HERALD COLLEGE KATHMANDU

UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

# Internet Software Architecture

## Individual Report

## ON

## WEATHER APP REPORT


Student Name: Aaryush Khatri

University Id: 2408485

Group: L4CG1

Module leader/Lecturer: Bishal Khadka

Tutor: Jenish Maharjan


Submitted on: 15/02/2023

# Contents

# 1.INTRODUCTION ON FINAL PROTOTYPE:

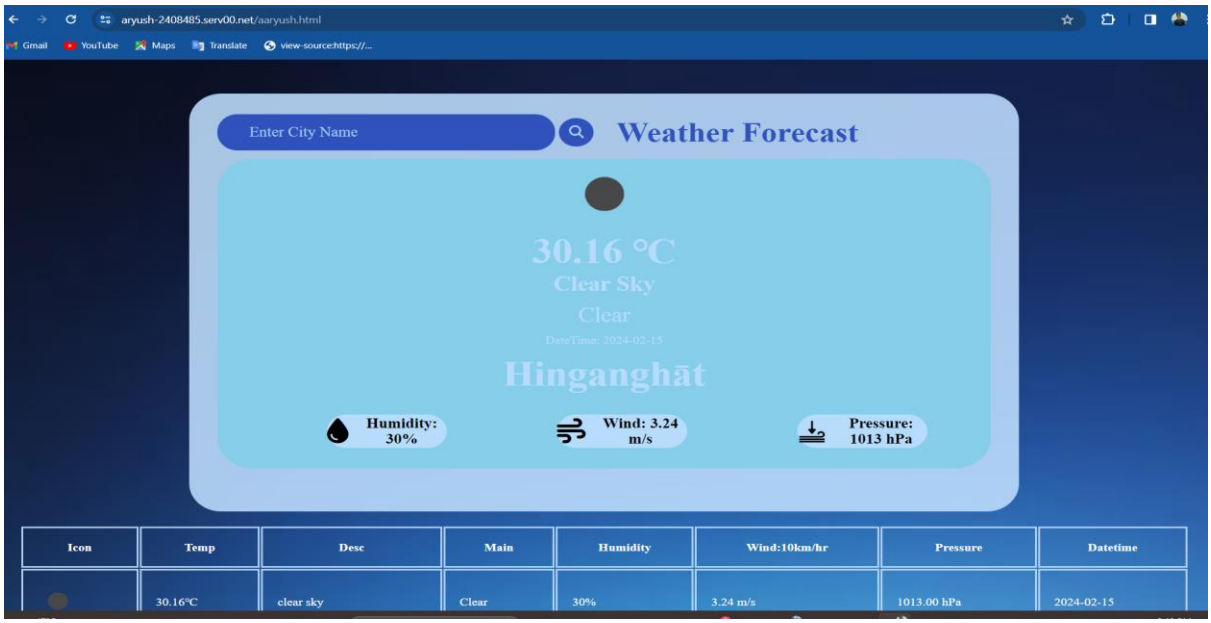
The weather application is a sophisticated web solution that smoothly combines HTML, CSS, JavaScript, and PHP technologies to provide up-to-date meteorological information. The HTML and CSS elements are specifically designed to facilitate user-friendly interaction, resulting in an intuitive and aesthetically pleasing interface. JavaScript improves the user experience by enabling interactive and adaptable elements. PHP enables server-side operations, enabling secure manipulation and storage of data. The application offers both real-time weather conditions and the ability to access previous data, ensuring a thorough picture of the weather. In addition, the use of API calls for instantaneous weather updates introduces a dynamic aspect to the user experience. The application is hosted online to improve accessibility and ensure global availability. This allows users to instantly receive precise and current meteorological information. The integration of these technologies and cloud-based hosting enhances the versatility and effectiveness of the weather application.An important benefit for the future advancement of the weather application might be the integration of machine learning algorithms to enhance the precision and customization of weather forecasts. Through the examination of past data and user preferences, machine learning models have the potential to provide customized predictions, enhancing the accuracy of the application in forecasting precise weather conditions for particular users.

