UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE
KATHMANDU

Internet Software Architecture
(4CS017)

Final Report

University Id	: 2332917
Student Name	: Niraj Chaudhary
Group	: L4CG16
Word Count	: 2093
Submission Date	: 21thMay, 2023

Table of Contents

1. Prototype 1

In prototype1 we should go to a simple HTML / JavaScript page to retrieve live weather data from the OpenWeatherMap API. Also that when your weather app loads for the first time, we have to show the weather info of your assigned city, And my assigned city is Sioux Falls. We need the Openweather api key. Simply, it is a service that provides weather data and forecasts through an API key. To get the API key there is some step as follow:

- We should go to the OpenWeatherMap website at https://openweathermap.org/.after that we have to sign up for a free account by clicking on the “Sign Up” button.
- Once we complete signing up or logging in. You can go to your profile page. You will find your API key. It should be a long alphanumeric string.
- And copy your API key and use it in your application to make PI requests to OpenWeatherMap.
- The provided code appears to be an HTML file that represents a simple weather app web page. It includes a basic HTML structure with a title, external CSS and JavaScript files, and various HTML elements such as heading, input fields, buttons, and div containers. The app allows users to enter a city name, click a button to search for weather information, and display details like temperature, city name, date, humidity, wind speed, pressure, and visibility.
- After that we provide JavaScript for a weather app. It shows by defining constants for the OpenWeatherMap API key and URL. Then, it declares a variable for the HTML element related to user input and weather display. The ‘check weather ()’ function is defined to fetch weather data from the API based on the user’s city input. If the API returns a valid response, the data is displayed on the HTML elements. If there is an error or the is not found, error messages are shown. The code also includes event listeners to trigger the weather check when the search button is clicked. Additionally, there are functions to format the current time and date, which are updated every 200 milliseconds using ‘setInterval’. Overview of prototype1, this code fetches weather data updates the U, and provides timely information for the weather app.

