Module Code and Name

4CS017: Internal Software Architecture

Formative Documentation of Prototype 1, 2 and 3

Student Name: Priya Shrestha

Uni ID: 2329818

Module Leader: Deepson Shrestha

Submission Date: May 21, 2023

Acknowledgement

I would like to express my extreme gratitude towards Herald College Kathmandu for providing the best learning environment and towards the entire faculty of Internet Software Architecture. This module has helped to boost up the knowledge about the web development. A special thanks to my tutor Mr. Karan Shrestha for guiding us throughout the coursework. These coursework has been nothing but a great opportunity to develop our learning process.

Table of Contents

Table of Figure

1 INTRODUCTION

This report depicts the learning outcomes from the coursework of Internet Software Architecture Module. This coursework includes total three prototypes. In first prototype, we have simply build a weather app using HTML, CSS and JavaScript and fetch all the current weather data from openweathermap API. Then in the second prototype, instead of using JavaScript, we have used php and database. Lastly, we have used local storage to save the searched data of every entered city and fetched the php file using JavaScript and host the website using infinityFree.

2 PROTOTYPE DESCRIPTION

Description on each prototypes are mentioned bellow:

2.1 Prototype 1

In prototype 1, I have client side scripting languages such as HTML, CSS and JavaScript. HTML created the frame for the weather app, CSS gave the styling to it and JavaScript gave the functionality. The current data as well as the data of every searched city is fetched using OpenWeatheMap API. I have implemented "document.querySelector" to select the HTML element and stored in the respective variable. As my assigned city was "Bakersfield", it was stored in a variable name CurrentCity. Then, event listener was added in submit even in search form. To prevent the default search action in the city, preventDefault() function is used. The city entered by the users in the input field is retrieved and CurrentCity variable is updated by the entered city for each search. Then, the getWeather function is used to get all the weather information. getWeather function fetches the weather data such as temperature, pressure, humidity, wind and icons from the OpenWeatheMap API according to the currentcity. TimeandDate function takes time, timezone as parameters and converts a UNIX timestamp and timezone offset into a formatted date and time string, while the CountryCode function converts a country code into the corresponding country name using the Intl.DisplayNames. The fetched data is then processed and used to update the various HTML elements to make them functional. At last, to make the default city's information is displayed initially each time the page loaded, the getWeatherfunction is called. This is how, I built a well functional weather app which displays the weather data of the default city as well as the searched city.

2.2 Prototype 2

In prototype 2, there is the demonstration of implementation of php and database. Instead of using javascript, Php is used to fetch the weather information through the openweathermap api. For the past data, history api is used and database is used to store the data .The PHP script that retrieves weather data from the OpenWeatherMap API and saves it to a database. . Mysqli_connect(4 parameters) are used to connect with the database. Apikey variable is used to store the api key and then getWeatherData function uses the OpenWeatherMap API to retrieve current weather data for a given city. To acquire forecast data for a given city, the getForecastData function uses the OpenWeatherMap API

We have put "Bakersfield" as the assigned default city unless another city is entered. 'SaveForecastData' function is used to save all the forecast data in the database. It loops through the forecast data, and checks if the data for a certain date already exists in the database and inserts the data f it doesn't. Another function 'saveCurrentWeatherData' saves the current weather data in the database.

The function getPastWeatherData returns historical weather data for a city. It uses the getCityCoordinates function to obtain the city's latitude and longitude, and then retrieves meteorological data for each day within a specified range.The previous weather data is saved in the database using 'savePastWeatherData' function. To acquire a city's latitude and longitude, the getCityCoordinates function uses the OpenWeatherMap API. 'getPastForecastData' function retrieves past forecast data from the database based on the specified city, start date, and end date.

The script uses getWeatherData to retrieve the current weather data for the selected city and saves it in the database using saveCurrentWeatherData. It uses getForecastData to retrieve forecast data for the selected city and saves it in the database using saveForecastData. It uses getPastWeatherData to retrieve past weather data for the selected city and saves it in the database using savePastWeatherData.

Finally, the script delivers a JSON-encoded response that includes the city, current weather data, previous forecast data, and any other data that was requested. In overall, the default city's weather data is fetched and displayed initially in the browser and then of the entered city

2.3 Prototype 3

In prototype 3, I have demonstrated the implementation of local storage, database and web hosting. Now, in this phase, weather app is made accessible in offline mode using local storage. In this prototype we have use, JS to fetch previous php file (weather.php). The retrieval of weather data is handled by the 'getWeatherData()' function. It first checks to see if the data for the selected city is already in the browser's local storage. If exists, the data is retrieved from the local Storage else, it fetches the php.file which has all the data stored in database, so is the data is unavailable in the local Storage, data is retrieved from the

database. When the response received, the data is stored in local storage, displayed on the webpage, and a message is logged to indicate that the data was fetched from the server.

When data for a city is already present in local storage, the 'fetchWeatherAndForecast()' method is called. It gets the data from local storage, displays it on the webpage, and logs a message about it (Data already in local storage, data was fetched from local storage). If the data is not available in local storage, 'getWeatherData()' is called to retrieve it from the server.

Then, an event listener is used to the submit event of the search form to directly call the `fetchWeatherAndForecast()` function and display the response when the form is submitted.

The `displayWeatherData()` function is used to display the current weather data on the webpage, which retrieves all relevant information to display the weather details. The `displayPastForecastData()` function displays the past forecast data on the webpage by iterating through an array of past forecast objects and generates HTML markup for each day's details.

Finally, when the webpage first loads, the 'window.onload' event handler calls the 'fetchWeatherAndForecast('Bakersfield')' method. This retrieves and shows weather information of a week for the default city.