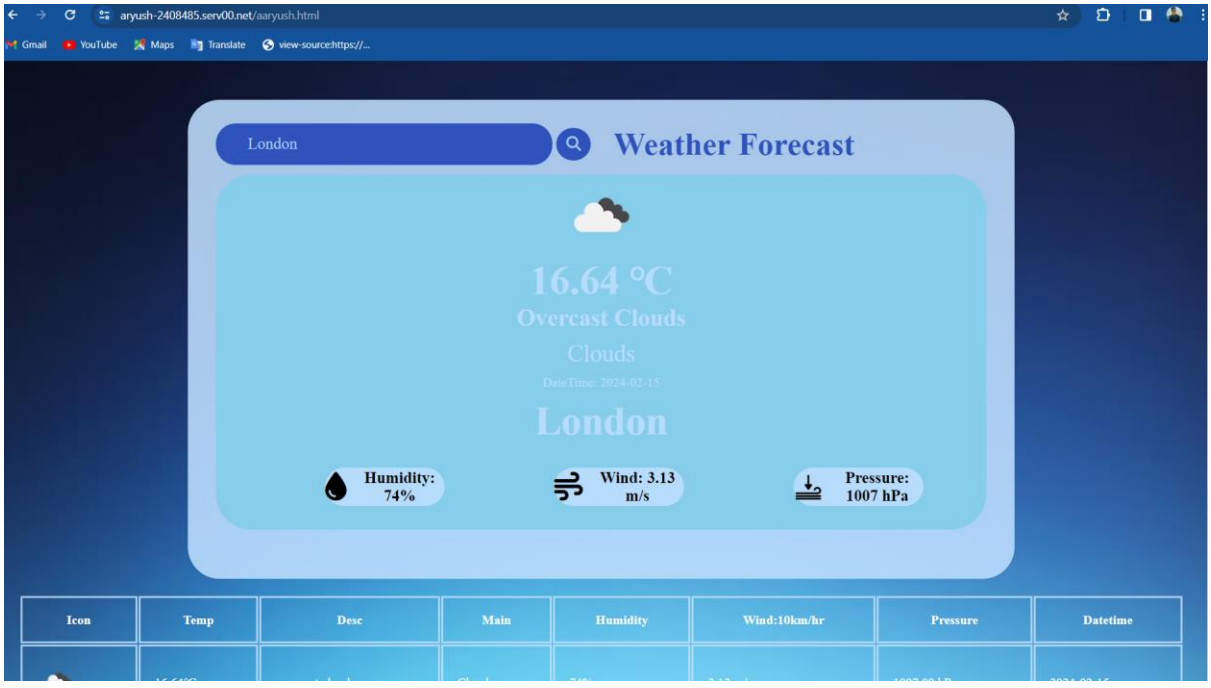
# 2.Positive Aspect And Negative Aspect:

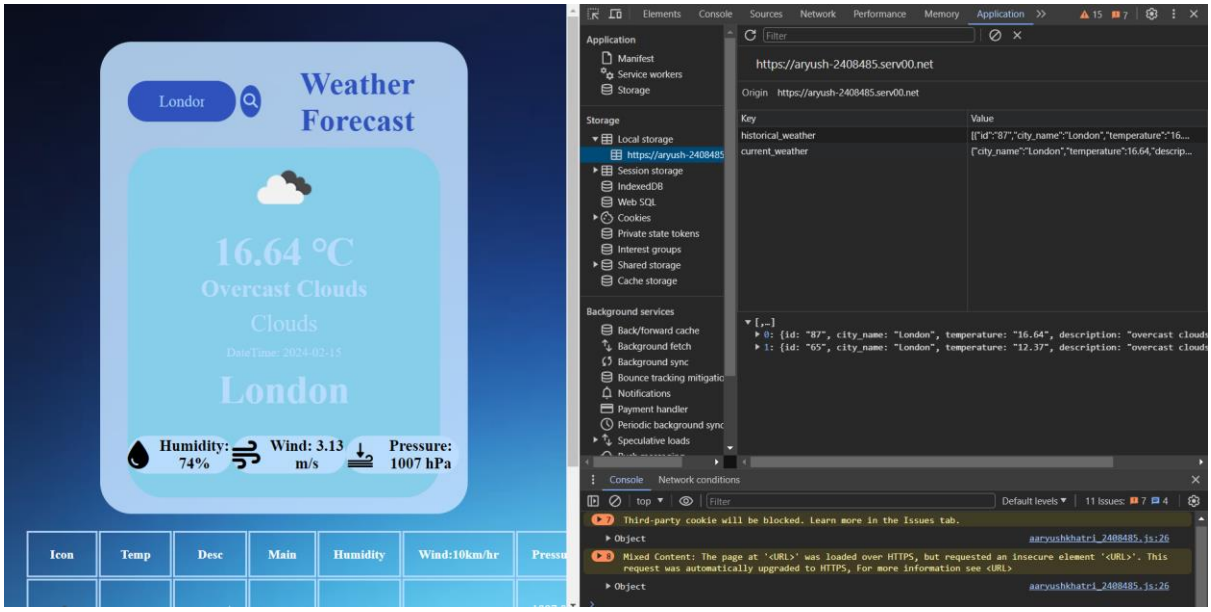
With a simple design, the document and website are orderly and easy to read. Real-time meteorological data for numerous cities provides a complete weather picture, including humidity, wind speed, and air pressure. The website shows the real-time date of the searched city to keep people updated on weather. Localhost weather icons enhance visual appeal. With parameterized queries to avoid SQL injection in database interactions, security is strong. Local servers eliminate the need to upload files to remote servers for testing, allowing developers to see local changes immediately. Using cURL instead of file_get_contents for API calls improves data retrieval security. Prepared statements in database queries defend against SQL injection, and sophisticated database connection error checks demonstrate a proactive approach to gracefully resolving difficulties. Making API keys constants simplifies key updates and improves code management. These positives create a trustworthy, safe, and user-friendly weather app.

