Ivan Banchev
Student ID: 24372449

Project Report - Reactive Internet Application

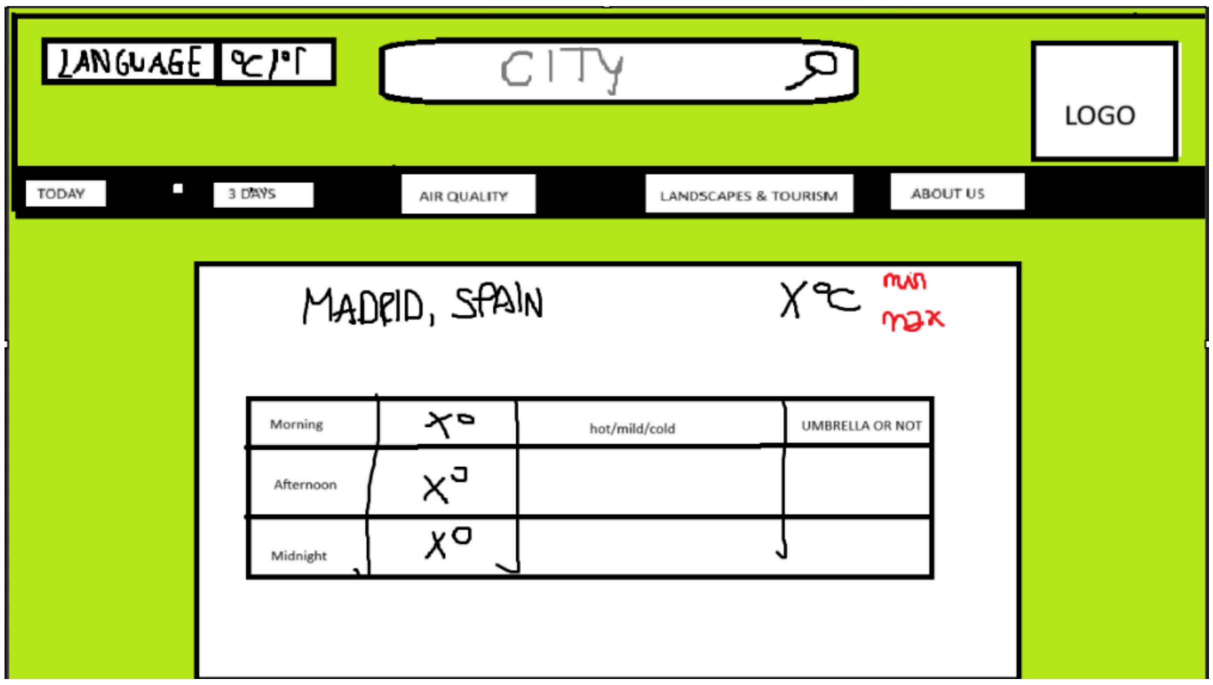
Introduction

1. Objectives : As a starting point, it was necessary to understand all the requirements that my application had.
   - 1.1. First of all, I recognised and analysed the functional requirements:
     - 1.1.1. Implementing the 3rd party API from OpenWeatherMap on the server-side to get the weather forecast in the following 3 days for the desired city..
     - 1.1.2. Displaying the weather information for a city on the client-side based on the input provided by the user.
       - 1.1.2.1. If the input is wrong, then an error message is displayed.
       - 1.1.2.2. Otherwise, the system will tell if the user should bring an umbrella based on the rain forecast, the type of weather (Cold, Mild, or Hot) and a summary table for the next 3 days in which it will show the temperature, the wind speed and the rainfall level.
     - 1.1.3. Additionally, using the same API from OpenWeatherMap, it will display the longitude and latitude of the city. From this data, we can get detailed information about the different air pollutants and categorise them in case some of them are potentially dangerous for your health.
     - 1.1.4. As a new innovative feature for my application, when the user inputs the desired city, it will automatically display the most famous landscapes and touristic attractions. This will be implemented by using Generative AI by adding the API from OpenAI into my server-side and displaying the information into the client-side.