3 UML Diagrams

3.1 Activity Diagram:

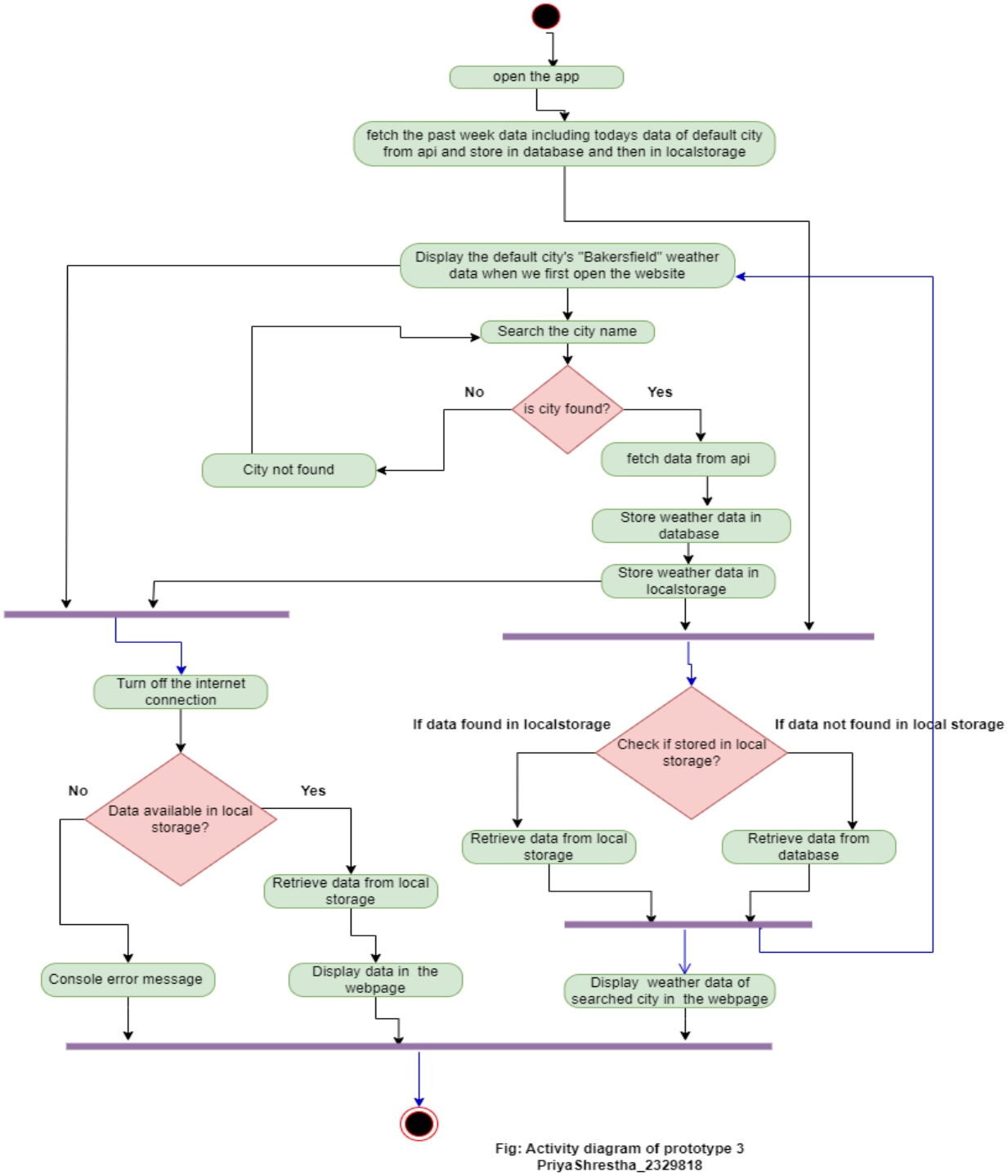


Figure 1 Activity diagram of prototype 3

3.2 Sequence Diagram:

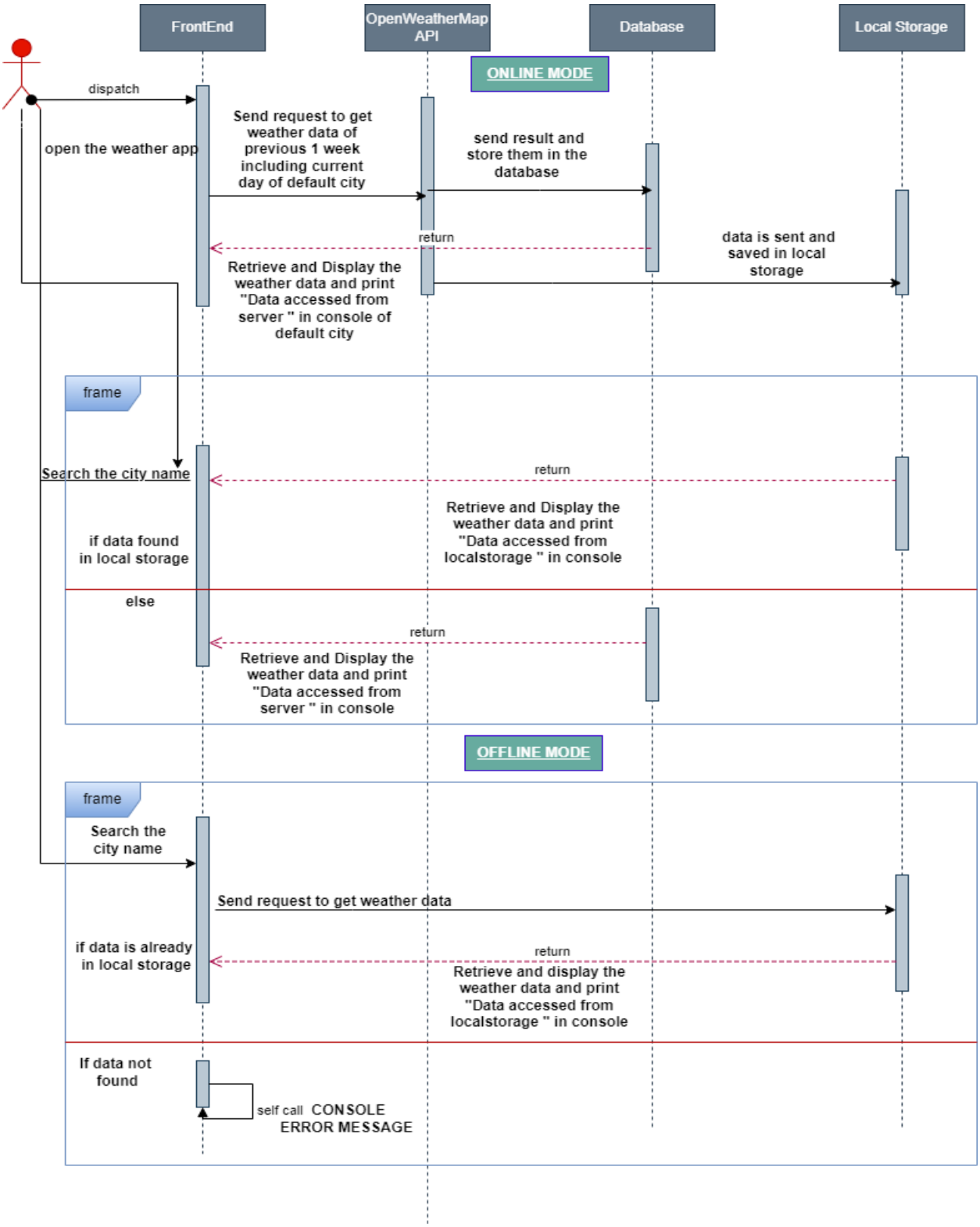


Figure 2 Sequence Diagram of prototype 3

3.3 Deployment Diagram:

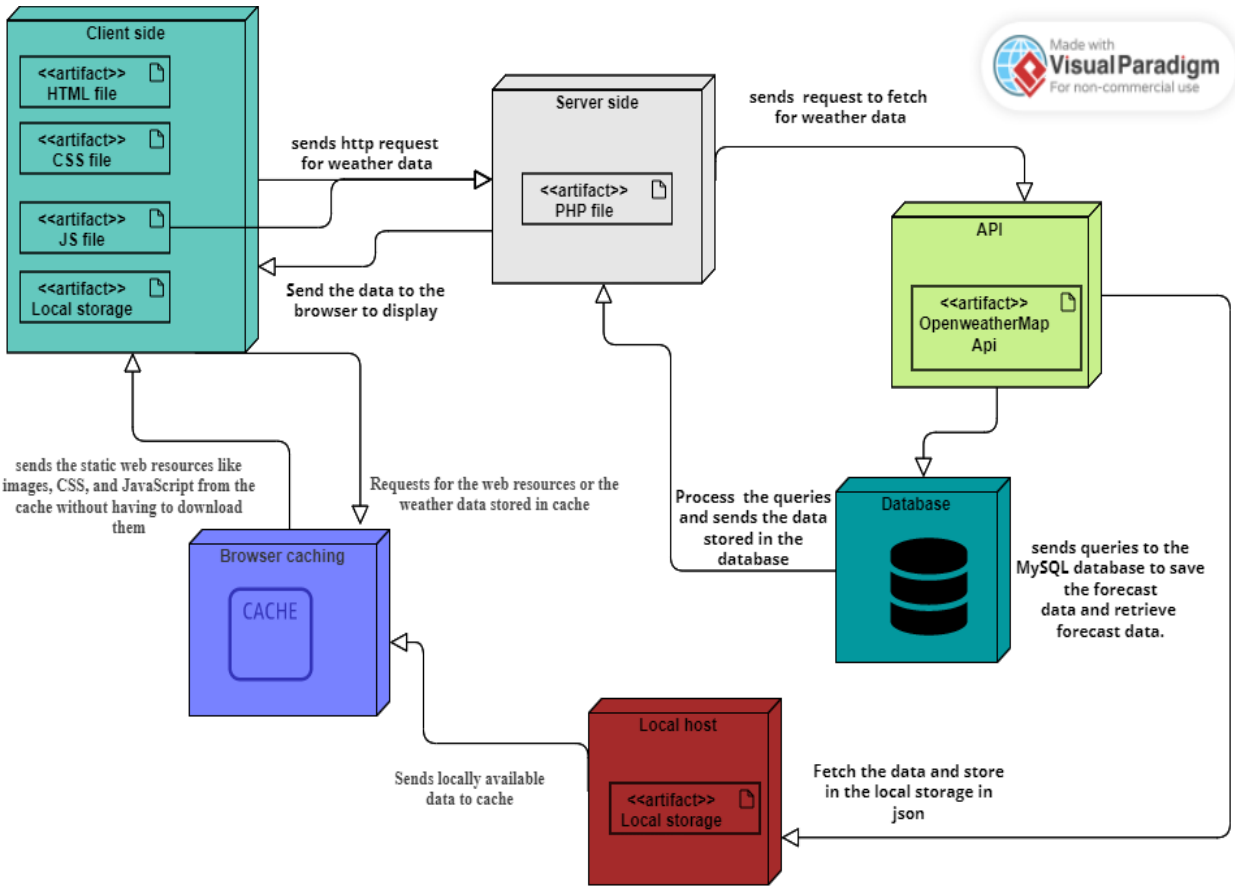


Figure 3 Deployment Diagram of pototype 3

3 Web Hosting

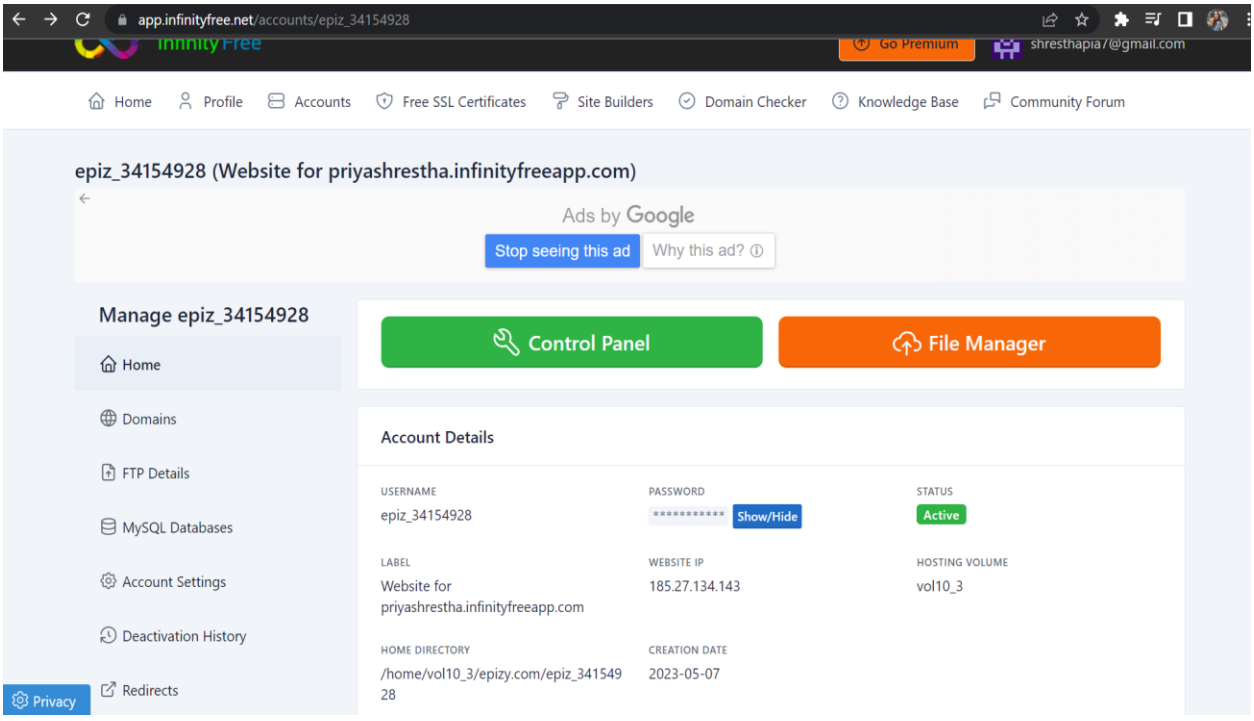
Infinity free is used for the hosting purpose of this weather app. Hence, I have signed in to the infinity free.

Then, we need to create a hosting account. To create the account, we can choose the sub domain of our desire and the domain extension. I have selected priyashrestha as sub domain and infinityfreeapp.com as domain extension and created an account using password. In the control panel, we can see the instruction which is needed to be approved. Then the vista panel opens.

