****

A PROJECT REPORT

ON

**OPEN WEATHER API**

Submitted By:

ARYA SHARMA - ‘C’ - 2115000209

VEDANT NIGAM - ‘C’ – 2115001107

JAGRATI AGRAWAL - ‘D ’ – 2115000485

***In partian fulfilment for the award of the degree of***

**Bachelor of Engineering**

**IN**

Computer Science

#### BONAFIDE CERTIFICATE

Certified that this project report **“OPEN WEATHER API”** is the bonafide work of “**Arya Sharma, Vedant Nigam and Jagrati Agrawal”** who carried out the project work under my/our supervision.

|  |  |
| --- | --- |
| **SIGNATURE**    **HEAD OF THE DEPARTMENT** | **SIGNATURE**    **SUPERVISOR**  Technical Trainer  Training and development |

Submitted for the project viva-voce examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

#### ACKNOWLEDGEMENT

The project work in this report is an outcome of continuous work over a period and drew intellectual support from various sources. We would like to articulate our profound gratitude and to all those people who extended their wholehearted co-operation and have helped us in completing this project successfully.

We are thankful to our mentor Ms. Ruchi Talwar ma’am for teaching and assisting us with the new technology and guiding us at every step, and it wouldn’t have been possible for us to finish the project in such short period of time if it were not for his motivation. I would also like to thank all the faculty members who were directly and indirectly contributed in the completion of our project.

#### TABLE OF CONTENTS

[List of Figures i](#_Toc10916)

[Abstract ii](#_Toc10918)

##### **Chapter 1 …………… Introduction**

##### **1.1 …………………… Client Identification**

**1.2 …………………… Identification of Problem**

##### **1.3 …………………… Identification of Task**

**1.4 ………………........ Timeline**

**1.5 …………………… Organization of the Report**

##### **Chapter 2 …………… Literature Survey**

##### **2.1 …………………… Analysis**

##### **Chapter 3 …………… Design Flow / Processes**

##### **3.1 …………………… Concept Generation**

##### **3.2 …………………… Evaluation & Selection of specifications / Features**

##### **3.3 …………………… Analysis of Specifications and features**

##### **3.4 …………………… Design Flow**

##### **3.5 …………………… Implementation Plan**

**3.5.1 …………………. Front-end tech**

**3.5.2 …………………. Back-end tech**

##### **Chapter 4 …………… Result Analysis and Validation**

##### **4.1 ……………………**

##### **4.2 ……………………**

##### **4.3 ……………………**

##### **4.4 ……………………**

##### **4.5 ……………………**

##### **Chapter 5 …………… Conclusion & Future Work**

##### **5.1 …………………… Conclusion**

##### **5.2 …………………… Future Work**

##### **REFERENCES**

**ABSTRACT**

Mini project is the requirement for all engineering students in order to complete their Bachelor of Engineering degree at the GLA University, Mathura. This project is a very important for us, since it complements both the academic and professional aspects of the engineering education. It exposed us to the practical experience and actual working environment, where we were able to develop our skills and capabilities, as well as enhancing our intellectual and emotional personal. The Mini Project also provides strong linkages between university-industries that shall pave opportunities for "smart partnerships" and industrially driven research. The outcomes of the EIT that are mainly based on the assessment covering the company's and university's evaluation will provide the feedback for student’s performance after 75% completion of their engineering study. The remarks from the companies on the students will very much helpful for the university to have a continuous quality improvement especially on curriculum practiced.

**CHAPTER 1**

#### INTRODUCTION

* 1. **Client Identification/Need Identification/Identification of relevant**

The OpenWeather API requires an API key for client identification. When you sign up for an account on the OpenWeather website and subscribe to one of their API plans, you will receive an API key that you need to include in your API requests. This key is used to authenticate your requests and identify you as a valid user of the OpenWeatherAPI.

**1.2. Identification of Problem**

The purpose of our project is used to authenticate Client requests and identify Client as a valid user of the OpenWeatherAPI. TheOpenWeatherAPI requires an API key for client identification. When you sign up for an account on the OpenWeather website and subscribe to one of their API plans, you will receive an API key that you need to include in your API requests.