2. Prototype2

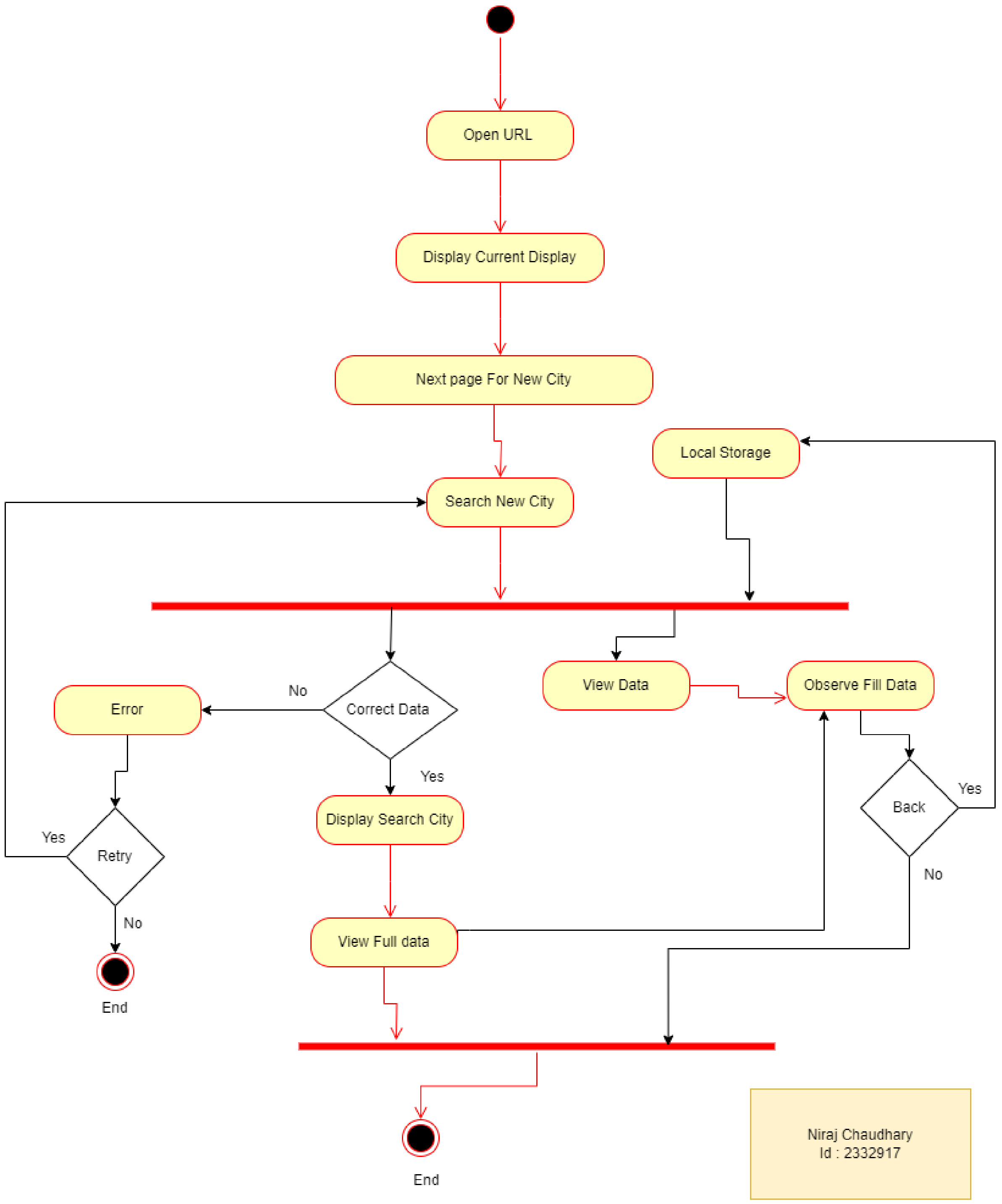
In prototype2 we have to save the past 7 days' data of the assigned city**.** And **also** we will work from PHP and MYSQL. We should be able to show the display of weather data for the past 7 days from the database's current data from local storage for the city. And After that, In prototype1 of the HTML file, I made "index.php" to run the code. There is also a link to another page for selecting a different city. Additionally, it seems to require a PHP file called " weather.php" for data processing. The code sets up a basic weather app interface and incorporates javascript and PHP for functionality and data retrieval. The provided PHP code sets up a database connection and creates a table called "weather" if it doesn't exist. It then retrieves historical weather data for the city "Sioux Falls" from the OpenWeatherMap API using the start and end timestamps. The retrieved data is inserted into the "weather" table. Finally, it fetches all the data from the table and displays it in an HTML table format. The code ensures that each unique date's data is inserted only once by checking the current date against the previous date. If the current date differs, the data is inserted into the table. The displayed table includes columns for ID, City Name, Temperature, Pressure, Humidity, Wind, Description, and date. This code fetches historical weather data, stores it in a database, and presents it in a tabular format. The code of PHP fetches data from a database table called "weather" and displays it in an HTML table format. First, a SQL query is defined to retrieve all entries from the "weather" table in descending order based on the "id" column. The query is performed using the $conn->query() function, and if successful a while loop is used to reiterate over individually row of the consequence set Inside the loop, the data from each row is accessed using the $data variable, and the desired values (such as city name, temperature) are echoed within table cell (<td>). As the loop iterates, each row's data is displayed in a new table row (<tr>). This code retrieves data from the "weather" table and generates an HTML table to present it in a tabular format on the webpage.