After correctly finalizing the files for the development of the weather app displaying 7 days weather forecast, all the files are imported to the file manager of the infinityFree. After that, in MySQL section, database is created which stores all the weather information. The connection.php file has my_sqli connect function which has 4 parameters:

Hostname, username, password and dbname. Then we modify all the values of the parameters by the hostname, username, password given by the infinity free when we create the account. Dababase name is put same as the name of database we have created. When all the importing of the corrected files id done, we refresh the page and use our domain link to see if it is working.

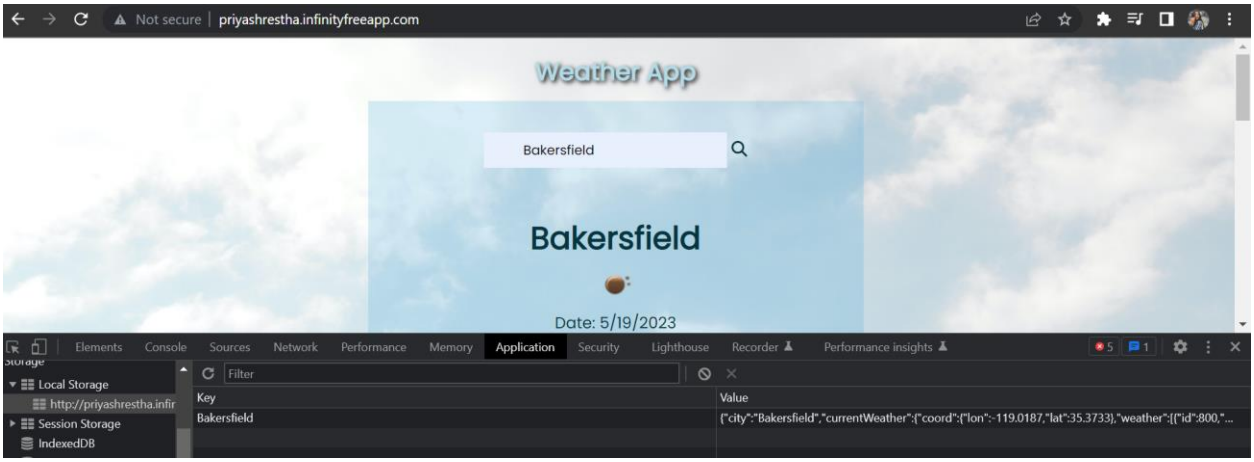
This is how, I have hosted my website.

