Project Report On

18CA433-Modern Web Application Development using MEAN Stack

By

Rakshith S Nadiger

MY.SC.I5MCA19021

In Partial Fulfillment of the Requirements

For the Degree of

Integrated Master of Computer Applications (IMCA)

Under guidance of

Mr.Sreekumar N R

Asst.Professor

Department of Computer Science

AMRITA VISHWA VIDYAPEETHAM UNIVERSITY

Amrita School of Computing

Mysuru Campus, Mysuru.

December 2023

WEATHER FORECAST APPLICATION

Abstract:

This Weather Forecasting Web Application is designed to provide accurate and real-time weather information. Using the OpenWeatherMap API for data retrieval and storage using MongoDB with Mongoose, the application is built using the Node.js backend and Angular.js frontend. The combination of these technologies ensures a seamless and responsive user experience. The purpose of this web application is to empower users with precise weather forecasts, historical data, and an intuitive interface for easy navigation. The scope encompasses diverse weather-related functionalities, from current conditions to extended forecasts, and the use of modern web development tools ensures scalability and maintainability. Several use cases demonstrate the versatility of the application, making it an essential tool for individuals and businesses reliant on accurate weather information.

Introduction:

Weather forecasting is a critical aspect of daily life, influencing activities ranging from personal planning to industrial operations. The Weather Forecasting Web Application is designed to address the need for reliable and accessible weather information. By integrating the OpenWeatherMap API, the application taps into a vast repository of global weather data. MongoDB, with the support of Mongoose, serves as the backend database, allowing for efficient storage, retrieval, and management of weather information. The frontend is developed using Angular.js, providing users with an interactive and dynamic interface. With Node.js as the server-side runtime, the application ensures swift data processing and seamless communication between the frontend and backend.

Purpose:

The purpose of the Weather Forecasting Web Application is to offer users an efficient and user-friendly platform for accessing accurate and real-time weather information. Whether it's a casual user checking daily forecasts or a business planning operations around weather patterns, this application caters to a diverse audience. By leveraging the OpenWeatherMap API, the application ensures reliable data, while MongoDB facilitates efficient data storage and retrieval. The purpose extends to delivering a responsive and intuitive user experience, encouraging regular usage for weather-related decision-making.

Scope:

The scope of the Weather Forecasting Web Application encompasses a wide range of weather-related functionalities. Users can access current weather conditions, detailed forecasts, and historical weather data. The application allows for location-based searches, enabling users to retrieve weather information for specific regions. The scope also includes the potential for future enhancements, such as personalized user profiles, weather alerts, and integration with other APIs for additional data layers. The use of a robust tech stack ensures that the application can scale to accommodate evolving user needs and technological advancements.

Tech stack & API:

- **OpenWeatherMap API:** The application leverages the OpenWeatherMap API to source accurate and up-to-date weather data. This API provides a comprehensive set of weather-related information, including current conditions, forecasts, and historical data.
- **MongoDB with Mongoose:** MongoDB, a NoSQL database, is employed as the backend storage solution for the application. Mongoose, an elegant MongoDB object modeling tool for Node.js, facilitates the interaction between the application and the database. This combination allows for efficient storage and retrieval of weather data, providing a scalable and flexible solution for handling weather information.
- **Node.js and Express JS:** Node.js is utilized as the server-side runtime for the Weather Forecasting Web Application. Its non-blocking, event-driven

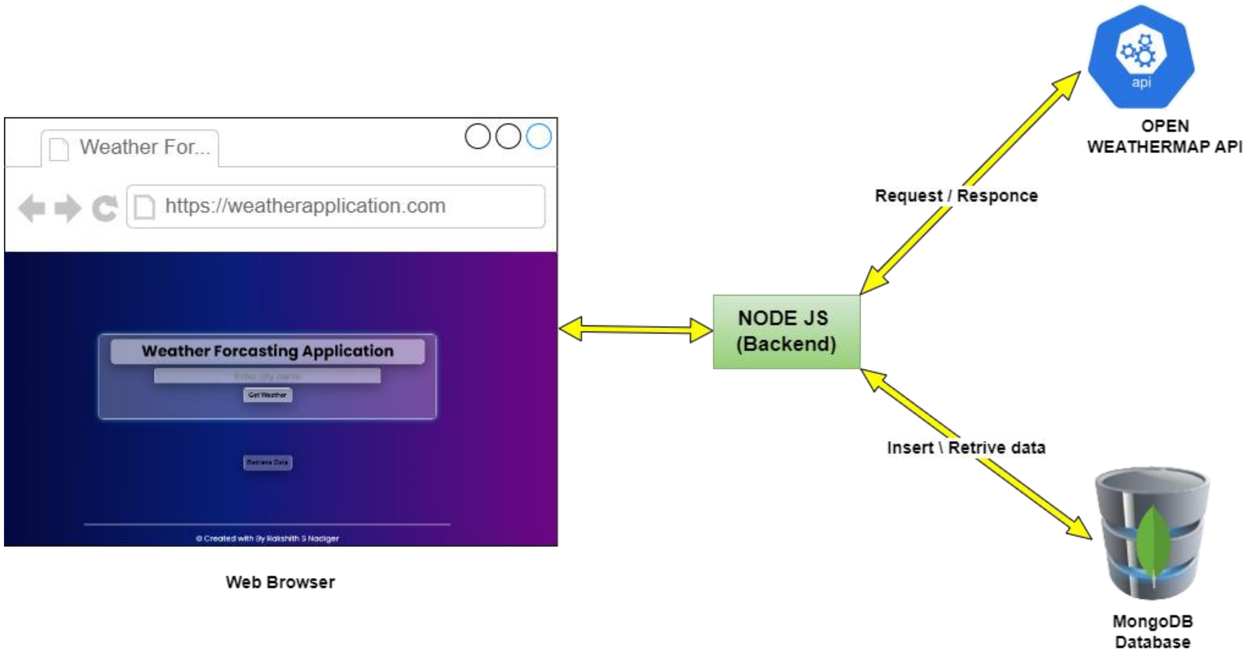
architecture ensures fast and responsive data processing. Express JS is used for routing the requests.