3. Prototype3

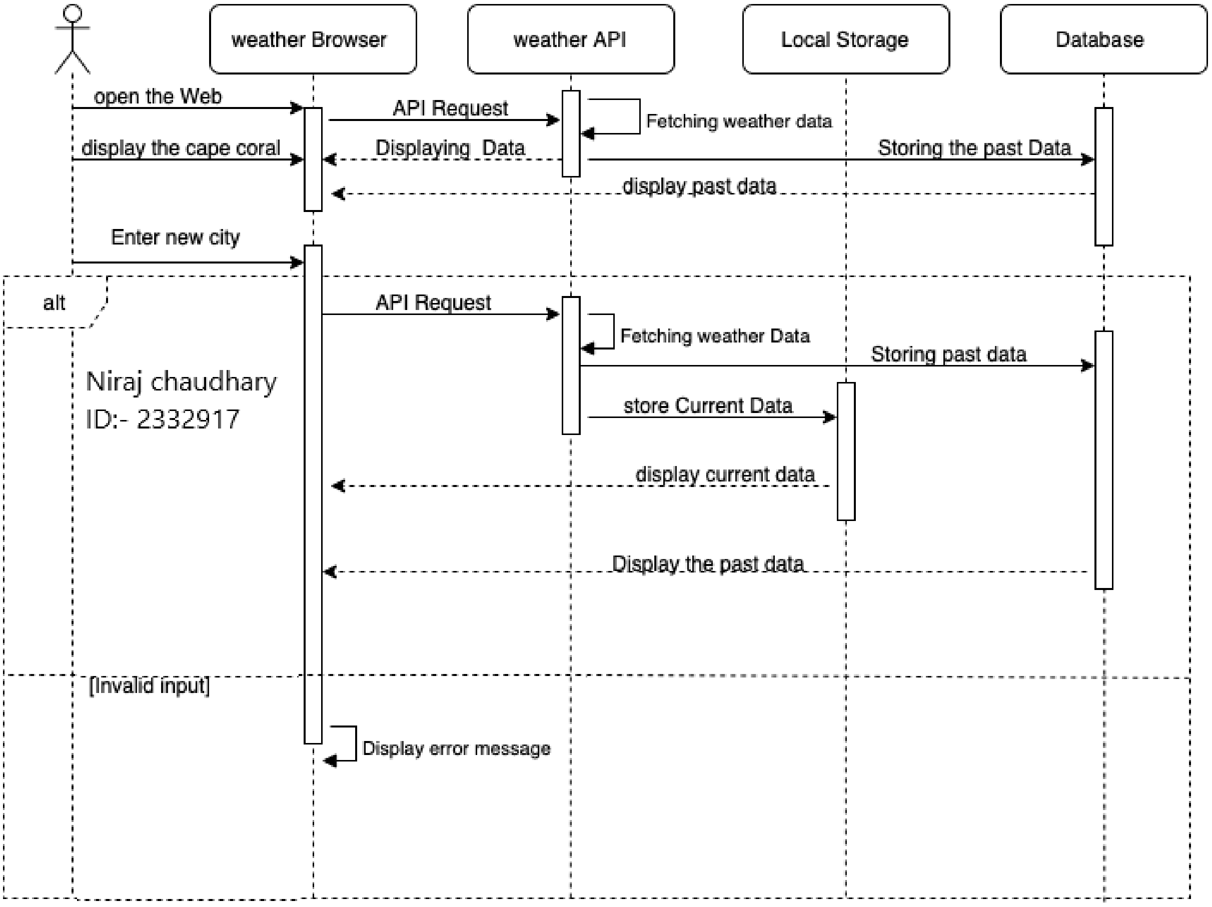
In prototype3 we have to off the wifi and then stored previously searches cities data should be displayed from the database and local storage if a user searches for a new city. we have to show either some alert msg about no internet connection or failed to fetch error. We have to also show the display on the current data of the searched city of 21st May. Prototype1 I have creates a hyperlink that redirects the user to "othercity.php" when clicked. The '<button>' element serves as a clickable button with the label "Oher city". This code is likely used to navigate the user to a different page(othercity.php) when they want to explore weather data for a different city. By clicking the "Other City" button, the user is different from the specified page where they can likely enter the desired city name or select a city from a list to view the weather information for that particular city. The code represents an HTML page structure. Inside the '<body>' tag, there is a main section with a container. Within the container, there is a form with an input field where users can enter a city name. When the form is submitted, it will send the data to "other.php". The page also includes a '<div>' element for displaying an error if any occurs. Below the form, there is a section for displaying the current time and date and another section for displaying temperature and other weather–related material. The code includes a reference to an external CSS file style and CSS is a design weather application. An external JavaScript file called "other.js" is for handling functionality. The code provides a structure for a weather application page with user input, an error display, and a weather information section. The Javascript code sets up event listeners and functions to fetch weather data from the OpenWeatherMap API based on user input. When the form is submitted the 'searchedWeatherDatas' function is called which fetches the weather data for a specified city. If the city is not found, an error is thrown. Otherwise, the 'showingAllDatas' function is called to display the weather information on the webpage. The code also includes displaying errors, storing weather data in local storage, retrieving stored weather data, and fetching weather data from local storage. There is a function 'update time' that updates the current times and dates on the web pages every second. The code also retrieves the last searched city from local storage and displays its weather data if available. The code provides functionality for searching and displaying weather information, handling errors, and updating the current time and date on the webpage. The Code PHP begins a connection to an MYSQL database and creates a table name "past_weather_data" if it does not already exist. When the form is submitted via a POST application, the code fetches the geographical data of the specified city from the OpenWeatherMap API and retrieves historical weather data for the past 7 days using the latitude and longitude of the location. The fetched data is then filtered to remove duplicate entries for the same date and city. The filtered data is inserted into the "past_weather_data" table, avoiding duplicates. Afterward, the code retrieves the weather data from the past 7 days from the database and displays it as an HTML table. If any error occurs during the process, appropriate error messages are displayed. The code handles the retrieval and storage of weather data for the specified city as well as displaying it in a tabular format on the webpage.