   - 1.2. Secondly, I recognised the non-functional requirements (**Visual Impact**):
     - 1.2.1. For the user interface (UI), I'll use HTML, CSS for the static structure of the web application and JavaScript to add some dynamic features.
     - 1.2.2. Since the style of the code increases readability, I'll choose appropriate names for variables that can properly describe the functionality of that variable and a similar concept will be applied for the names of the different project files.

2. First approach : To start implementing the different features of my Web Application, I first needed to have a clear structure or schema of what my website should look like, so I started designing a sketch.

Ivan Banchev
Student ID: 24372449



Being that the first prototype of my Web Application **(index.html)**, I started creating the Document Object Model (DOM) of the web application, in which I created all the HTML elements that are involved in the main page, such as titles, headers, body, text elements, etc…

3. Evolution : After creating the DOM, I started implementing the functionality of the search bar, which required the OpenWeatherMap API. At this point, problems started to arise, since I had never used an API before, so I had to deal with it. After reading the documentation and watching several tutorials, I was able to do the proper request to the API and fetch it in order to get the information and process it with my script. Then I implemented the "Today button", which shows a summary of the different weather features (pressure, humidity, etc…). Moreover, I had the idea to display the 24h temperature forecast but finally this wasn't implemented due to the time limitations and the API.

   After spending several hours, I realised that I had to use Vue.js, which was absolutely new for me and I had to change many things of my code in order to adapt the code. This issue delayed the whole project a few days, since I had to read the Vue.js documentation to be able to understand how to work with this framework.

   After having my code adapted to Vue.js and having the most basic notions about how to deal with its syntax, I started implementing the rest of functionalities to each button. For the "3 DAYS" button, I had to make another API call, in this case, the forecast call, in which I had to input the city and the public key. Once the API url was fetched, I had to process the information. Now there was a slight difference from the "Today" button, because in this approach, I had to understand how to treat the data from different days. I classified them by day, so the data was correctly grouped. In order to implement it into the HTML code, I had to use for the first time a v-for which was pretty unclear for me, but after struggling with it, I finally discovered how to use it. Once this feature was implemented, I wanted to proceed with the rest of features, but I realised that the “TODAY” button was incomplete, because I needed another API call which was the Geocoding API so that I could extract the latitude and longitude of the desired city.

Moving to the next button, “Air Quality” it was pretty easy to get the data once the API call was fetched. I just stored the data into my variables and it was straight forward. To determine whether the air was good or not, I took into account the AQI (Air Quality Index) and also the different values that act as a threshold between what is acceptable and what is dangerous for the living beings, displaying which pollutants were the ones that caused that the air was not suitable for us.

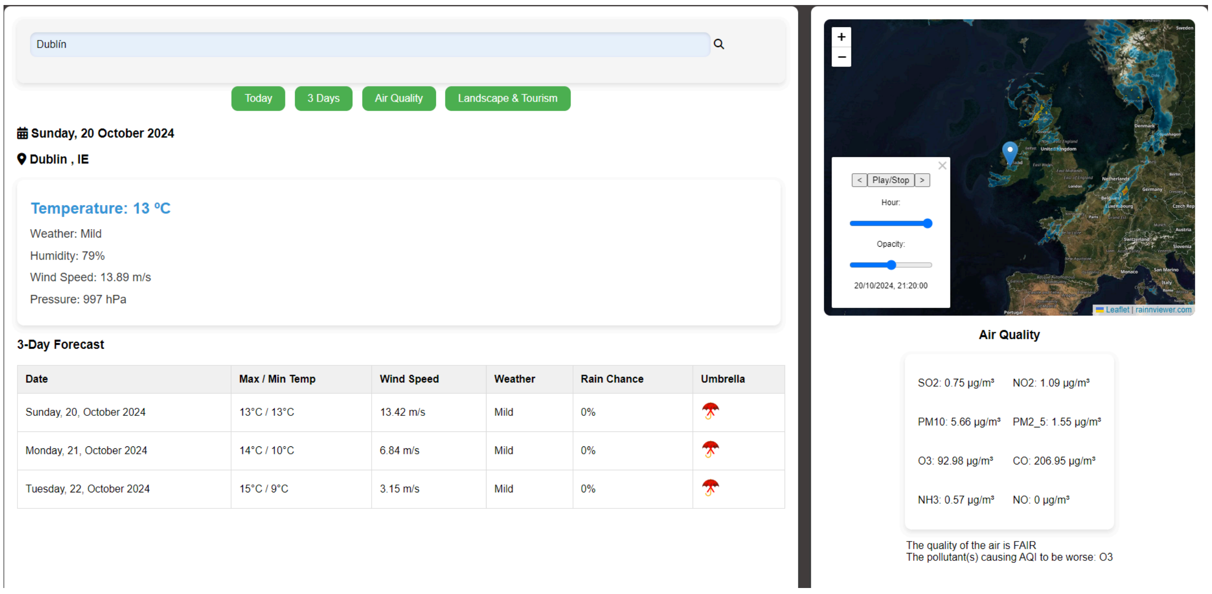
Once these three main buttons where implemented and working, I wanted to implement my innovative and amazing new feature which was the following idea: Since my web application’s audience were tourists, I decided to go a step further and try to give them the best user experience (UX) so that users could input the desired city to visit and the weather forecast would be displayed but they would also have the option to click on the “Landscape & Tourism” button and the AI would display the top 5 places to visit, giving a little bit of context for each landscape or attraction and showing a photo.