- **Angular.js**: Angular.js is used for the frontend development of the application. Angular.js enables the creation of a single-page application, allowing users to navigate through weather information seamlessly.
- **HTML and CSS:** HTML and CSS are fundamental technologies used for structuring and styling the user interface of the Weather Forecasting Web Application. HTML provides the markup structure, while CSS is responsible for the visual styling, ensuring a clean and user-friendly design. The combination of these technologies contributes to the overall aesthetics and usability of the application.

Use Cases:

- **Personal Weather Updates:** Users can easily check current weather conditions and receive accurate forecasts for their location, aiding in daily planning and activities.
- **Event Planning:** Individuals organizing events can utilize the application to make informed decisions based on upcoming weather forecasts, ensuring the success of outdoor activities.
- **Agricultural Planning:** Farmers can benefit from the historical weather data and forecasts to plan planting, harvesting, and other agricultural activities, optimizing crop yields.
- **Logistics and Transportation:** Companies involved in logistics and transportation can use the application to anticipate weather-related disruptions, optimizing routes and schedules.
- **Emergency Preparedness:** Governments and emergency services can leverage the application for timely weather alerts and better preparedness in the face of adverse weather conditions.

Methodology:



Architecture

The user will enter the name of city and clicks on "Get Weather" button. Then the request is sent to backend (NodeJS server). The request is forwarded to Openweathermap API. The API sends back the response to NodeJS server and server sends the data to webpage (browser). Then, weather forecast data and weekly forecast is displayed on the screen. At the same time, Node JS server will connect to Mongo DB database and stores the data received from Openweathermap API. It will store the data with time stamp. Whenever the user clicks "Retrieve data" button, The data which is stored in MongoDB will be retrieved and displayed to the user. The background of the webpage will change based on the retrieved weather. This is an additional functionality added to the website.

Implementation:

The Weather Forecasting Web Application is implemented using HTML, CSS, Angular.js for the frontend, and Node.js with MongoDB for the backend. Below, I provide an overview of the key components and functionalities of the application.

Frontend (HTML, CSS, Angular.js):

The HTML file (index.html) serves as the main structure of the application. It includes sections for displaying current weather information, a weekly forecast table, and a data retrieval feature. The Angular.js framework is employed to enhance interactivity and dynamically update the user interface.

The main features of the HTML file include:

Weather Display Section: Users can enter a city name, click the "Get Weather" button, and view real-time weather information, including temperature, date, weather description, humidity, and wind speed.

Weekly Weather Forecast Table: A table is presented with columns for date, weather description, and temperature, providing a concise overview of the week's weather forecast.

Data Retrieval Section: Users can click the "Retrieve Data" button to fetch and display previously saved weather data from MongoDB. The retrieved data is presented in a table, including date, city, temperature, and weather description.

Backend (Node.js, MongoDB):

The server-side code is implemented using Node.js, Express, and Mongoose for MongoDB interaction. The server handles requests from the frontend, communicates with the MongoDB database to store and retrieve weather data, and serves static files.

The main features of the Node.js server file (server.js) include:

Express Setup: Configuring the Express framework to serve static files and redirect all routes to the main entry point (index.html).

MongoDB Connection: Establishing a connection to the MongoDB database hosted on MongoDB Atlas. A schema and model are defined to structure and interact with weather data in the database.

Data Insertion Endpoint: Implementing a route (*/insertData*) to handle the insertion of weather data into MongoDB. Data sent from the frontend is saved as documents in the specified collection.