4. UML Diagram

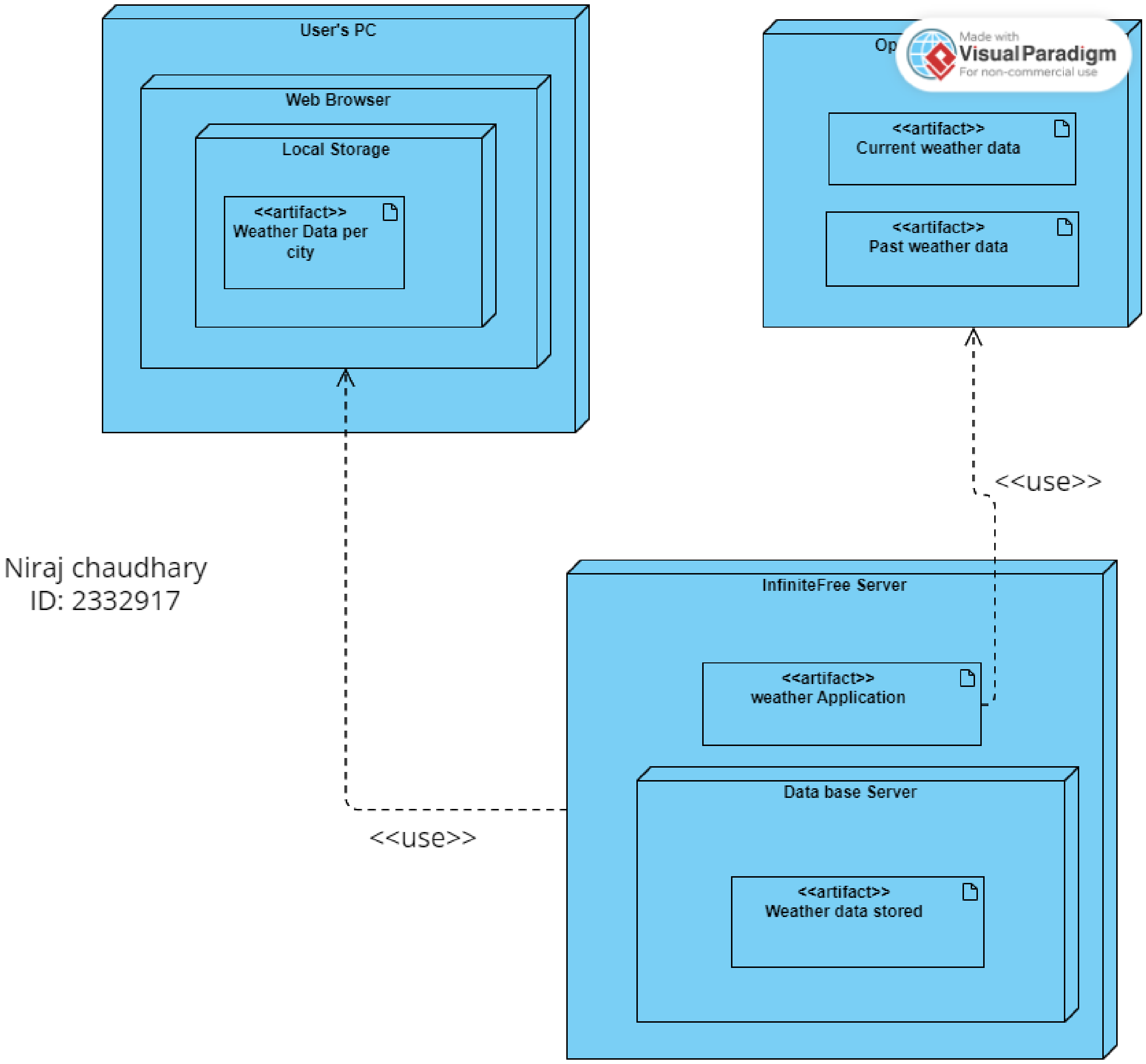
4.1. Activity Diagram



4.2. Sequence Diagram



4.3. Deployment Diagram



5. Summary of web hosting

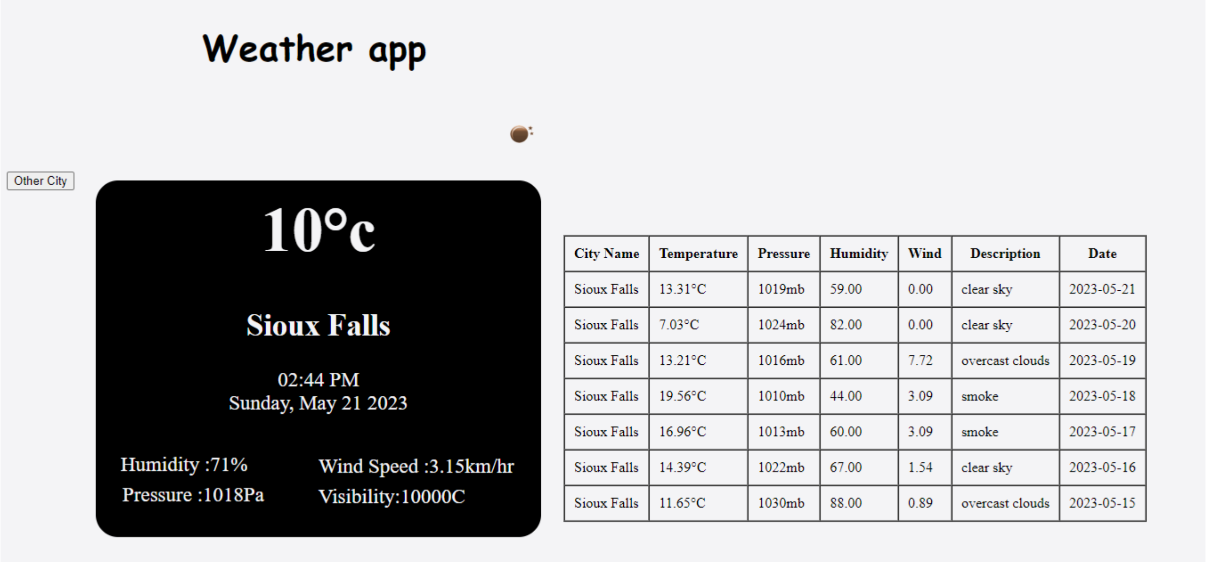
To create a website using InfinityFree a free web hosting platform that supports PHP and MYSQL. There are so steps as follow

1. We have visited the infinityFree website at https://infinityfree.net/. After that click on the “Sign Up ” button or a similar option to create a new account.

2. Fill in the required information, including your email address, password, and website details.
3. Choose a subdomain for your website. This will be a part of your website’s URL(e.g. yoursite.infinityfreeapp.com).
4. Finished the registration process and log in to Your InfinityFree account.
5. Once we finished the registration process then log in. You will be taken to the control panel. Look for an option to add a new website or domain.
6. Select the option to add a new website or domain. Enter your desired domain name if you have one or choose to use the subdomain provided by InfinityFree.
7. We should follow the instruction to set up your domain or subdomain. This may involve updating your domain’s nameservers or creating a CNAME record if you’re using a custom domain.
8. After your domain or subdomain is set up, navigate to the file Manage section in the control panel This is where you can manage your website’s files.
9. Upload your website’s PHP files and any other necessary assets to the appropriate directory in the File Manager. You can create new folders if needed.
10. Create a MYSQL database for your website. In the control panel, find the MYSQL database section and follow the instruction to create a new database. Create note of the database name, username, password, and hostname provided.
11. Update the connection details in your PHP Files to reflect the database information. You will have a ‘connection.php’ file where you will set the hostname, username, password, and database name.
12. Test your website by accessing the domain or subdomain in a web browser. Make sure your PHP scripts are running correctly and able to connect to the database.
13. After that Last message we will find Congratulation! You have successfully created a website on InfinityFree with PHP and MYSQL support. You can continue to manage your website and make updates through the InfinityFree control panel and File Manager. Remember that Infinity hosting service with certain limitations. So be sure to review their term and condition to understand any restrictions or upgrade options that may be available.

After hosting your website, it is important to check if everything is working as per your requirements. This can be done by systematically testing various aspects of your website. Start by visiting your website’s pages and ensuring that they load properly without any errors. Test any forms or interactive elements to ensure they function correctly. Check database connectivity and functionality by performing database operations and verifying the results. Test different user scenarios and functionalities to ensure proper behavior. Additionality, check for any broken links or missing assets on your website. By thoroughly testing these elements, we can ensure that your website is functioning as intended and provides a seamless user experience.