The offered PHP code showcases commendable qualities in terms of document structure, real-time meteorological data presentation, and secure database interactions. However, there are several areas that necessitate correction. The code's inadequate error handling may hinder efficient debugging, and comprehensive validation of user input is necessary for enhanced security. The exposure of the API key within the code presents a potential security vulnerability, and migrating API queries via HTTPS would bolster the security of data transmission. To optimize and enhance security, it is necessary to resolve redundancy in historical data retrieval queries and eliminate hard-coded database credentials. It is advisable to include modularity guidelines into code organization to improve maintainability. In addition, the process of deploying the software to a live server may face difficulties, as local servers may not accurately duplicate the production environment. Dependence on the developer's computer to operate local servers presents possible complications, such as hindered development procedures caused by hardware breakdowns or other concerns. Attending to these problems will enhance the resilience, security, and efficiency of the weather application.

## 3.Screenshots of my hosted application

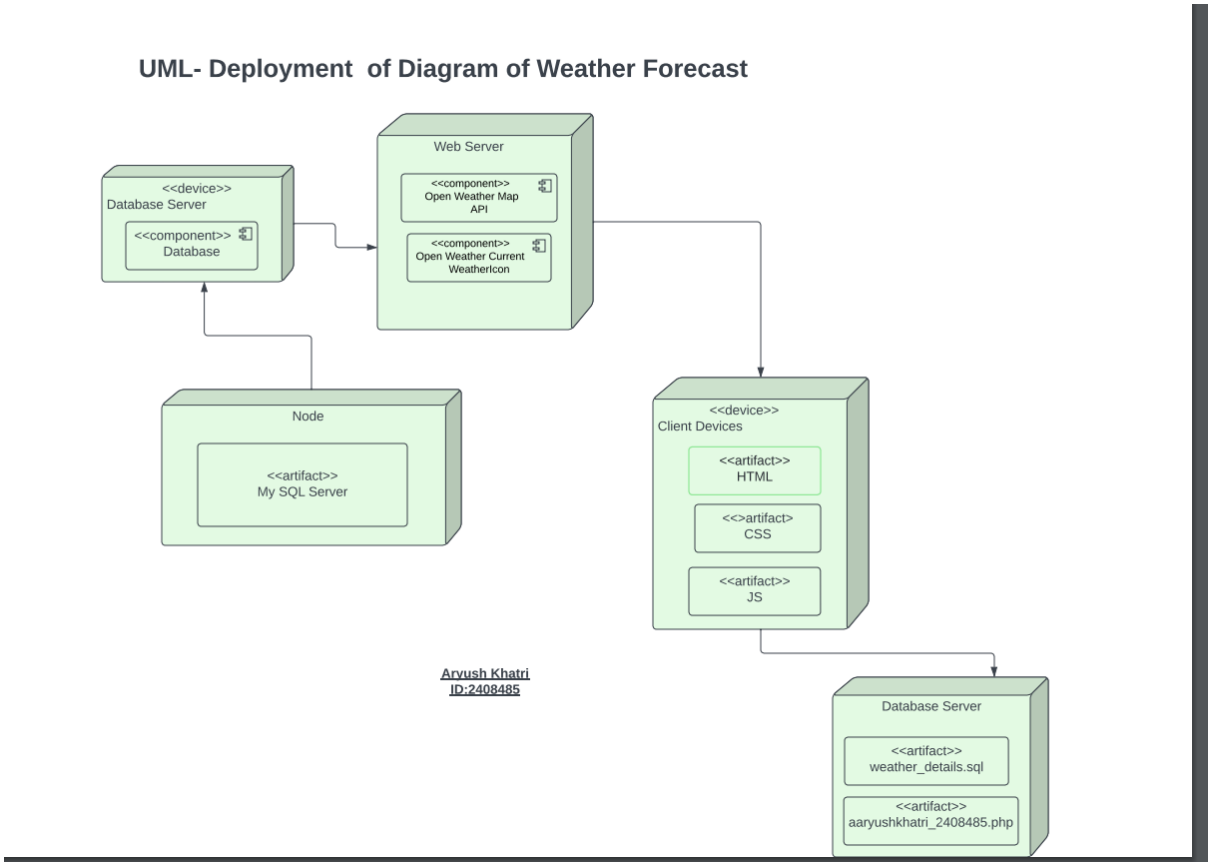


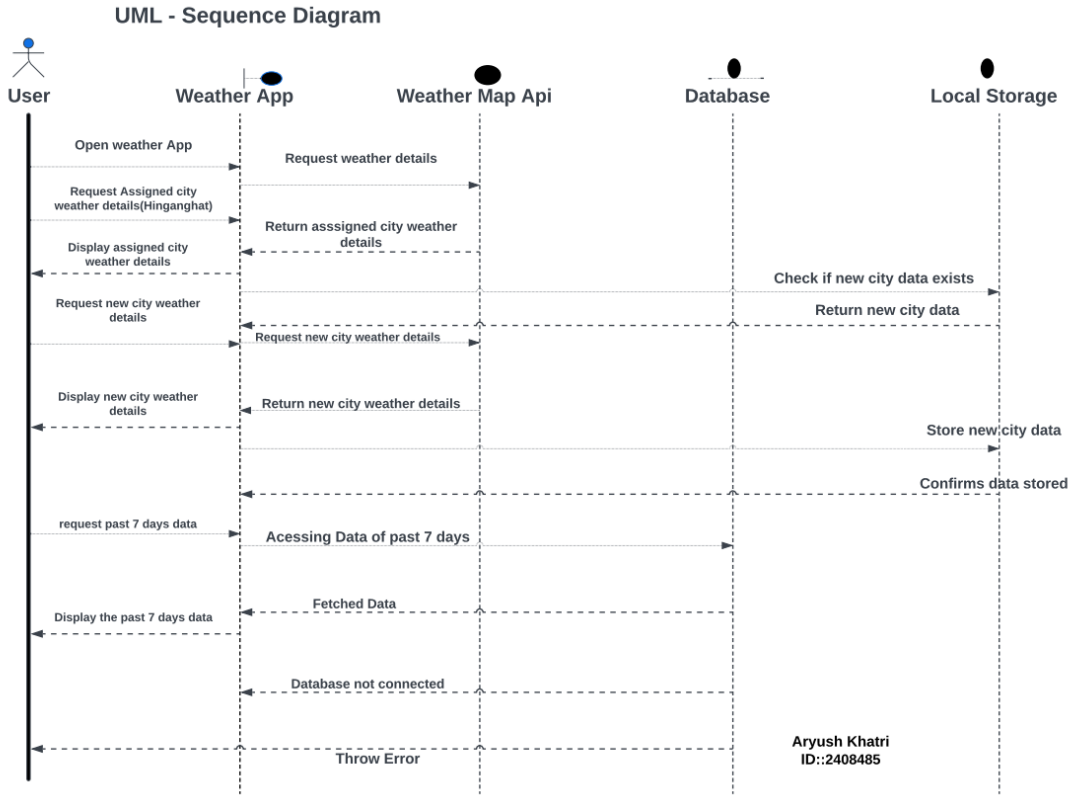





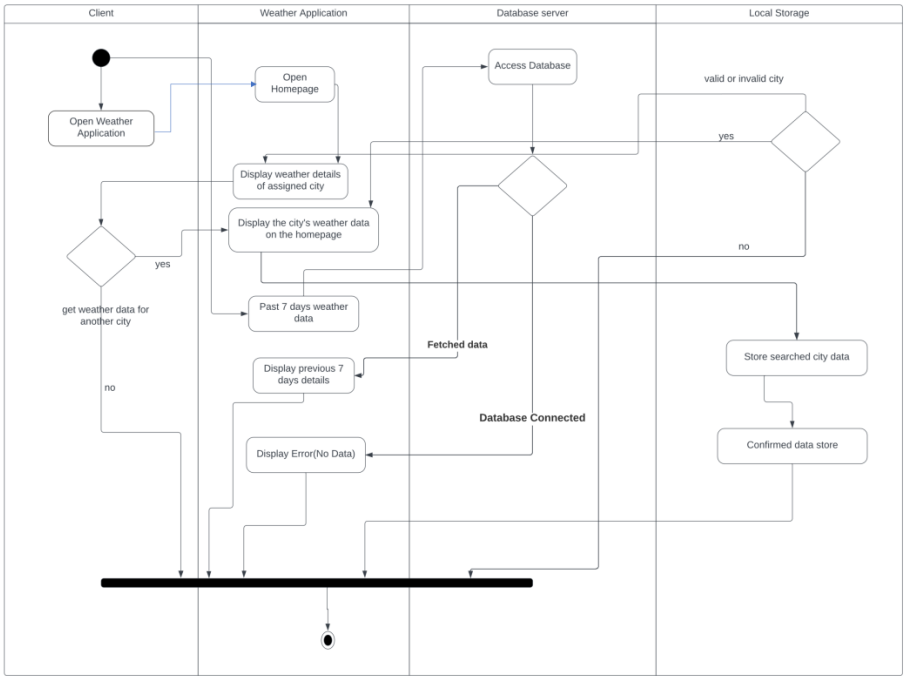

| Icon | Temp | Desc | Main | Humidity | Wind:10km/hr | Pressure | Datetime |
|---|---|---|---|---|---|---|---|
| | 30.16°C | clear sky | Clear | 30% | 3.24 m/s | 1013.00 hPa | 2024-02-15 |
| | 25.53°C | overcast clouds | Clouds | 28% | 3.73 m/s | 1014.00 hPa | 2024-02-14 |
| | 26.68°C | broken clouds | Clouds | 50% | 2.32 m/s | 1015.00 hPa | 2024-02-11 |