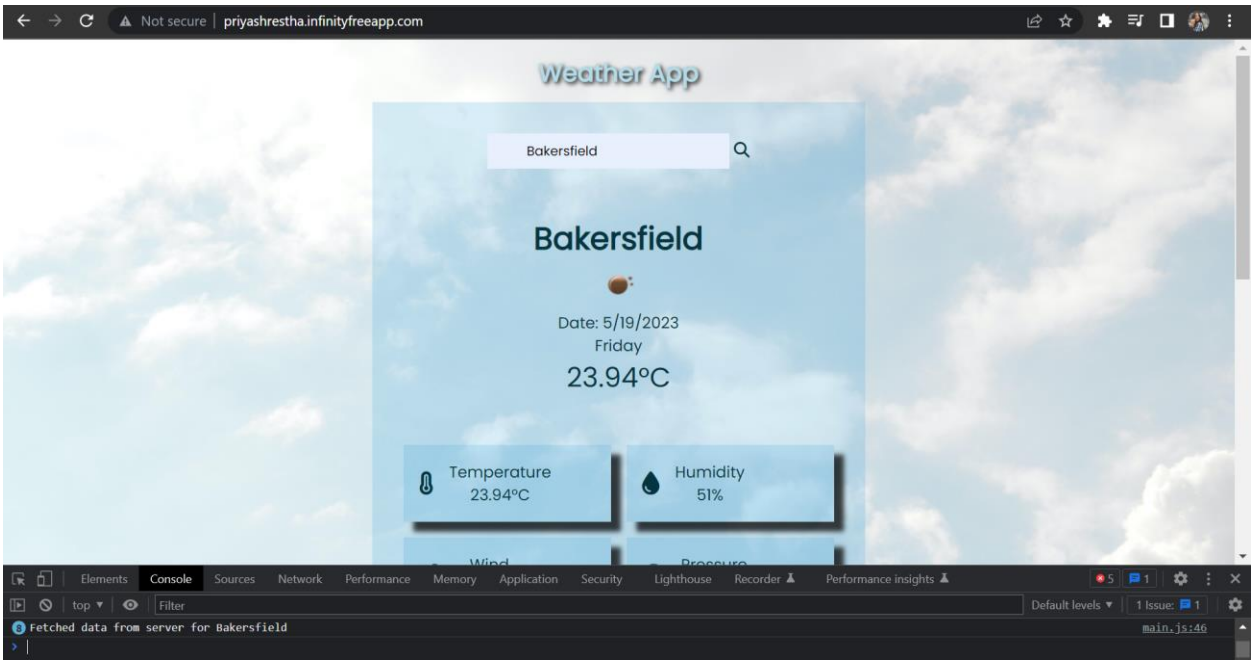


4 Screenshots

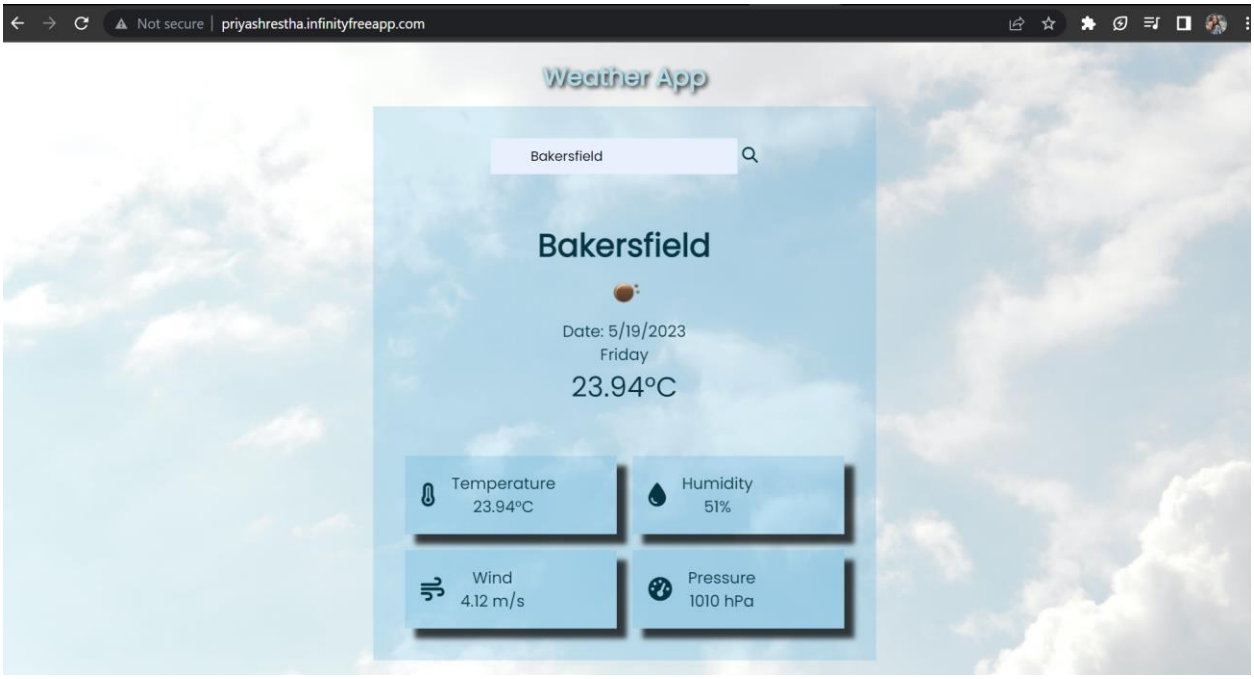
5.1 Online mode:

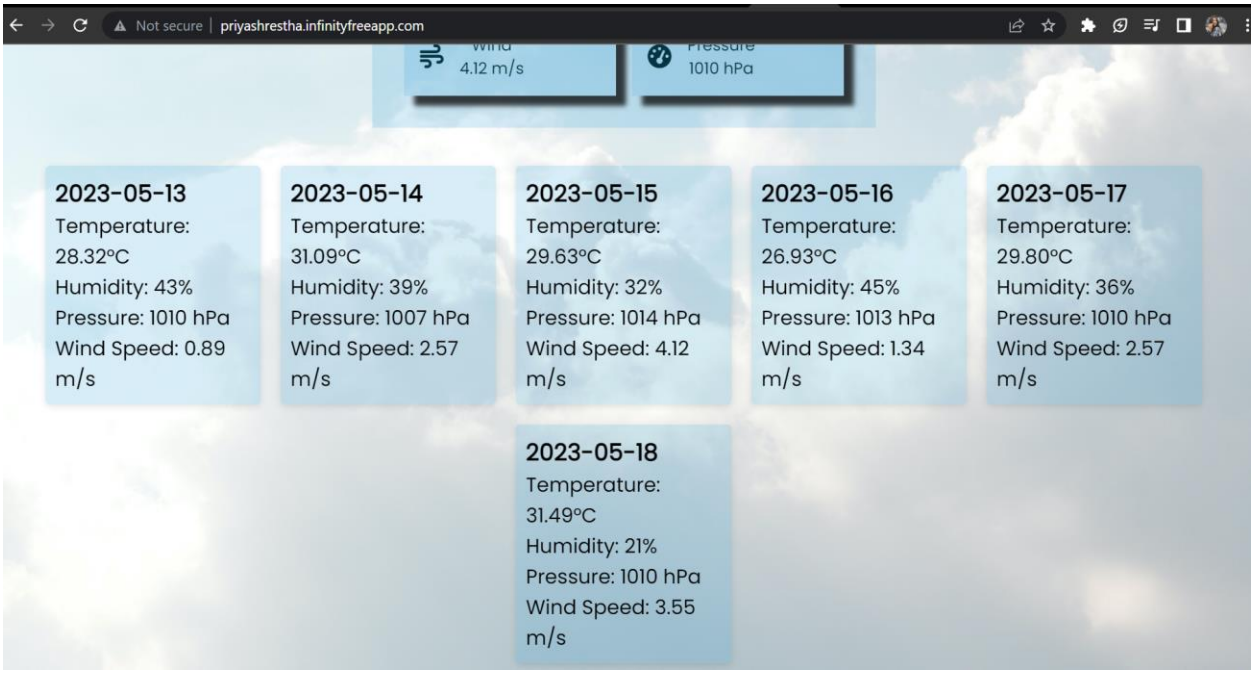
5.1.0 Local Storage and Console Display of default city