6. Screenshots of your host application



Weather app

10°c

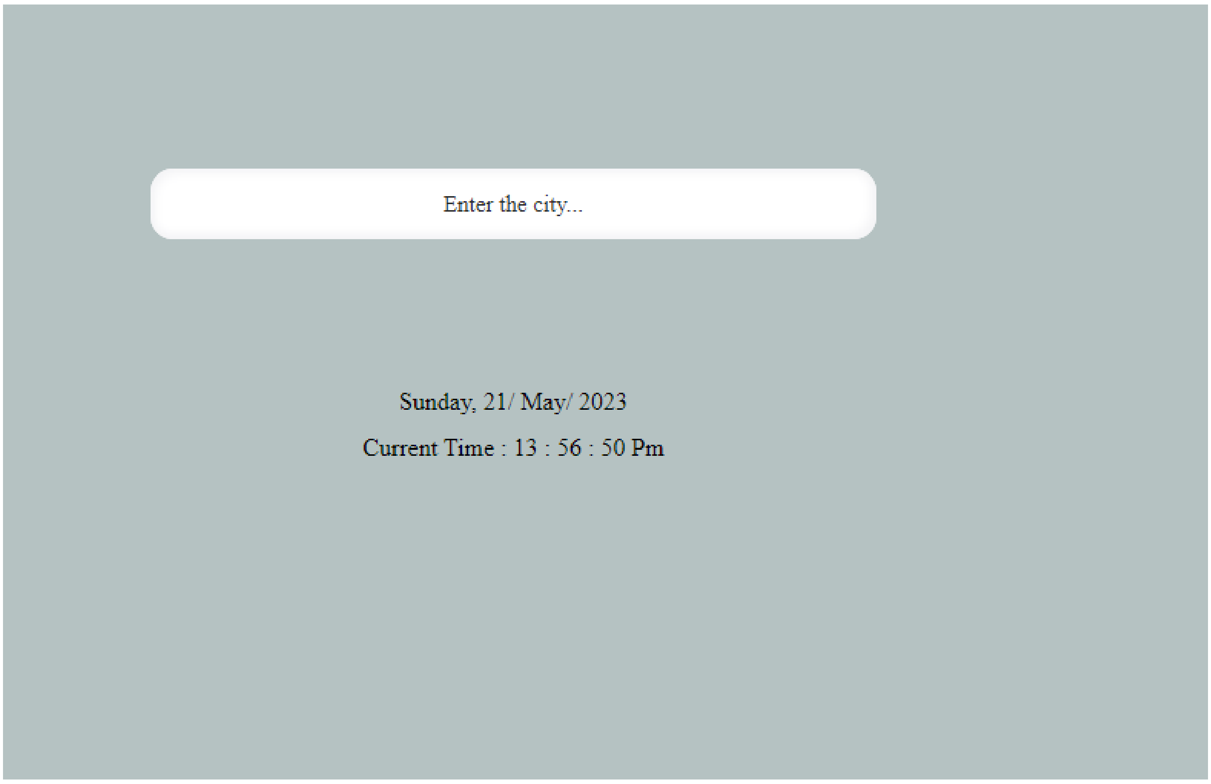
Sioux Falls

02:44 PM
Sunday, May 21 2023

Humidity :71%
Pressure :1018Pa

Wind Speed :3.15km/hr
Visibility:10000C

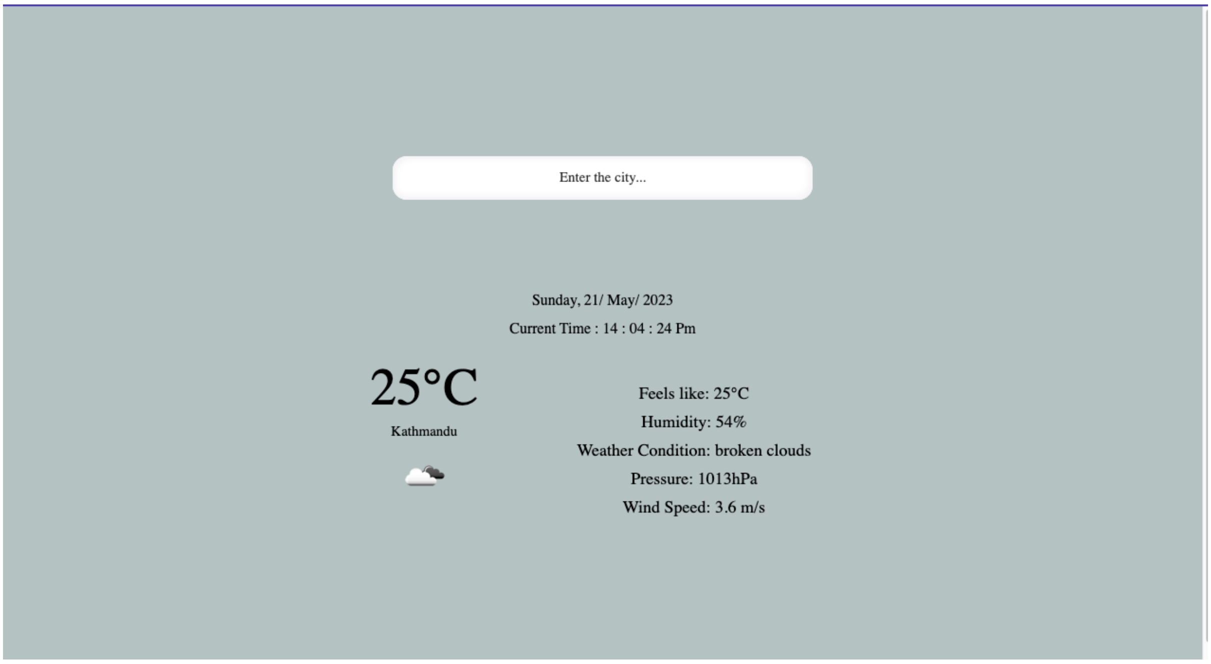
City Name	Temperature	Pressure	Humidity	Wind	Description	Date
Sioux Falls	13.31°C	1019mb	59.00	0.00	clear sky	2023-05-21
Sioux Falls	7.03°C	1024mb	82.00	0.00	clear sky	2023-05-20
Sioux Falls	13.21°C	1016mb	61.00	7.72	overcast clouds	2023-05-19
Sioux Falls	19.56°C	1010mb	44.00	3.09	smoke	2023-05-18
Sioux Falls	16.96°C	1013mb	60.00	3.09	smoke	2023-05-17
Sioux Falls	14.39°C	1022mb	67.00	1.54	clear sky	2023-05-16
Sioux Falls	11.65°C	1030mb	88.00	0.89	overcast clouds	2023-05-15



Enter the city...

Sunday, 21/ May/ 2023

Current Time : 13 : 56 : 50 Pm



Enter the city...