**1.3. Identification of Task**

The Open Weather API project Allow users to customize units (e.g., Celsius/Fahrenheit) for weather parameters.It allow user to Acquire an API key from OpenWeather to authenticate requests.

**1.4. Timeline**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task / Period | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 |
| Project Selection |  |  |  |  |  |
| Mentor Allocation |  |  |  |  |  |
| Project Planning |  |  |  |  |  |
| Prototype and Designing |  |  |  |  |  |
| Documentation |  |  |  |  |  |

**1.5. Organization of the Report**

GLA University

**CHAPTER 2**

#### LITERATURE SURVEY

The Open Weather API is a web service that provides user to access to a wide range of weather data.It allow users to integrate weather information into their applications,websites, or services,offering and Forecasted weather data for locations worldwide.

The OpenWeatherAPI holds significance due to several key factors:

Real-Time Weather Data : OpenWeather API provides access to up-to-date and accurate weather information, allowing developers to integrate real-time weather data into applications and services.

Global Coverage: The API covers a vast range of geographic locations worldwide, making it valuable for applications with a global user base or those requiring weather data for diverse regions.

Diverse Applications: Its versatility allows developers to incorporate weather data into a wide range of applications, from simple weather widgets on websites to complex solutions in mobile apps, IoT devices, and smart systems.

Weather Forecasting: Developers and researchers can leverage the API for weather forecasting, enabling applications to provide users with reliable predictions and insights into future weather conditions.

Educational Use: OpenWeatherMap API is often used in educational contexts, allowing students and developers to learn about API integration, data handling, and real-world application development through weather-related projects.

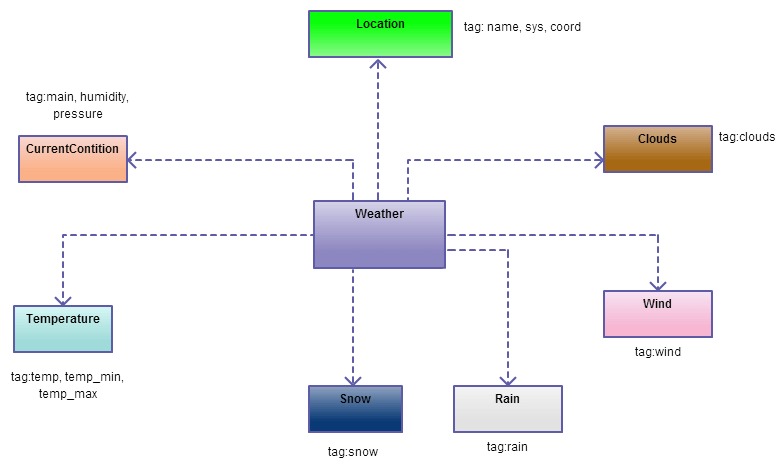
**CHAPTER 3**

#### DESIGN FLOW/PROCESS

**3.1. Concept Generation**

Develop a personalized weather assistant leveraging the OpenWeather API, providing users with tailored weather updates and recommendations based on their location. The application could include features such as daily weather forecasts, clothing suggestions, and alerts for severe weather conditions. Integrating natural language processing, the assistant can respond to user queries about upcoming weather events and offer insights for planning outdoor activities. Additionally, consider incorporating machine learning algorithms to enhance accuracy in predicting local weather patterns, creating a user-centric and intelligent weather companion

##### **3.2. Evaluation & Selection of Specifications/Features**



**3.3. Analysis of Specifications and Features**

We have worked on responsive frameworks to ensure the compatibility of our Open Weather Api across various browsers and devices.Other than that we have used optimized search algorithms for quickly finding out the weather of the location entered by the user and providing relevant results with less delay.

**3.4. Design Flow**

Users provide location details, either manually selecting a city or enabling geolocation for automatic retrieval.

**API Key Authentication:**

The application authenticates the OpenWeather API by including the unique API key in the request headers.