Data Retrieval Endpoint: Creating an API endpoint (*/api/retrieve*) to retrieve all weather data from MongoDB. CORS is used to allow cross-origin resource sharing.

Server Start: Starting the server on a specified port.

Frontend-Backend Interaction: The Angular.js file (weatherfetch.js) contains the Angular module and controller. It facilitates the interaction between the frontend and backend. Key functionalities include:

Weather Data Retrieval: Making HTTP GET requests to the OpenWeatherMap API to fetch current and weekly weather data based on the user's input.

Data Storage in MongoDB: Sending HTTP POST requests to the Node.js server to store weather data in the MongoDB database.

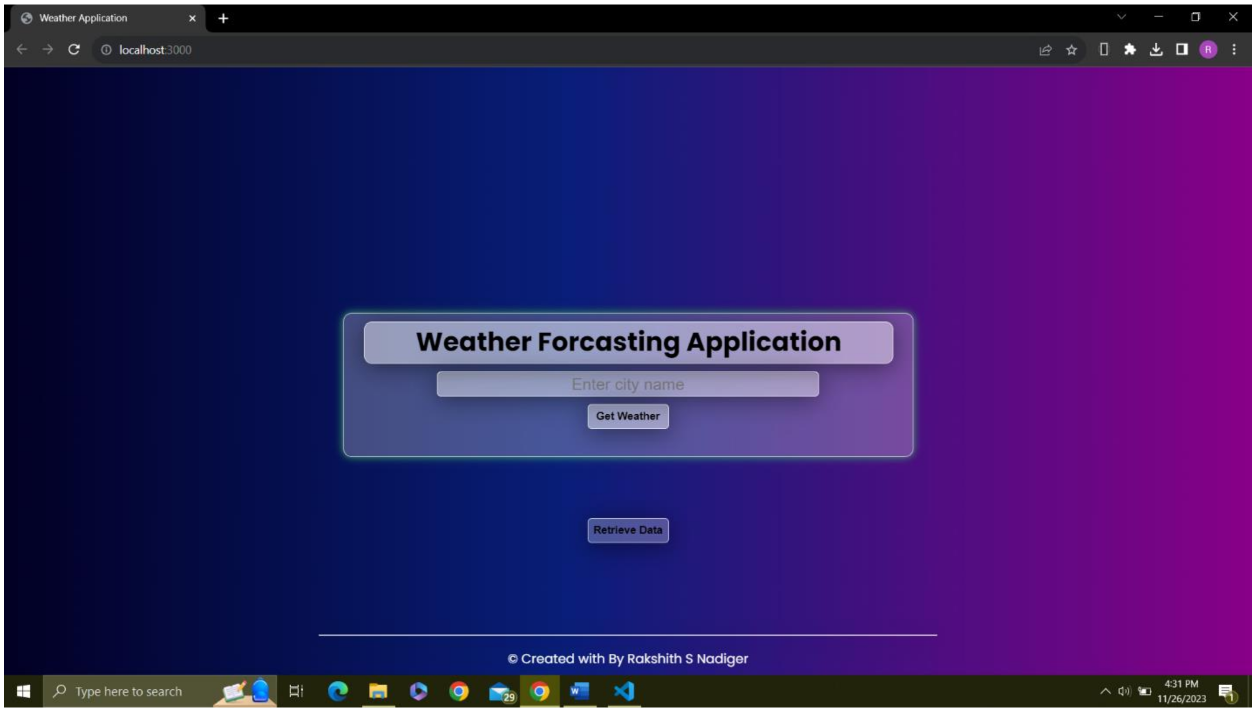
Stylesheet Update: Dynamically changing the stylesheet based on the weather conditions to provide a visually appealing user interface.

Visual Elements:

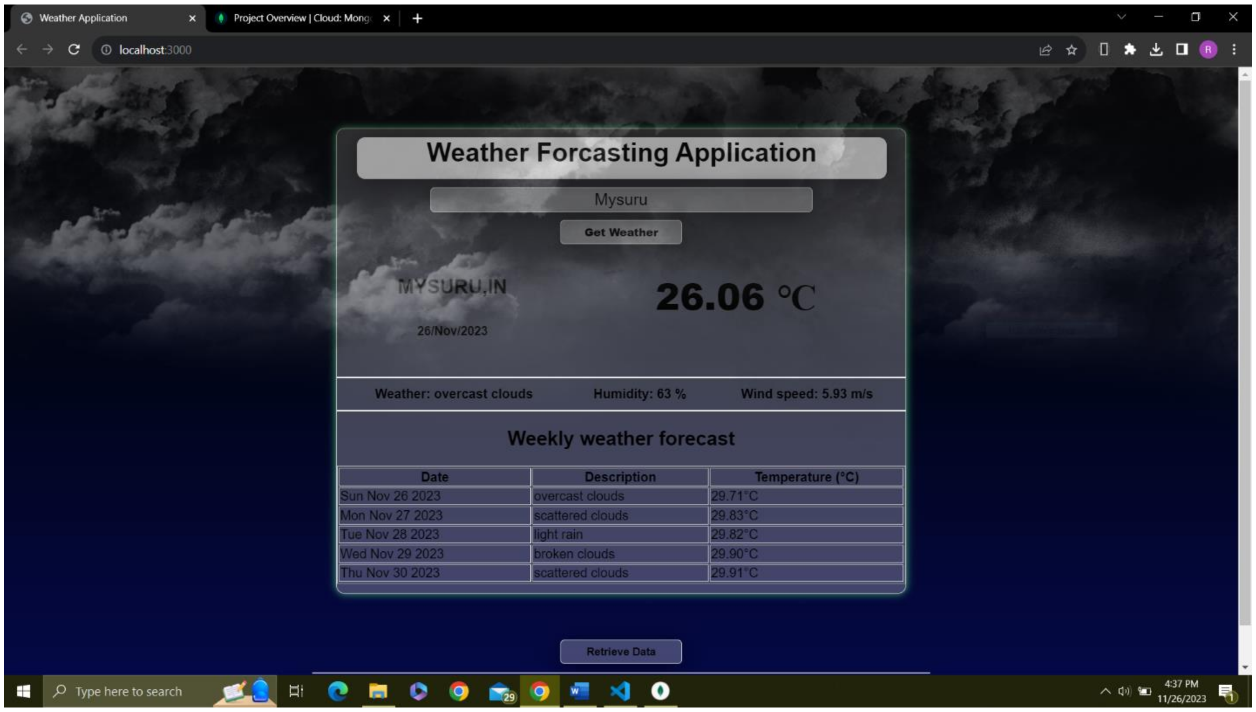
The application incorporates visual elements using CSS to enhance the user experience. I have made four different stylesheets to display four different kinds of weather. It includes animations for rains, clouds, fog, and a sun based on the weather conditions.

Screenshots of Implementation

Home Page:



Weather Forecast (with cloud background):



MYSURU,IN

26/Nov/2023

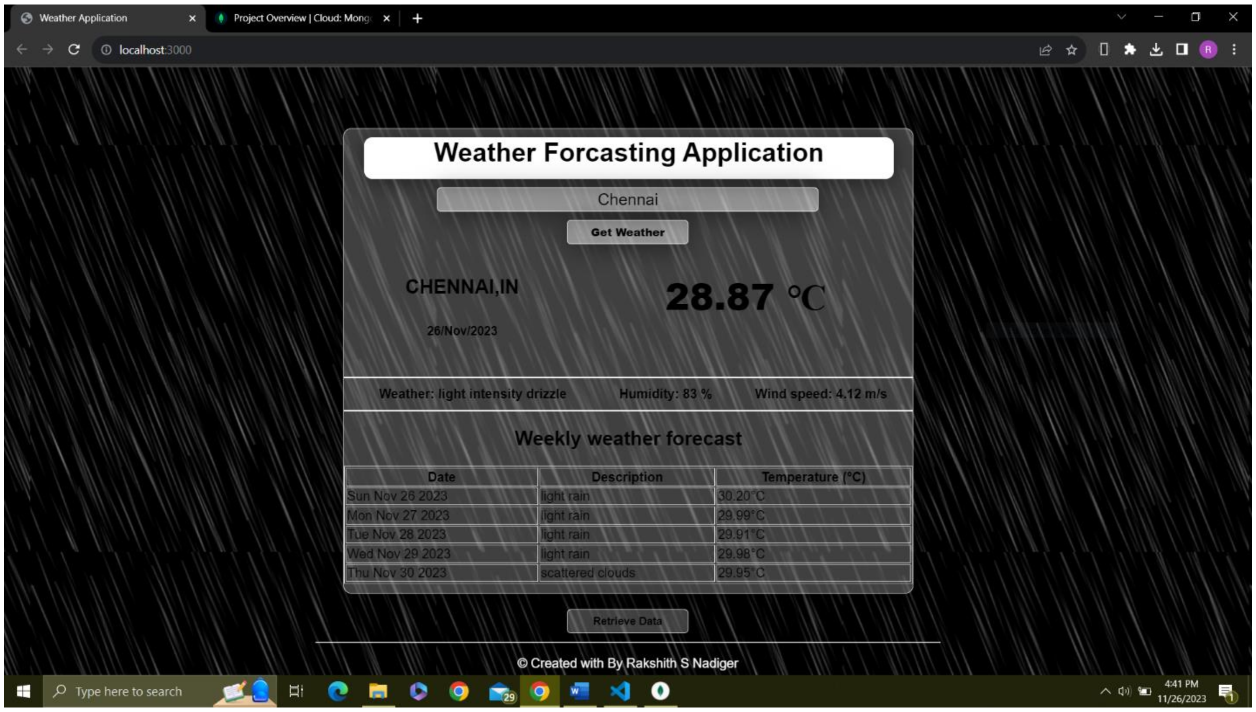
26.06 °C

Weather: overcast clouds Humidity: 63 % Wind speed: 5.93 m/s

Weekly weather forecast

Date	Description	Temperature (°C)
Sun Nov 26 2023	overcast clouds	29.71°C
Mon Nov 27 2023	scattered clouds	29.83°C
Tue Nov 28 2023	light rain	29.82°C
Wed Nov 29 2023	broken clouds	29.90°C
Thu Nov 30 2023	scattered clouds	29.91°C

