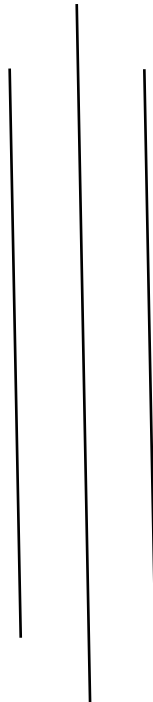


## **4CS017 – INTERNET SOFTWARE ARCHITECTURE**

### **Prototype 1**



**Nijmi Bajracharya**  
**ID: 2508912**

## **A short reflective summary**

### **Introduction**

In this prototype1, I have developed a weather app using basic knowledge from HTML, CSS and JavaScript by fetching the weather data with the help of Open Weather Map API. The project was interesting, and I had a great time learning and researching various resources. I was confident and had a clear vision of how I was going to do it.

With the access of Open Weather Map API, I have displayed the necessary weather information such as temperature, humidity, pressure, and many more.

### **Strengths**

- a. Simple and easy to debug- As the codes are done in HTML, CSS and JavaScript the codes can be easily identified and debugged.  
Also, the codes are easy to understand and learn.
- b. Speed- As Open Weather API is a client server, there is no delay in the initial page load.
- c. User-friendly- It is visually attractive with the help of CSS codes as it has icons which get access with links that make user easy to access and user-friendly.
- d. Real-time update- By using JavaScript it helps to fetch real time update with the help of Open Weather Map API.

### **Weaknesses and challenges**

- a. Lack of responsiveness- This app is not optimal for small screen devices such as mobile phones, tablets, etc. Hence, it lacks responsiveness.
- b. API dependent -As the Open Weather API is a third-party whether API means you are dependent on the provider's infrastructure, updates, and policies and if the service changes its terms, pricing, or ceases to operate, it could impact application.
- c. Basic error- The error alert message is very basic as just shows pop up on the screen and it can be improved by appealing various ideas and codes.
- d. Limited – As the codes are limited so it has limited features and if we have to add some new features, we need to rewrite the codes again.

### **Conclusion**

The weather app is a client-side web application with simple and efficient architecture to deliver real-time weather information to users. I have learnt how asynchronous JS works using fetch and also gained some knowledge of API. Hence, this architecture is well-suited for small to medium sized applications like the weather app, offering simplicity, ease of use, and flexibility.