**Request Processing:**

The application sends a request to the appropriate OpenWeatherMap API endpoint, specifying the desired weather data, such as current conditions or forecasts.

**API Response:**

OpenWeatherMap processes the request and returns a JSON response containing the requested weather information.

**Data Parsing:**

The application parses the JSON response, extracting relevant weather parameters such as temperature, humidity, and wind speed.

**User Interface Display:**

The extracted weather data is dynamically displayed on the user interface, incorporating visual elements for a user-friendly presentation.

**Optional Features:**

Optional features, such as unit customization, historical data retrieval, and additional services like weather alerts, can be implemented based on user preferences.

**Error Handling:**

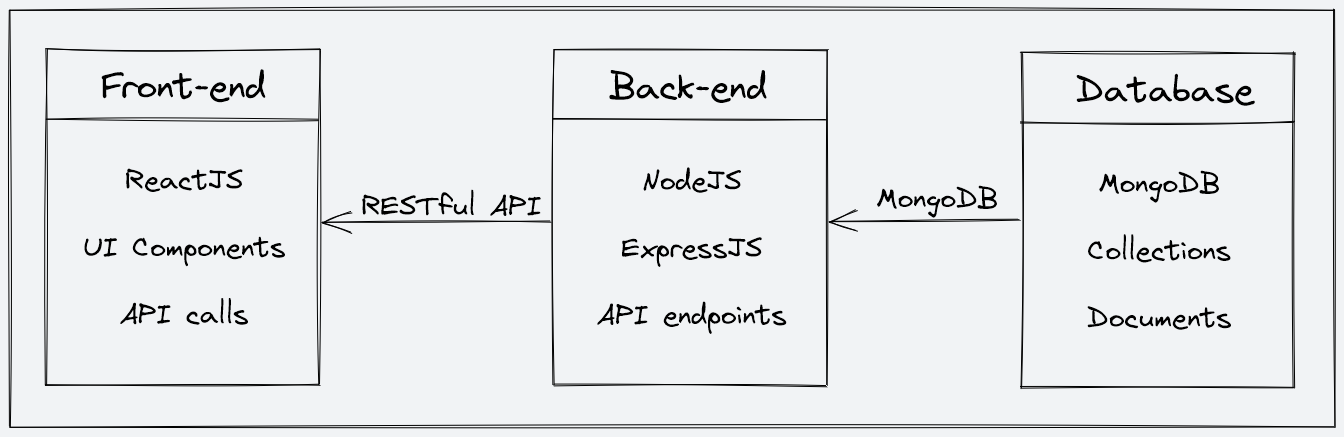
Robust error-handling mechanisms are in place to manage API request failures or unexpected responses, ensuring a smooth user experience.

**User Interaction:**

Users can interact with the displayed weather information, asking for more details, accessing historical data, or configuring preferences.

Continuous Updates:

The application can implement periodic updates by making scheduled API requests to keep weather information current.



Architecture Diagram

**3.5. Implementation Plan**

We have implemented the designing of our Open Weather API with the latest tools available to maintain the compatibility of our website with the modern tech.

Below is the description of the tech stack that we have used in our project.

Technologies used:

Our project is based on front-end and back-end technologies.

**3.5.1.** For front-end we are using React JS.

**React JS** is a JavaScript library that is widely used for building user interfaces is one of the most popular front-end development frameworks, and it provides a rich set of tools and features that allow developers to customize the behavior and appearance of their user interfaces. With React, developers can create unique and engaging user experiences that are tailored to the needs of their users.

React is useful in a project because it allows developers to build complex user interfaces with ease. By using React, developers can create reusable UI components that can be combined to create a variety of different layouts and user experiences. This makes it easier to build *scalable* and *maintainable web applications*.

One of the key features of React is its ability to provide real-time updates to the user interface. By using a technique called "virtual DOM," React can update the user interface without having to reload the entire page. This helps to create a seamless user experience that is *faster* and *more responsive* than traditional web applications.