Weather Forecast (with rainy background):



Weather Forcasting Application

Chennai

CHENNAI,IN 28.87 ℃

26/Nov/2023

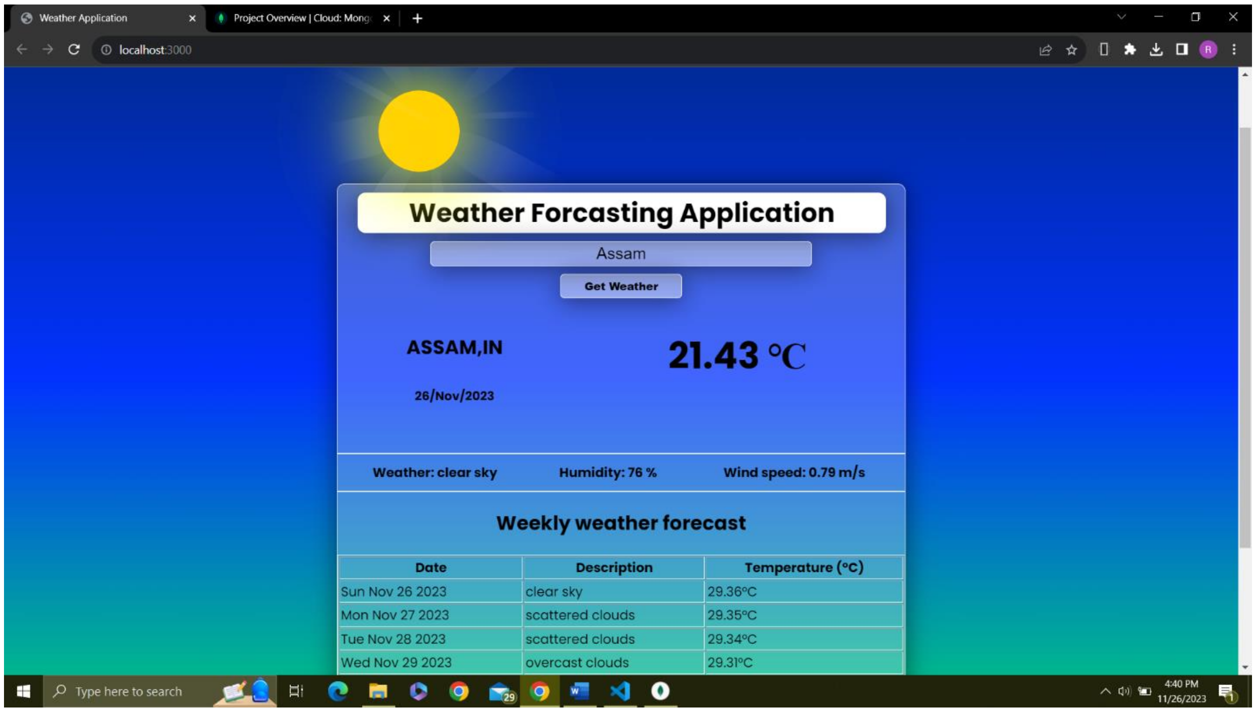
Weather: light intensity drizzle Humidity: 83 % Wind speed: 4.12 m/s

Weekly weather forecast

Date	Description	Temperature (°C)
Sun Nov 26 2023	light rain	30.20°C
Mon Nov 27 2023	light rain	29.99°C
Tue Nov 28 2023	light rain	29.91°C
Wed Nov 29 2023	light rain	29.98°C
Thu Nov 30 2023	scattered clouds	29.95°C

© Created with By Rakshith S Nadiger

Weather Forecast (with sunny background):



