**Weather Data Base**

Information about weather is very important especially if we want to anticipate weather changes that can affect businesses such as street hawkers, farming, aviation, entertainment etc. Having information about weather beforehand can help people prepare to avoid financial, mental, physical or health losses. At a very simple and ground root level, we, as a human, wants to know whether to carry an umbrella or a sunglass with us while going out. People also want to know how the weather will be when they will be planning a vacation. On another side, our data scientists, academician, and researcher also require the historical and current weather data for predictive analysis, for forecasting the chances of thunderstorm or tsunami at a point on globe to serve the mankind. Accurate information about weather dataset is very crucial and of utmost importance in the current modern world.

For this purpose, it is important to have an application that can let people know about weather alerts and store the weather data in a database to serve the academicians for weather analytics. We propose our work on creating a weather forecast app with Python programming language using OpenWeatherMap API and MongoDB as a database solution. Our application will also surface the data analyzed along with the generated insights to the any consumer.

Our application will store the forecasted data in MongoDB and will display weather alerts such as snow, rain or freezing temperatures, if present, in forecast for those cities. It also creates a weather map for each of the forecast and shows weather forecast on a map. The application will also provide API to academician to use historical, current, forecasted and analyzed weather data.

**Dataset**

We will use OpenWeatherMap as a data provider to our application. OpenWeatherMap is an online service that provides current, historical, and weather forecast data for analytics. OpenWeatherMap provides a multitude of API to fetch current, historical, and forecasted weather data at hour or day level. Some of the APIs provided by OpenWeatherMap are below:

Graphical user interface, text, application, email, website

Description automatically generated

Graphical user interface, text, application

Description automatically generated

We plan to use current weather data API and Climatic forecast 30 days API to retrieve the data needed for this business proposal. To communicate with the weather data, a user must subscribe to the OpenWeatherMap website at <https://openweathermap.org/> and then a user can get API access key. Once user acquires the API access key, the weather data can be downloaded simply by requesting data from server API endpoint. The server will provide the data in json format, and we will parse it to store it into our mongo db database solution.

**Data Collection**

1. Created an account in <https://openweathermap.org/api>

Graphical user interface, text

Description automatically generated

1. Got API access key by subscribing to OpenWeatherMap website.

A screenshot of a computer

Description automatically generated with medium confidence

1. Made a multi-threaded program to connect to API. Locations to be monitored was placed in configuration file.

Text

Description automatically generated

1. One thread would download 5 days/3-hour forecast.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. One thread downloads weather map:

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. All data was stored in database (MongoDB) as separate collections/table.
2. One thread would open the latest weather map and display the map in window (should show last image as per last time stamp)
3. Forecasted threads should print out alerts if there is rain/snow or freezing temperatures (<2-degree Fahrenheit) in any of forecast period.
4. Displayed forecast/previous data from database as a graph.

**Database Design Diagram:**

**Diagram

Description automatically generated**

The database that we have designed for weather application can be logically divided into three sections for the better understanding.

1. **User Preference**: The first one is User Preference section where we will be storing all the users’ accounts and user’s preference about the unit of measurement. For instance, a user may like the temperature to be in Celsius whereas another user may prefer it in Fahrenheit.
2. **User Profile:**  The section is user’s profile section where we store the city’s reference data and user’s profile data like what are the cities that a user is interested to get weather alert for.
3. **Weather Log:** This will be the final logical division of our database design where we will be storing the weather data in the different tables. We will be storing hourly weather data for every city in weather\_hourly\_forecast table and daily weather data in weather\_daily\_forecast table. Different attributes in these table will be city, weather status – Sunny/Rainy/Cloudy, temperature, humidity, wind speed, start\_time and end\_time of weather forecast, etc. We will be storing the data in these tables using metrics units and will be converted to unit preferred by user while delivering the data through our application or application API.

**Pros and Cons of Using RDBMS v/s NoSQL**

Before considering the pros and cons of SQL v/s NoSQL for this use case, we would like to look into our use case and dataset specifically. We want to see the main features and attributes of our dataset and use case which can help us evaluate the pros and cons of using SQL v/s NoSQL.

The use case of our project is to provide our application consumers with forecasted weather data for day-to-day activities or to provide weather data feed to researchers and academicians for predictive analysis. And the structure of the weather dataset to be used in our application is fixed. It contains various weather-related parameters like temperature, humidity, pressure, visibility, etc. For a weather dataset, the structure of these parameters remains fixed, and we do not anticipate changes in the schema and structure for this use case. An additional property of our application is that it’s read-heavy than write-heavy – meaning we will be getting way more read requests than write requests.

Now considering the above attributes, we feel that using RDBMS will provide us benefits in terms of the simplicity of the design. It will allow us to keep the data normalized and allow us to join the data and serve our users' needs. But considering the hourly volume of data for every city for many years, we feel that our dataset will become quite huge and may degrade performance.

Using a NoSQL database will allow us to scale horizontally and serve the requests with high performance. We anticipate that we will be getting high read requests for current data than historical data. We would use different collections for current and historical data in Mongo DB with data sharding in each collection. The highly demanded requests for the latest weather data get served from the smaller collection with high performance. As we don’t anticipate many writes into our system, we won’t be having the eventual consistency that NoSQL DB provides.

**The Cost of Data Collection:**

To get weather data for any location on the globe immediately with OpenWeatherMap superb APIs, we just need to sign up with email and start using minute forecasts, hourly forecasts, history, and other weather data in your applications for free. For more functionality, they have some generous subscriptions.

Graphical user interface

Description automatically generated with medium confidence

For simple projects, we can get the required information for free and can upgrade to any subscription at any time depending on the requirements of the individuals. We can also get the historic weather information for analysis at the below shown prices.

Graphical user interface

Description automatically generated

The main advantage with OpenWeatherMap is it provides free data for students and educators for both current weather information and medium plan for historic weather information.

Graphical user interface, text, application, email

Description automatically generated

**Cloud Storage**

We are planning to store our data in MongoDB Atlas database as this is relatively lesser in price compared to AWS RDS. The core of MongoDB Cloud is MongoDB Atlas, a fully managed cloud database for modern applications. Atlas is the best way to run MongoDB, the leading modern database. MongoDB’s document model is the fastest way to innovate, bringing flexibility and ease of use to the database.

MongoDB Atlas is the only globally distributed, multi-cloud database that enable applications to make use of two or more clouds at the same time. Atlas is available on 70+ regions across AWS, GCP, and Azure. Best-in-class automation and proven practices guarantee availability, scalability, and compliance with the most demanding data security and privacy standards. In the MongoDB Atlas, we have the option of choosing from 3 different services which is flexible. Here is the screenshot showing the price details of their cloud services.

Graphical user interface, application

Description automatically generated

MongoDB Atlas is offering upto 5GB of storage for free for the small applications or for the teams learning MongoDB which is very advantageous. Also, we can clearly see in the above screenshot that the Dedicated plan is recommended because it has the medium pricing with more benefits like the storage upto 4TB which is pretty good for the advanced applications.

**Timeline for the project:**

* Data Collection (Created an account with <https://openweathermap.org/api>)
* Access to API key by subscribing to OpenWeatherMap website
* Multi-threaded program to connect to API and locations to be monitored was placed in configuration file
* One thread would download 5 days/3-hour forecast
* One thread downloads weather map
* Data was stored in MongoDB
* One thread would open the latest weather map
* Alerts will be printed if there is any snow/rain
* Forecast/previous data will be shown in graph