Another benefit of using React in a project is its ability to work with other JavaScript libraries and frameworks. React is designed to be modular and can be easily integrated with other libraries and frameworks, such as Redux, React Router, and Axios. This allows developers to build powerful and complex web applications that can meet the needs of their users.

React is also useful in a project because it is highly customizable. React provides a rich set of tools and features that allow developers to customize the behaviour and appearance of their user interfaces. With React, developers can create unique and engaging user experiences that are tailored to the needs of their users.

In conclusion, React is a powerful and flexible JavaScript library that is useful in a wide range of web development projects. Its ability to provide real-time updates, work with other libraries and frameworks, and provide a high degree of customization makes it an essential tool for building scalable and maintainable web applications. Whether you're building a small website or a large web application, React is a valuable tool that can help you create a great user experience for your audience.

**3.5.2** For back-end, Node JS, Express JS, MongoDB and Mongoose are used.

1. **Node JS** is an open-source, cross-platform, back-end JavaScript runtime environment that allows developers to build fast and scalable applications. Node JS interacts with the server to handle the client requests while utilizing the commands that are being designed in Express JS.

It uses an event-driven, non-blocking I/O model that makes it highly efficient for building real-time applications, especially those that involve large amounts of data.

It is useful in a project in several ways, here are some of them:

1. **High performance**: It is built on the V8 JavaScript engine, which is the same engine used by Google Chrome. This makes Node.js *highly efficient* and allows it to handle a large number of requests without affecting the performance of the application.
2. **Scalability**: It is highly scalable, which means *it can handle a large number of connections simultaneously without slowing down*. This makes it ideal for building applications that need to handle a large amount of traffic.
3. **Single language**: With Node.js, developers can use JavaScript on both the front-end and back-end, making it easy to develop full-stack applications using a single language.
4. **Large community**: Node.js has a large and active community of developers who contribute to the development of various libraries and modules. This makes it easy for developers to find solutions to their problems and improve the quality of their applications.

Node.js is a powerful tool that can help developers build fast, scalable, and efficient applications. Its ease of use and large community make it an attractive choice for developers who want to build full-stack applications using a single language.

1. **Express JS** is a popular open-source web application framework built on top of Node.js. It provides a set of tools and utilities for building web applications and APIs in Node JS. Express JS is known for its simplicity, flexibility, and scalability, and it is widely used by developers to build web applications, APIs, and micro services.

It provides light-weight framework of Node JS and with it we can perform crud operations and create our own server easily fulfilling our project requirements. It also acts as a *powerful middleware for a range of operations such as authentication, logging and error handling*.

Some of the key advantages of using Express JS includes:

1. **Simplicity**: Express JS is a lightweight framework that provides a simple and intuitive API for building web applications and APIs. It has a minimalist design philosophy that emphasizes simplicity and ease-of-use.
2. **Flexibility**: It is highly flexible and customizable, allowing developers to build web applications and APIs that meet their specific needs. It provides a modular architecture that allows developers to use only the components they need and replace or extend them as needed.
3. **Scalability**: It is highly scalable and can handle high volumes of traffic and requests. It provides a non-blocking I/O model that enables asynchronous processing and supports clustering for horizontal scaling.
4. **Middleware**: It provides a powerful middleware system that allows developers to easily add features and functionality to their web applications and APIs. Middleware functions can be used for a wide range of purposes, such as authentication, logging, error handling, and more.
5. **Routing**: It also provides a powerful routing system that allows developers to easily define the routes for their web applications and APIs. It supports a wide range of HTTP methods, such as GET, POST, PUT, DELETE, and more.
6. **Integration**: Express.js integrates seamlessly with other popular Node.js libraries and tools, such as MongoDB, Socket.io, and more.

Express.js is a powerful and flexible web application framework that provides a wide range of benefits for developers. It enables developers to build fast, scalable, and customizable web applications and APIs with ease, making it an ideal choice for projects of all sizes and complexities.