I was very excited about this idea, since I do really think that it’s useful and I haven’t found any other weather forecast apps that included this feature. Nevertheless, problems started arising. All of the AI that I could use for that end, weren’t free. After hours of researching I thought that I had found the perfect option, but it didn’t work.

Having this option partially discarded, I decided to introduce a map. This map would take your location and display your zone. There was also the possibility to zoom in and zoom out as desired. The cool thing about this was the precipitation view that it provided. When the user pressed the little icon at the bottom-left, an animation of the squalls would appear within 2 hours, so that the user could know if there were any rain probabilities in his zone. I tried to implement the OpenWeatherMap maps, but they didn’t work for me. The main idea was to have a button that, when pressed, would toggle the different maps that OpenWeatherMap provides, from the precipitation map, to the temperature map and the pressure map. Again, this feature was discarded because no free maps were available and easy to implement into my code.

While I was dealing with the implementation of the functionalities, I had to give some shape to the web application. Of course, I had almost no idea about HTML and CSS, which made me spend a lot of time reading the documentation and watching tutorials. The sketch helped me to have a clear understanding of what I wanted to do, but finally I decided to change the structure of the web application. While I was working on it, I changed my mind and did something different.

Ivan Banchev
Student ID: 24372449



Sunday, 20 October 2024

Dublin , IE

Temperature: 13 °C

Weather: Mild

Humidity: 79%

Wind Speed: 13.89 m/s

Pressure: 997 hPa

3-Day Forecast

Date	Max / Min Temp	Wind Speed	Weather	Rain Chance	Umbrella
Sunday, 20, October 2024	13°C / 13°C	13.42 m/s	Mild	0%	☂
Monday, 21, October 2024	14°C / 10°C	6.84 m/s	Mild	0%	☂
Tuesday, 22, October 2024	15°C / 9°C	3.15 m/s	Mild	0%	☂

Air Quality

SO2: 0.75 μg/m³	NO2: 1.09 μg/m³
PM10: 5.66 μg/m³	PM2_5: 1.55 μg/m³
O3: 92.98 μg/m³	CO: 206.95 μg/m³
NH3: 0.57 μg/m³	NO: 0 μg/m³

The quality of the air is FAIR
The pollutant(s) causing AQI to be worse: O3

Talking about server-side and client-side, I'd say that most of my application was developed around the client-side because of the requirements. I needed something fast, responsive and with a high user experience, and that's mainly in the client-side.

Conclusions

After spending more hours than I would have liked, I can say that this project was challenging for me. Starting from scratch and going into something that uses different new technologies such as Node.js or Vue.js wasn't easy but finally I achieved something.

Nevertheless, I would have liked to have more time to make something better in terms of functionality and in the visual part, but I can say that I have learned a lot.

Used technologies:

- Github
- VSCode
  - Vue.js
  - Node.js
  - JavaScript
  - HTML
  - CSS
- Third Party APIs

Ivan Banchev
Student ID: 24372449