Weather Forcasting Application

Assam

ASSAM,IN 21.43 ℃

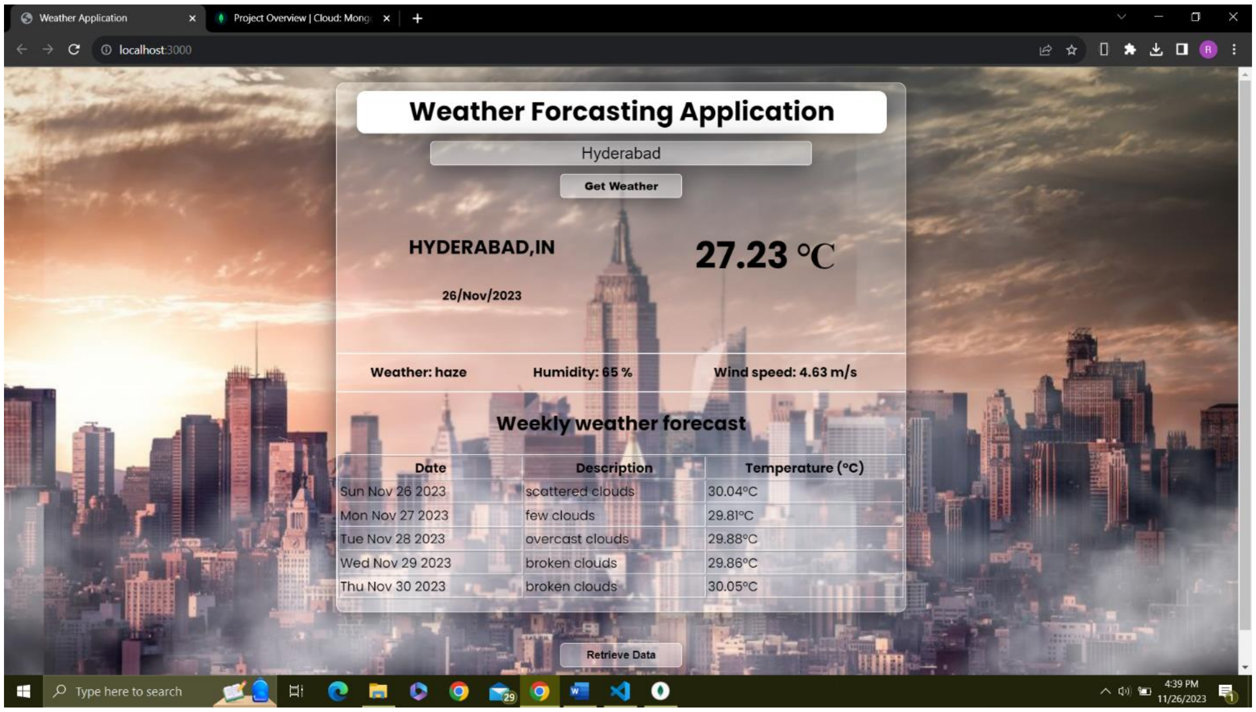
26/Nov/2023

Weather: clear sky Humidity: 76 % Wind speed: 0.79 m/s

Weekly weather forecast

Date	Description	Temperature (°C)
Sun Nov 26 2023	clear sky	29.36°C
Mon Nov 27 2023	scattered clouds	29.35°C
Tue Nov 28 2023	scattered clouds	29.34°C
Wed Nov 29 2023	overcast clouds	29.31°C

Weather Forecast (with fog background):