1. **MongoDB** is a popular NoSQL document-oriented database that provides a flexible, scalable, and high-performance solution for handling unstructured or semi-structured data. It stores data in JSON-like documents with dynamic schemas, making it easy to store and retrieve complex data structures. Unlike traditional relational databases, MongoDB does not requires predefined tables to store data, which makes it ideal for handling large amount of data.

Some of the key features of MongoDB are:

1. **Schema-less Database**: It is the great feature provided by the MongoDB. A Schema-less database means one collection can hold different types of documents in it. Or in other words, in the MongoDB database, a single collection *can hold multiple documents and these documents may consist of the different numbers of fields, content, and size*. It is not necessary that the one document is similar to another document like in the relational databases. Due to this cool feature, MongoDB provides great flexibility to databases.
2. **Document Oriented**: In MongoDB, all *the data stored in the documents* instead of tables like in RDBMS. In these documents, the data is stored in fields (key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.
3. **Indexing**: In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then database search each document with the specified query which takes lots of time and not so efficient.
4. **Scalability**: MongoDB provides horizontal scalability with the help of sharding. Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database.
5. **Replication**: MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.
6. **Aggregation**: It allows to perform operations on the grouped data and get a single result or computed result. It is similar to the SQL GROUPBY clause. It provides three different aggregations i.e, aggregation pipeline, map-reduce function, and single-purpose aggregation methods
7. **High Performance**: The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.
8. **Mongoose** is an Object Data Modeling (ODM) library for MongoDB. It defines a strongly-typed-schema, with default values and schema validations which are later mapped to a MongoDB document. It provides an incredible amount of functionality around creating and working with schemas. Mongoose currently contains eight Schema Types that a property is saved as when it is persisted to MongoDB. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