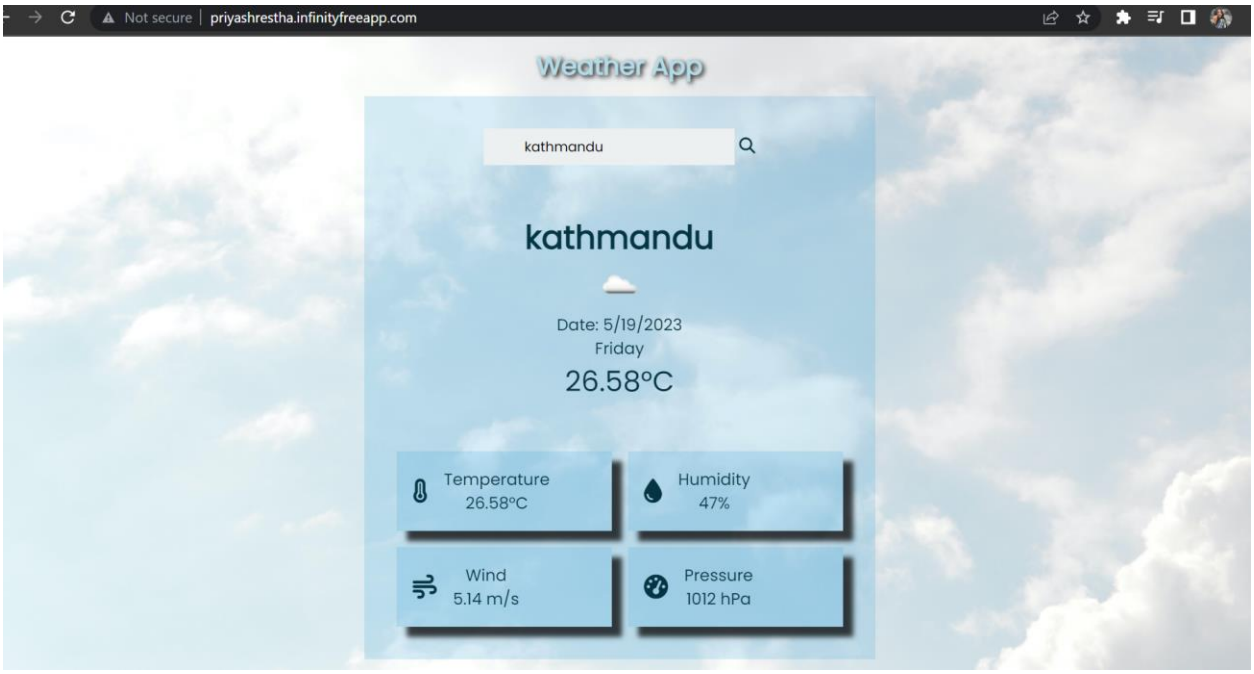


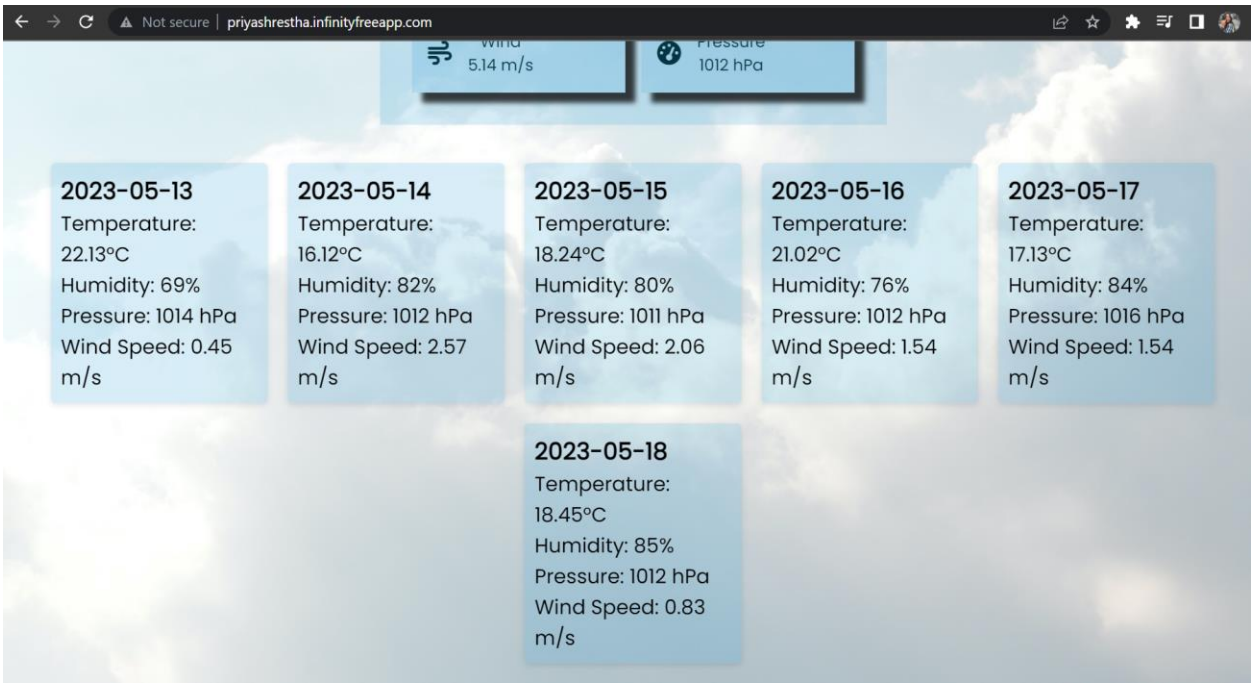
5.1.1 Webpage Display of 7 days weather forecast of default city