| | 20.61°C | clear sky | Clear | 65% | 2.89 m/s | 1018.00 hPa | 2024-02-07 |
| | 32.98°C | clear sky | Clear | 22% | 2.99 m/s | 1013.00 hPa | 2024-02-05 |
| | 24.48°C | clear sky | Clear | 35% | 3.82 m/s | 1013.00 hPa | 2024-02-01 |

# 4.UML DIAGRAMS



# UML - Activity Diagram for Weather Forecast




**Aryush Khatri**
**ID:2408485**

# 5.Link to my website:

https://aryush-2408485.serv00.net/

# 6.Conclusion:

Users can use a web-based application in the third prototype that offers meteorological data specific to a city. The application has an intuitive user interface with a search input field and a search button. By entering the name of the city in the input area and then selecting the search option, users can obtain meteorological information for that specific city. The program retrieves weather information via the **OpenWeatherMap** API by sending an HTTP request with the city name as an argument and receiving a JSON-formatted response with the weather information. The weather data is then displayed on the page when the application has analyzed the response it has got.

The information provided includes the name of the city, the date, the temperature, humidity, wind speed, and the amount of rainfall. The temperature that is shown is in Celsius. An emblem displaying the current weather condition is displayed in addition to the description that goes with it.

By tapping the "Show History" button, users of the app can also check the weather history for the selected city. A PHP script receives a request to retrieve the weather history data from a MySQL database when the button is pushed. After that, the information is dynamically presented on the page in the form of a table. To store meteorological data for later use, the app uses local storage. The most recent meteorological data for a city is downloaded and stored as a collection of objects in the local storage. The name of the city, the time, the humidity, the wind speed, the rainfall, and the current temperature are all recorded. The app checks to see if the data for a given city is already in the local storage before submitting an API request. If the data is there, it is retrieved from the local storage and shown on the page. Fewer API queries are made while the program operates more efficiently. Users may easily search for and retrieve weather information for different cities using the weather application. It successfully incorporates OpenWeatherMap API real-time weather data, presents it appealingly, and gives users access to weather history for cities. Additionally, using local data, the application acts as a caching mechanism, enhancing user experience and reducing reliance on external API requests.

# 7.Acknowledgement:

Firstly, I grateful to my Module Leader [Bishal Khadka], as well as the other members of this faculty, for providing me with important direction, support, and encouragement during developing this project. This work would not be as efficient as it is without their significant contributions of expertise, insights, and constructive criticism.

Throughout the development of this prototype, my friends have been helpful, providing me with support and encouragement. I would want to express my gratitude to them for their patience, understanding, and constant support.

Lastly, I would like to express my gratitude to every individual who guided, helped me in this project, and provided feedback, since they were very generous with their time and insights, which contributed to the overall success of this coursework.