Sunday, 21/ May/ 2023

Current Time : 14 : 04 : 24 Pm

25°C

Kathmandu

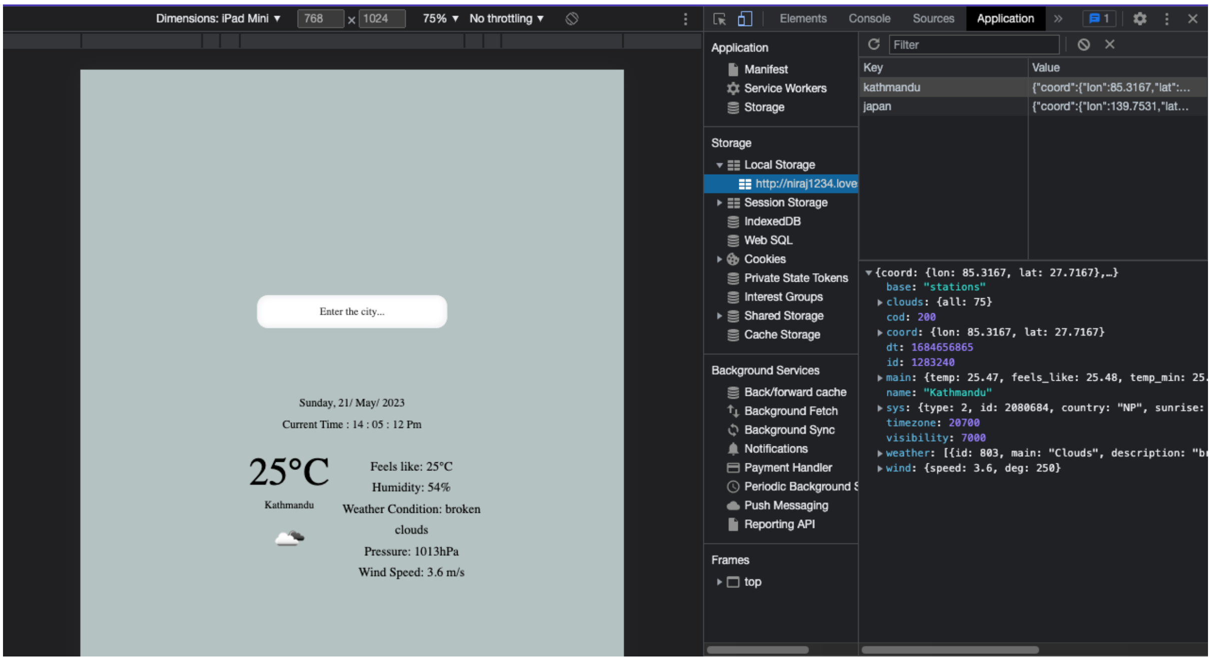
Feels like: 25°C

Humidity: 54%

Weather Condition: broken clouds

Pressure: 1013hPa

Wind Speed: 3.6 m/s



Enter the city...

Sunday, 21/ May/ 2023

Current Time : 14 : 05 : 12 Pm

25°C

Kathmandu

Feels like: 25°C

Humidity: 54%

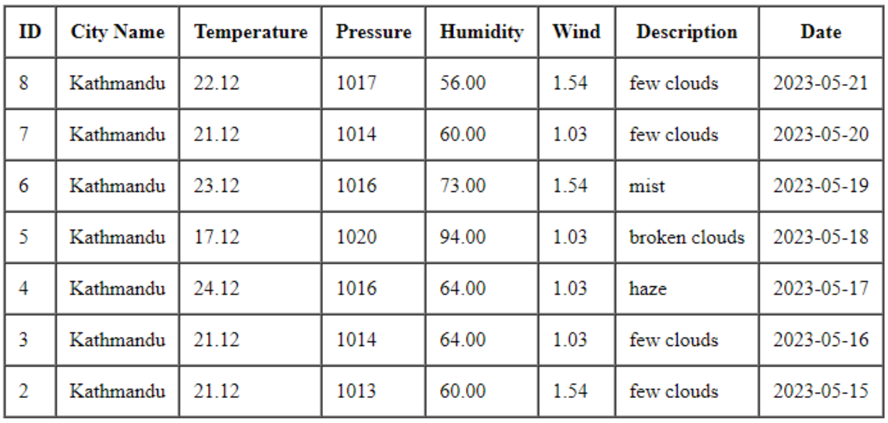
Weather Condition: broken clouds

Pressure: 1013hPa

Wind Speed: 3.6 m/s

Key	Value
kathmandu	{"coord":{"lon":85.3167,"lat":...
japan	{"coord":{"lon":139.7531,"lat...

```
▼{coord: {lon: 85.3167, lat: 27.7167},…}
   base: "stations"
 ▶ clouds: {all: 75}
   cod: 200
 ▶ coord: {lon: 85.3167, lat: 27.7167}
   dt: 1684656865
   id: 1283240
 ▶ main: {temp: 25.47, feels_like: 25.48, temp_min: 25.
   name: "Kathmandu"
 ▶ sys: {type: 2, id: 2080684, country: "NP", sunrise:
   timezone: 20700
   visibility: 7000
 ▶ weather: [{id: 803, main: "Clouds", description: "br
 ▶ wind: {speed: 3.6, deg: 250}
```



ID	City Name	Temperature	Pressure	Humidity	Wind	Description	Date
8	Kathmandu	22.12	1017	56.00	1.54	few clouds	2023-05-21
7	Kathmandu	21.12	1014	60.00	1.03	few clouds	2023-05-20
6	Kathmandu	23.12	1016	73.00	1.54	mist	2023-05-19
5	Kathmandu	17.12	1020	94.00	1.03	broken clouds	2023-05-18
4	Kathmandu	24.12	1016	64.00	1.03	haze	2023-05-17
3	Kathmandu	21.12	1014	64.00	1.03	few clouds	2023-05-16
2	Kathmandu	21.12	1013	60.00	1.54	few clouds	2023-05-15

7. Learning Outcomes and Conclusion

In this module, we have gained valuable insights into the process of building modern software applications. We have focused on understanding the exchange of data between hardware and software layers through HTTP services, which are crucial for web development. Throughout the module, we have learned and practiced JavaScipt, PHP, and SQL which are essential tools for developing dynamic web applications. The project specifically the weather application has been instrumented in deepening our understanding of both Front-end and back-end development. It has allowed us to collaborate and appreciate the roles of different team members involved in the software development process. Overall, this module has provided us with practical experience and knowledge that will be valuable in our future endeavors in software development.

8. Website Webpage

URL: http://nirajchy.great-site.net/Final%20prototype/