Weather Forcasting Application

Hyderabad

HYDERABAD,IN

27.23 °C

26/Nov/2023

Weather: haze

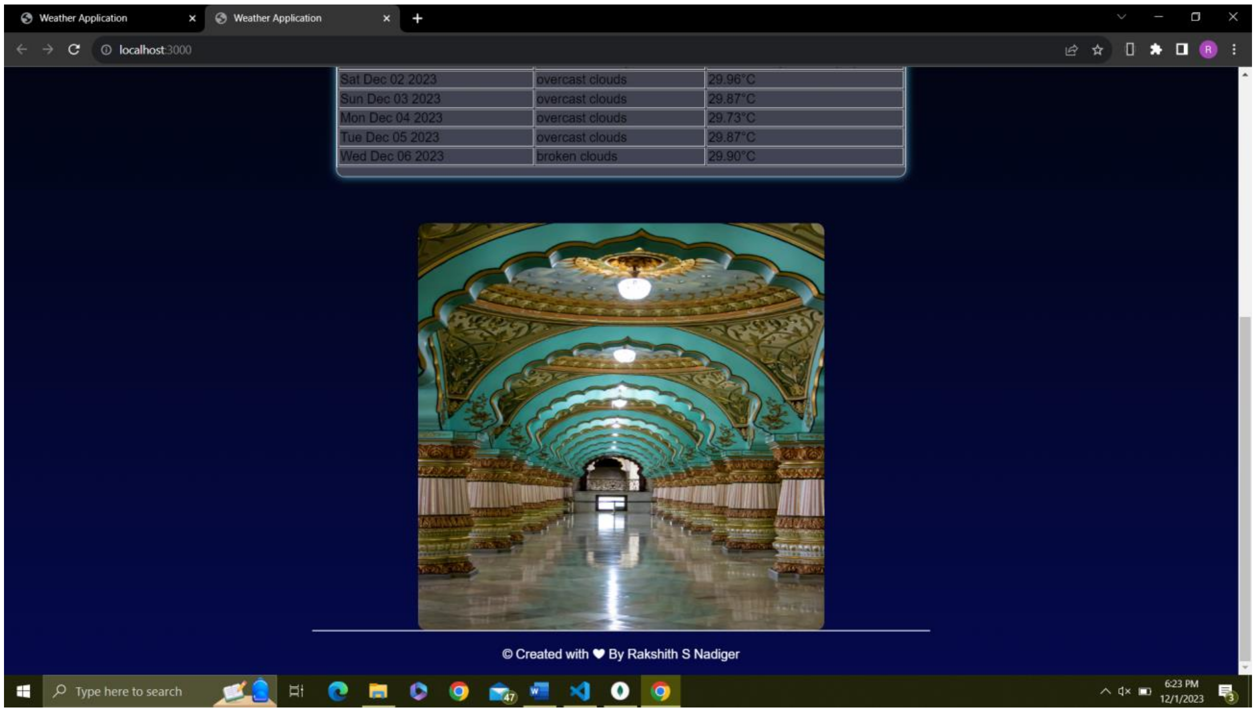
Humidity: 65 %

Wind speed: 4.63 m/s

Weekly weather forecast

Date	Description	Temperature (°C)
Sun Nov 26 2023	scattered clouds	30.04°C
Mon Nov 27 2023	few clouds	29.81°C
Tue Nov 28 2023	overcast clouds	29.88°C
Wed Nov 29 2023	broken clouds	29.86°C
Thu Nov 30 2023	broken clouds	30.05°C

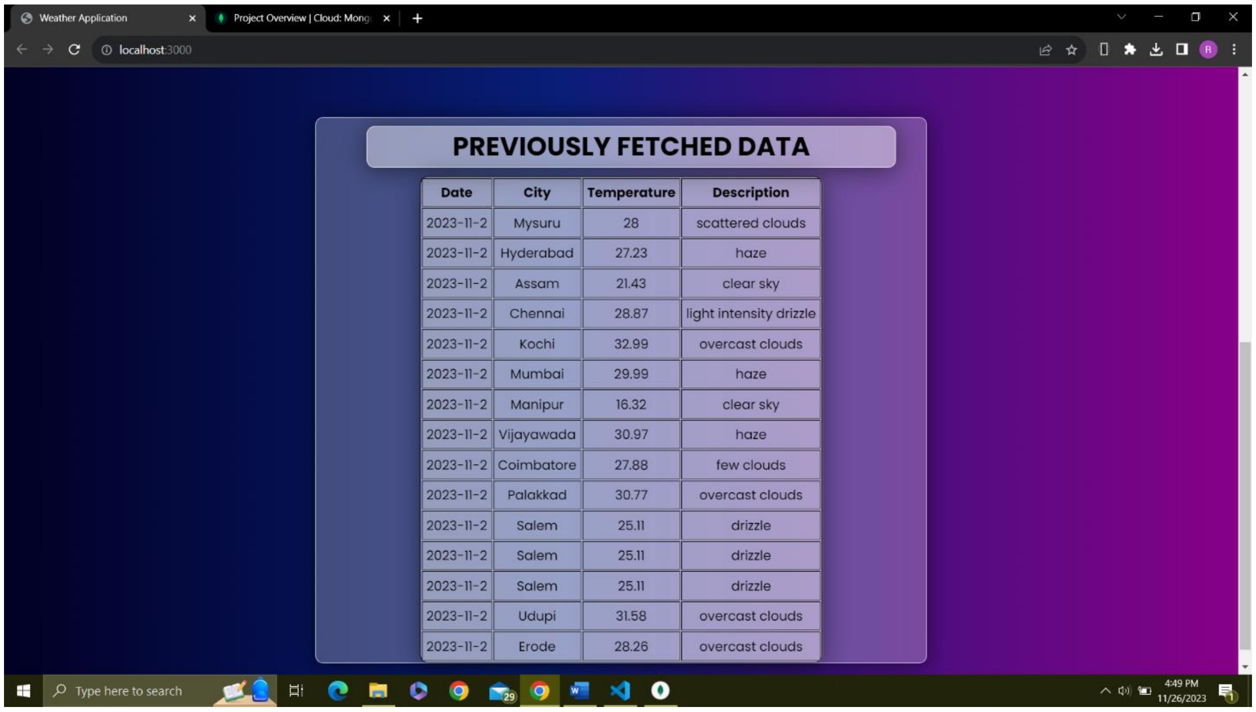
Weather forecasting with image of city:



Sat Dec 02 2023	overcast clouds	29.96°C
Sun Dec 03 2023	overcast clouds	29.87°C
Mon Dec 04 2023	overcast clouds	29.73°C
Tue Dec 05 2023	overcast clouds	29.87°C
Wed Dec 06 2023	broken clouds	29.90°C