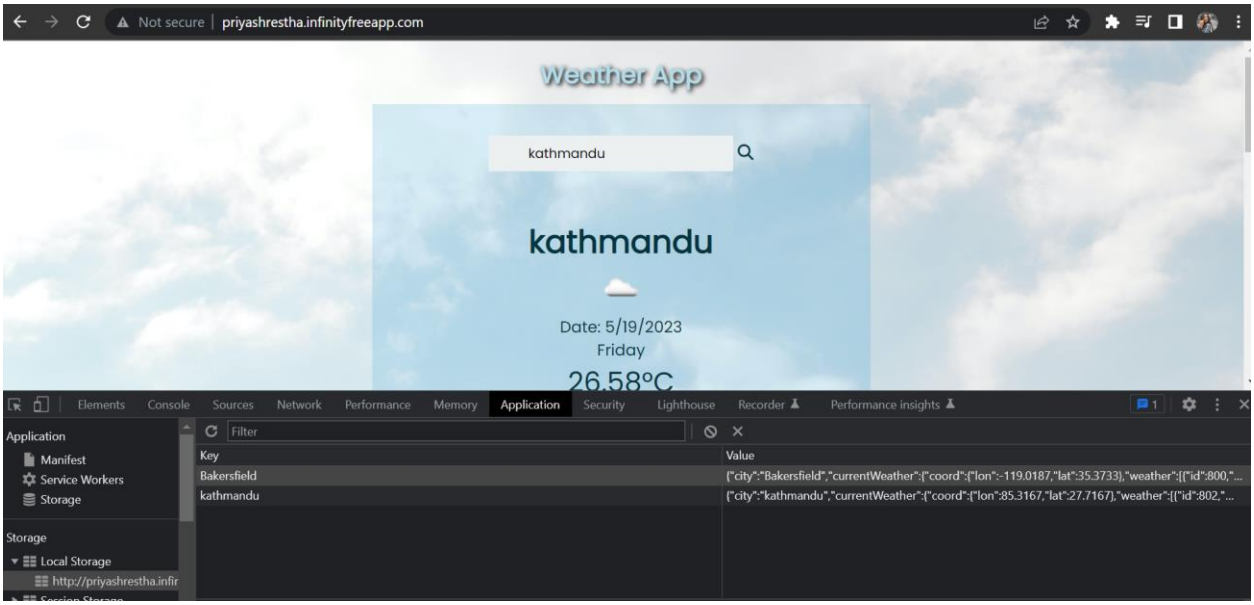


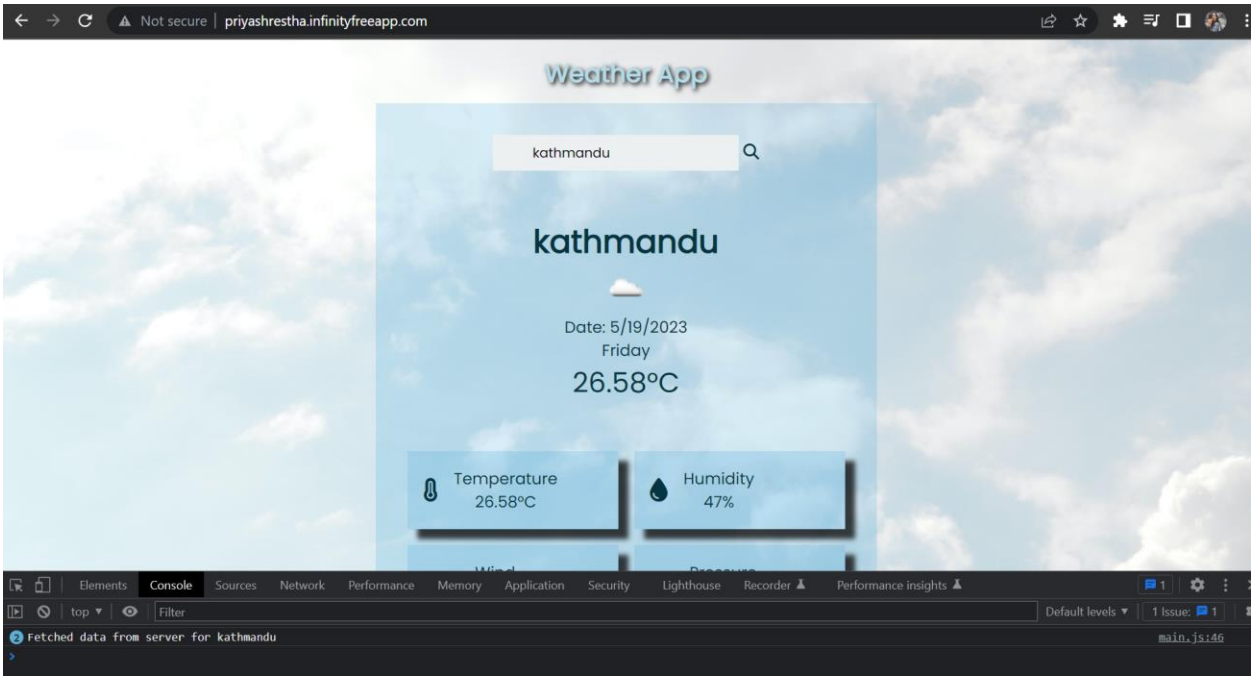
5.1.2 Searched city Webpage display showing 7 days weather data:





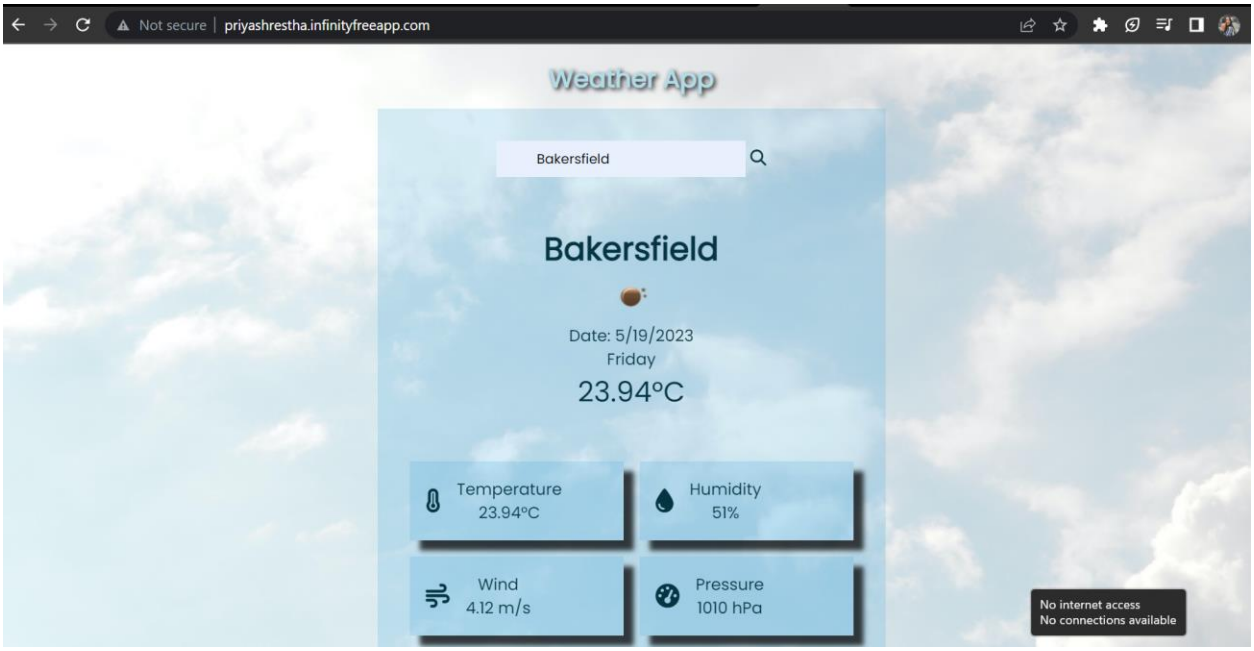
5.1.3 Searched city Local Storage and console display:

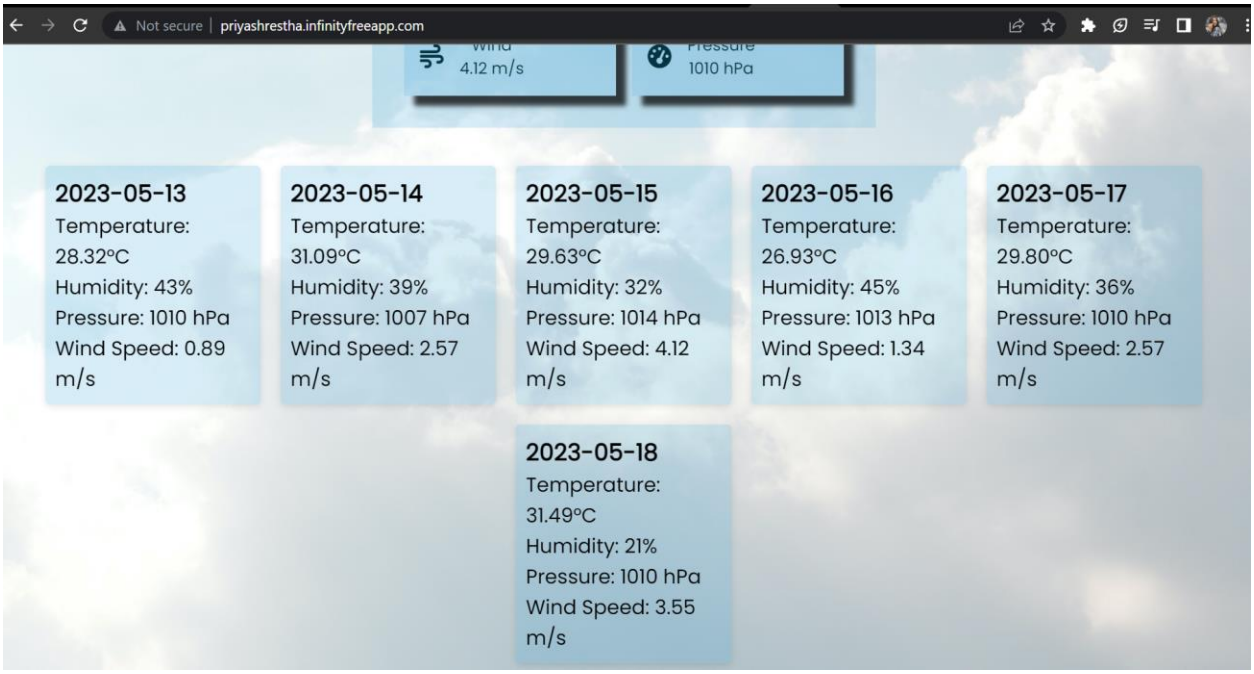




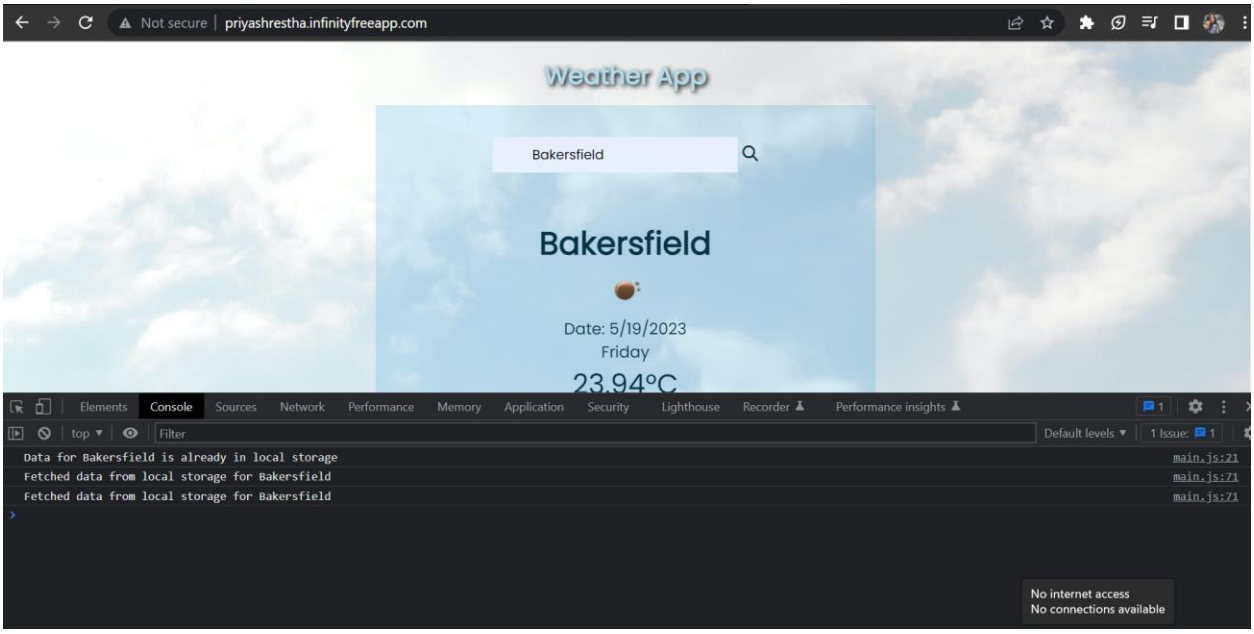
5.2 OFFLINE MODE:

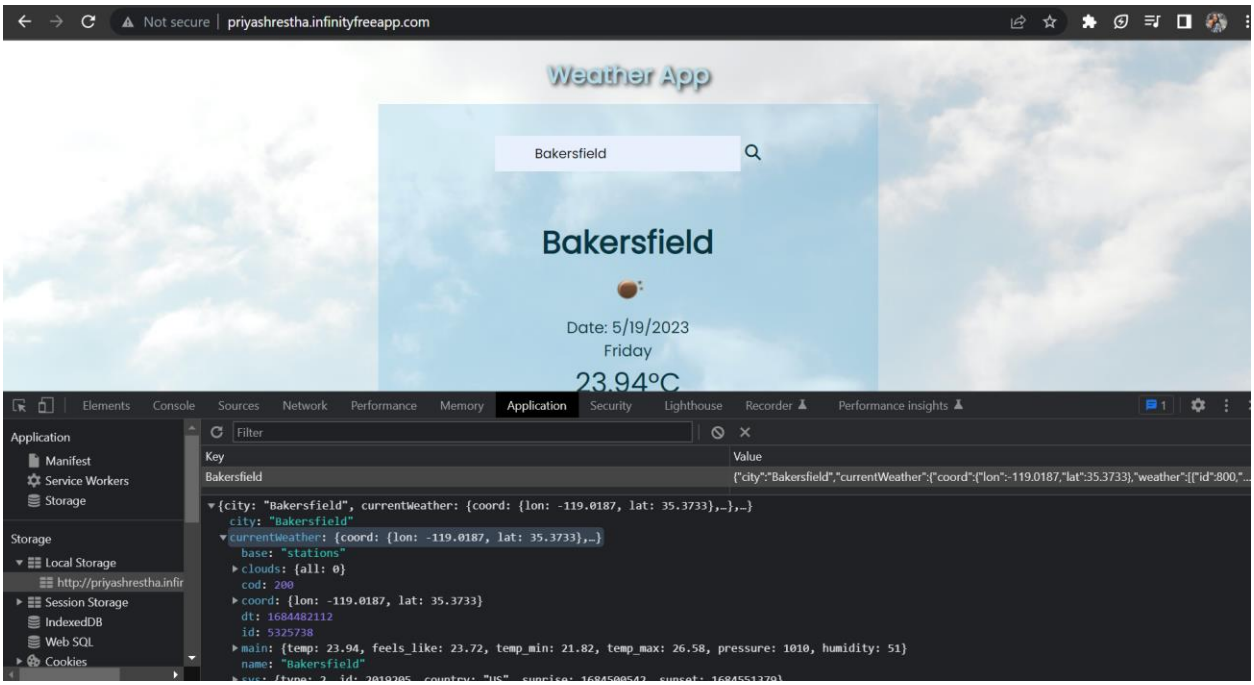
5.2.0 Webpage Display of “DEFAULT CITY “ in offline mode:



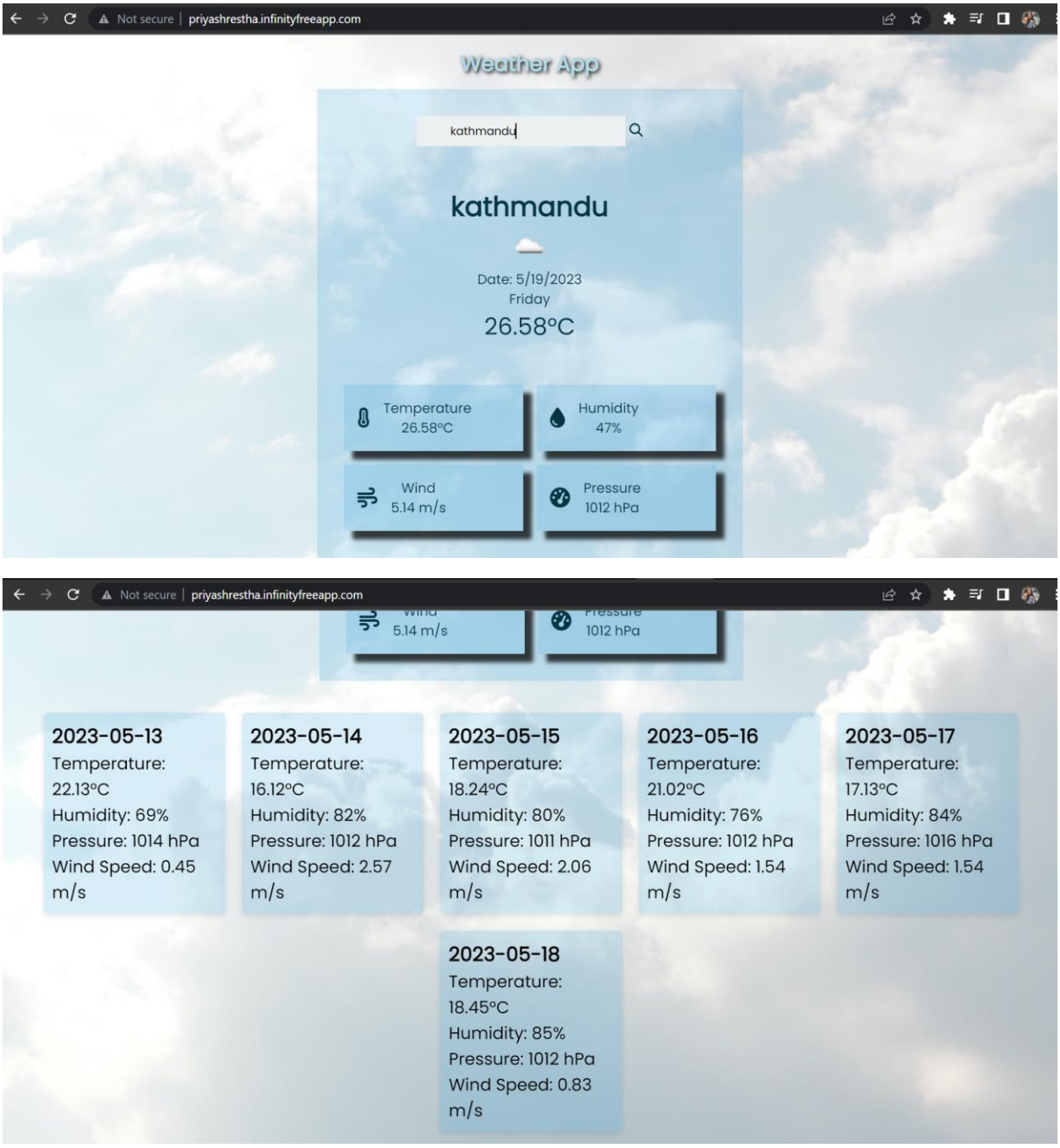


5.2.1 Console and local storage Display of “Default City” in offline mode:

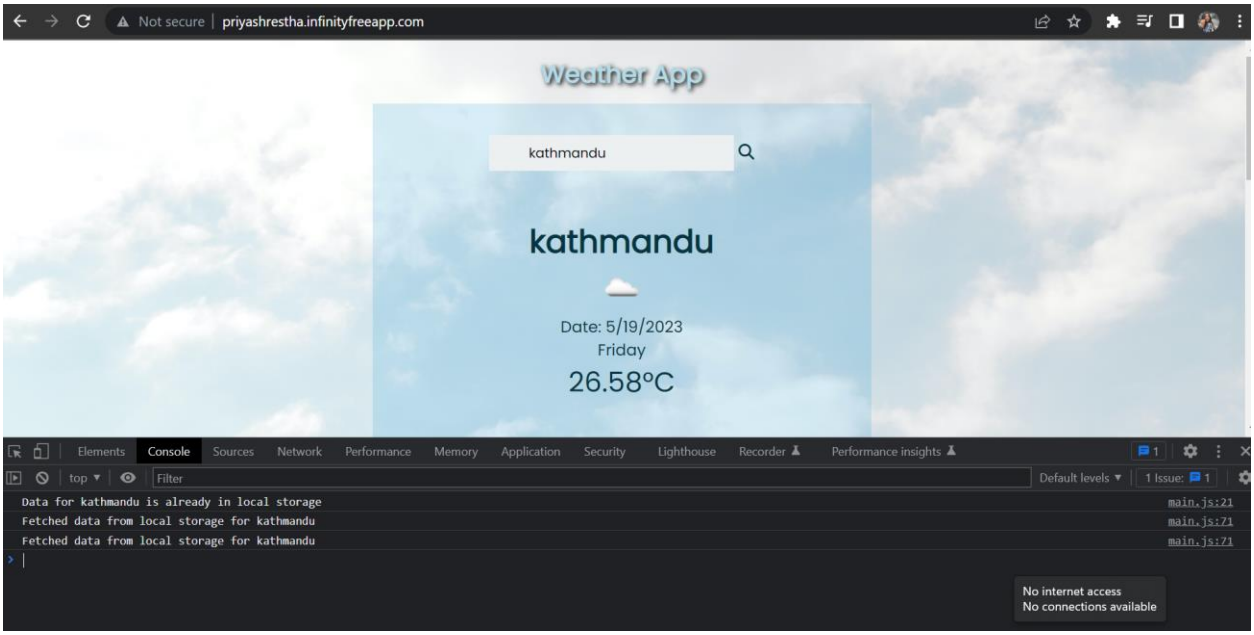


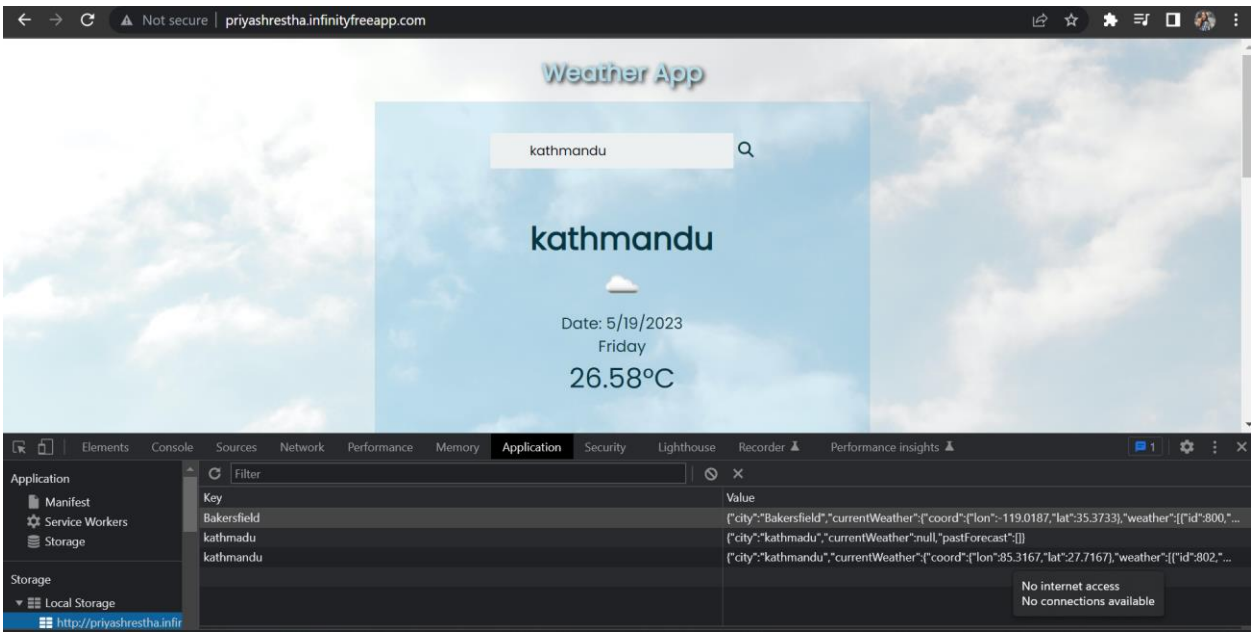


5.2.2 Webpage Display of “Searched City “ in offline mode:



5.2.3 Console and local storage Display of “Searched City” in offline mode:





5 Domain Link

Visit this link to view the weather app website:

http://priyashrestha.infinityfreeapp.com/