather api providers that were free. First I found open-meteo.com but I noticed that you couldn't search by city but with coordinates. I decided to use this website as well but later after I figured out a solution to get the users coordinates. It took a while to find an api that was going to fit my needs but I found a website called weatherapi.com. It provided me with a possibility to search with the city's name rather than trying to convert the user's search to coordinates and searching with that. After this I created an account and tested the api and sure enough, it worked perfectly.

I first added a search bar where a user can search a location and then the api would provide me with the data. Weatherapi's free version only provides 3 days of forecast but it was fine as I planned on using open-meteo for the 7 day forecast. I noticed that the data from weatherapi had the coordinates of the location and icons for the weather provided. This was perfect for me. So I created a container where I would input the current weather data. This included the city's name, region, country, date and time, icon of the conditions, temperature and conditions. Then I did the unit conversion so that the user can see the temperatures in celsius, fahrenheit and kelvin. I used radio buttons for this and an event listener for the scenario where the user wants to see the already searched location in different units. Then after this was working, I used frappe charts to create a chart that showed the hourly forecast. After this I used the geolocation api to give the user a possibility to search weather from their location directly. I added a button for showing the users locations weather and modified the url for the api to use coordinates for the search instead of the city name. After this was working I did the weekly temperature container that showed 7 days of min and max temperatures and weather condition for the searched location. I used open-meteos api by fetching the data from weatherapi and using its coordinates and using them in the url. Then I used a simple for loop to create the 7 day's data. I also decided to modify the frappe-chart so it shows both apis hourly forecasts.

After the functional parts were done, I decided to modify the css so that the application looks nicer. I added a feature where the look and feel of the application changes depending on the weather and time of day. So when the temperature is cold, the text will be blue, and when it's hot, the text is red etc. I also made it so that the background color is dark gray when the local time is after 6pm-6am and white when the local time is 6am-6pm. I also made the frappe chart lines to change color depending on the day's average temperature. I decided to also use a bootstrap for the navbar to make it look nicer. Then I made a simple intro to the website with an image and text. At the end I noticed that openweathermap.org had a temperature map tileLayer I could use with Leaflet. So I decided to implement a map that zoomed in on the searched

location and had a temperature map from openweathermap. I also used openstreetmap as the world map. I also added descriptions to the weather elements on what api was used

Points requested:

a well written report: 3 points

Application is responsive and can be used on both desktop and mobile environment: **4 points**

Application works on Firefox, Safari, Edge and Chrome: **3 points**

The application has clear directory structure and everything is organized well: **2 points**

User can search for locations: **1 point**

User can use his/her location GPS-coordinates (Geolocation API): **2 points**

At least two data/forecast providers are used: **3 points (weatherapi, open-meteo)**

At least three data/forecast providers are used: **2 points (weatherapi, open-meteo, openweathermap)**

User sees the current weather at a specific location: **1 point**

User sees the forecast for the next 24 hour, hourly based: **3 points**

User sees the forecast for the next 7 days: **3 points**

All the weather forecast elements uses icons (and numbers) for e.g. sunny and cloudy weathers: **3 points (current weather has an icon and weekly has descriptions)**

The look and feel of the application reflects the current weather (e.g. it is blueish, when it is cold; reddish, when it is hot; dark, when it is night...): **2 points (this is described more in the documentation)**

User sees simultaneously two forecast in a graph, e.g. there is temperature forecast for the next 24 hours and there are two lines telling how the data sources are providing (a bit) different data: **3 points (hourly forecast has 2 lines)**

User has an option to switch between celsius and fahrenheit degrees and kelvins: **3 points**

User sees a temperature map of the world and it is zoomed on the searched location: **3 points**

TOTAL POINTS REQUESTED: 38 points