**CHAPTER 4**

#### RESULT ANALYSIS AND VALIDATION

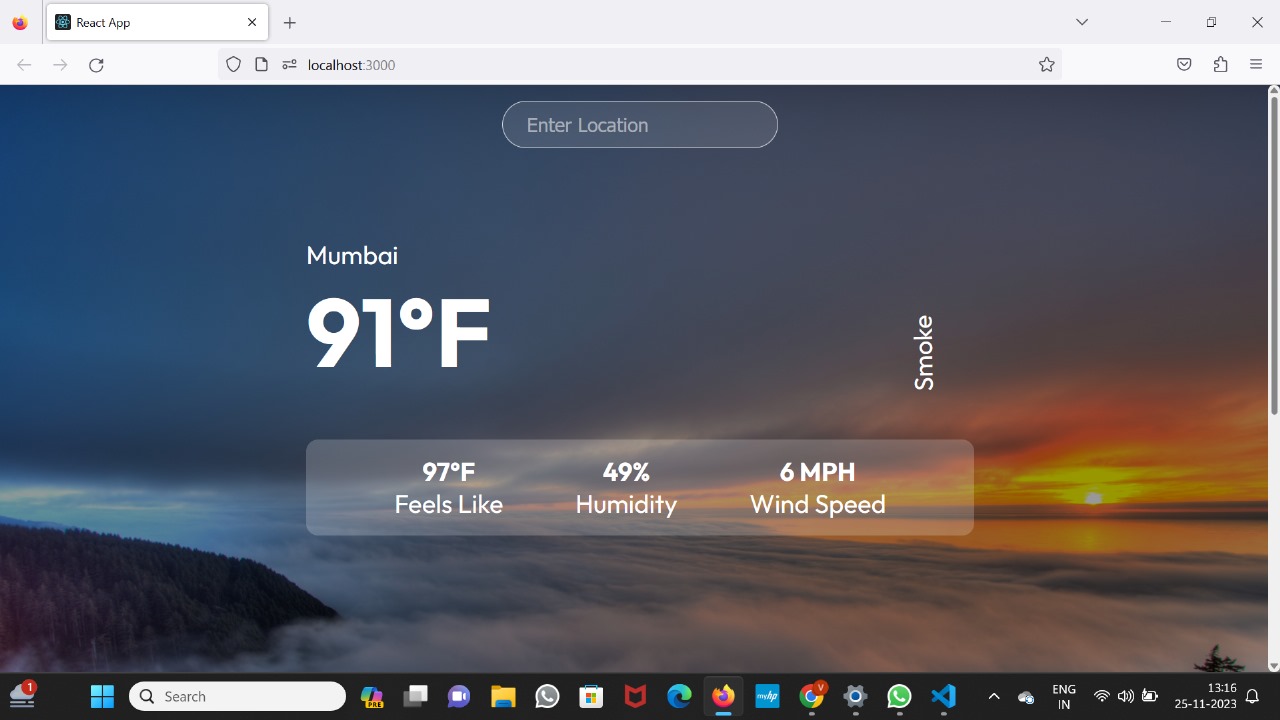


FIG- 1: OUTPUT WEATHER PAGE

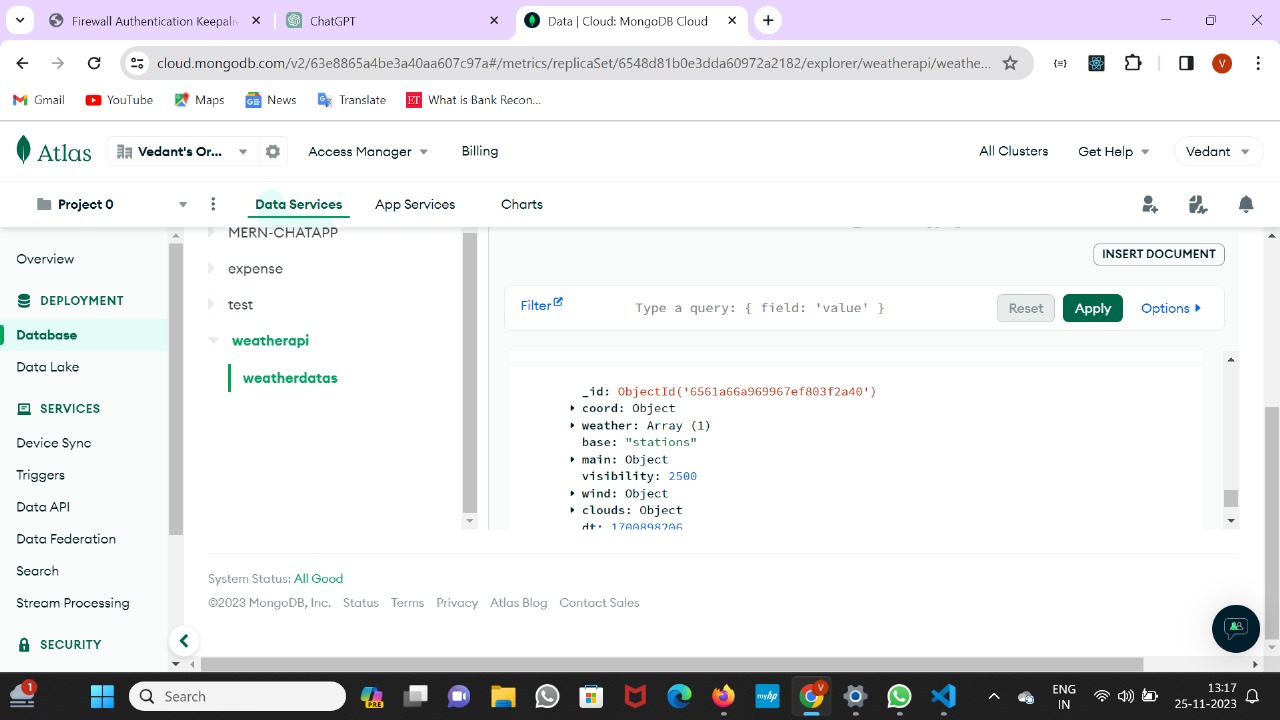


FIG-2: MONGODB ATLAS PAGE

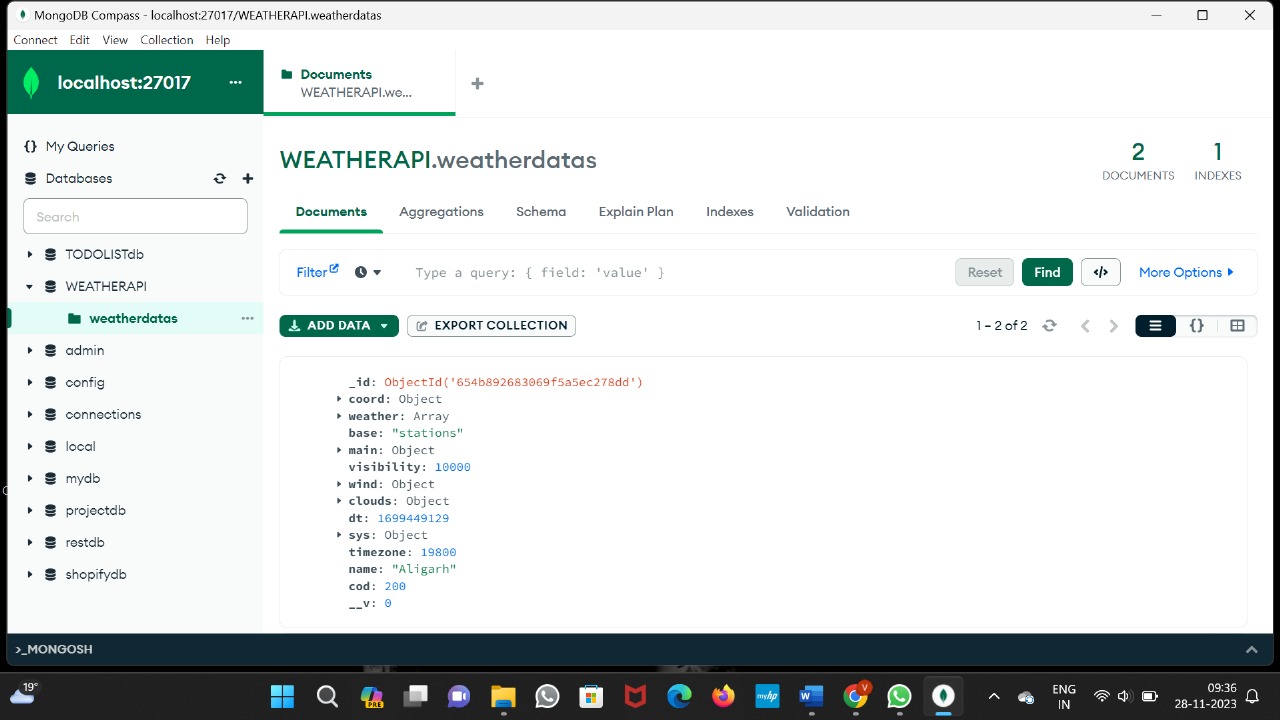


FIG-3: MONGODB COMPASS PAGE

**CHAPTER 5**

#### CONCLUSION AND FUTURE WORK

##### **6.1. CONCLUSION**

##### In conclusion, the Open Weather API stands as a versatile and valuable tool, offering developers global access to real-time and forecasted weather data. Its robust features, including customization options and community support, make it pivotal in diverse applications, from mobile apps to IoT devices. The API's significance lies in its ability to empower decision-making processes, enhance user experiences, and contribute to educational initiatives in the realm of weather-related applications.

##### **6.2. FUTURE WORK**

**Advanced Forecasting Models:** Implementing more sophisticated weather prediction models, incorporating machine learning and advanced algorithms for enhanced accuracy in long-term forecasts.

**Climate Change Analysis:** Expanding capabilities to support climate change research by providing historical data analysis and long-term climate trend predictions.

**Integration with Emerging Technologies:** Exploring integration possibilities with emerging technologies such as augmented reality (AR) or virtual reality (VR) for immersive weather experiences.

**REFERENCES**

* <https://openweathermap.org/api>
* https://tailwindui.com/templates
* https://www.youtube.com/@codecommerce