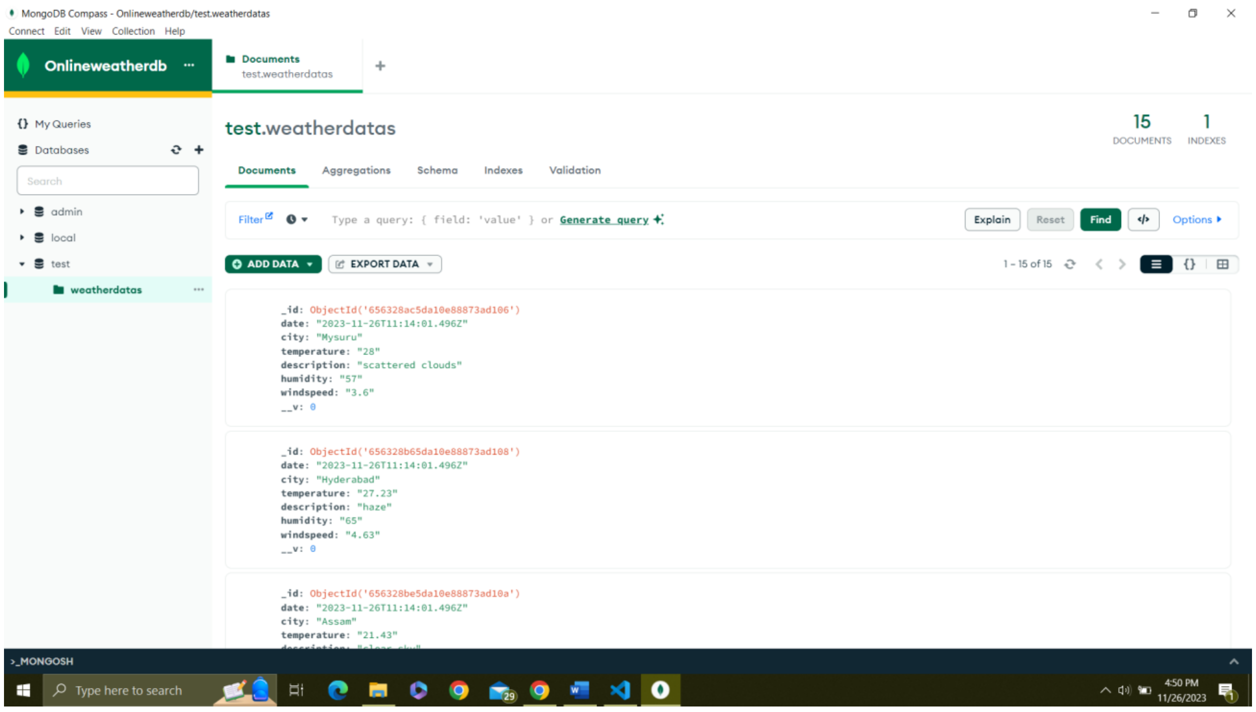
© Created with ❤ By Rakshith S Nadiger

Data Retrieved from Database:



Date	City	Temperature	Description
2023-11-2	Mysuru	28	scattered clouds
2023-11-2	Hyderabad	27.23	haze
2023-11-2	Assam	21.43	clear sky
2023-11-2	Chennai	28.87	light intensity drizzle
2023-11-2	Kochi	32.99	overcast clouds
2023-11-2	Mumbai	29.99	haze
2023-11-2	Manipur	16.32	clear sky
2023-11-2	Vijayawada	30.97	haze
2023-11-2	Coimbatore	27.88	few clouds
2023-11-2	Palakkad	30.77	overcast clouds
2023-11-2	Salem	25.11	drizzle
2023-11-2	Salem	25.11	drizzle
2023-11-2	Salem	25.11	drizzle
2023-11-2	Udupi	31.58	overcast clouds
2023-11-2	Erode	28.26	overcast clouds

MongoDB database:



```
_id: ObjectId('656328ac5da10e88873ad106')
date: "2023-11-26T11:14:01.496Z"
city: "Mysuru"
temperature: "28"
description: "scattered clouds"
humidity: "57"
windspeed: "3.6"
__v: 0

_id: ObjectId('656328b65da10e88873ad108')
date: "2023-11-26T11:14:01.496Z"
city: "Hyderabad"
temperature: "27.23"
description: "haze"
humidity: "65"
windspeed: "4.63"
__v: 0

_id: ObjectId('656328be5da10e88873ad10a')
date: "2023-11-26T11:14:01.496Z"
city: "Assam"
temperature: "21.43"
```

Conclusion:

The Weather Forecasting Web Application successfully integrates frontend and backend technologies, offering users a responsive and feature-rich platform for accessing real-time weather information. The use of Angular.js, Node.js, MongoDB, and external APIs demonstrates a modern and scalable approach to weather forecasting. The application's visual elements contribute to an engaging user interface, making it a comprehensive solution for weather-related needs.