Rule Engine

Github Repo - https://github.com/paritosh31mishra/Rule_Engine

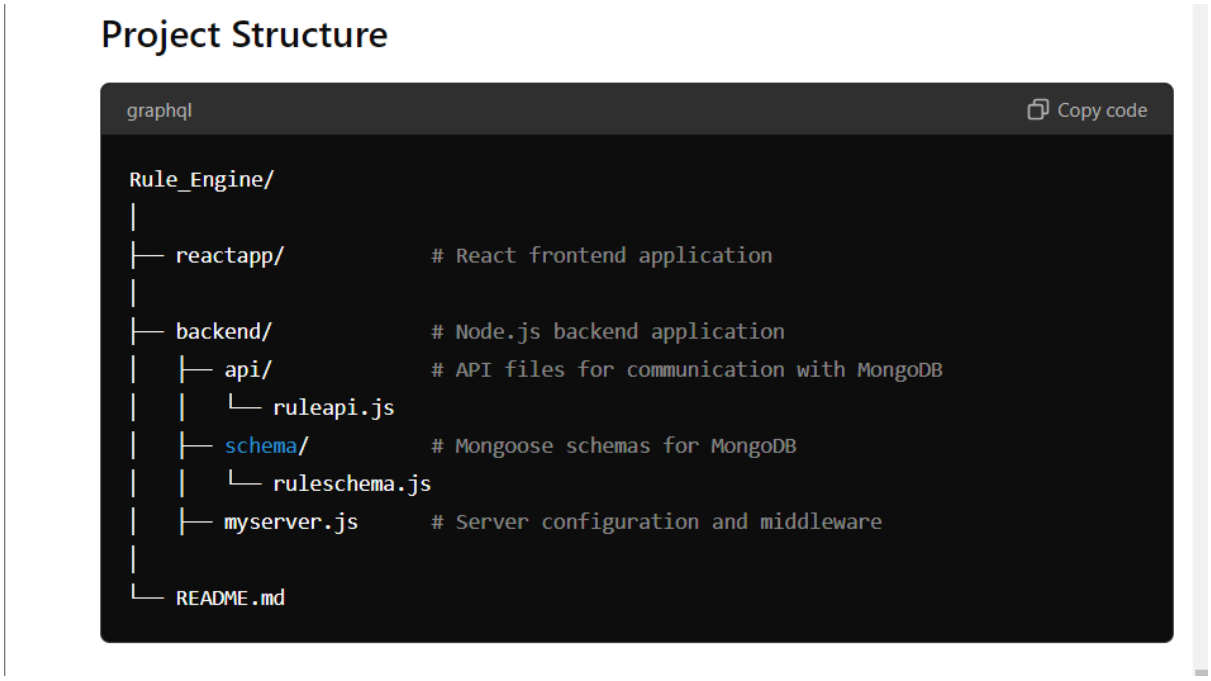
Overview

This application is built with the MERN stack (MongoDB, Express.js, React.js, Node.js) and provides a rule engine that allows users to create, combine, and evaluate rules. It stores rule-related data in a MongoDB database and provides real-time evaluations based on user-defined rules.

Technologies

MERN stack (MongoDB, Express.js, React.js, Node.js)

Project Structure



Prerequisites

Ensure you have the following installed:

- **Node.js** (v14.x or later)
- **MongoDB** (Community Edition)
- **npm** (Comes with Node.js)

- **Git** (For cloning the repository)

Dependencies

Backend

- **express**
- **mongoose**
- **cors**

Frontend

- **react**
- **react-dom**
- **react-scripts**

Getting Started - Run command in vscode terminal or bash

Clone the Repository

git clone https://github.com/paritosh31mishra/Rule_Engine.git

cd Rule_Engine

Setting Up the Backend

1. Navigate to the Backend Directory

cd backend

2. Install Dependencies

Install the required npm packages:

npm install

3. Install Mongodb

npm install mongodb

4. Run the Backend Server

nodemon myserver.js / node myserver.js

5. Install Mongodb compass (community version) from internet

And create database Rule_Engine

Setting Up the Frontend

1. Navigate to the React Application Directory

cd ../reactapp

2. Run the React Application

Start the React application:

npm start

The frontend will be available at http://localhost:3000.

Access the Application:

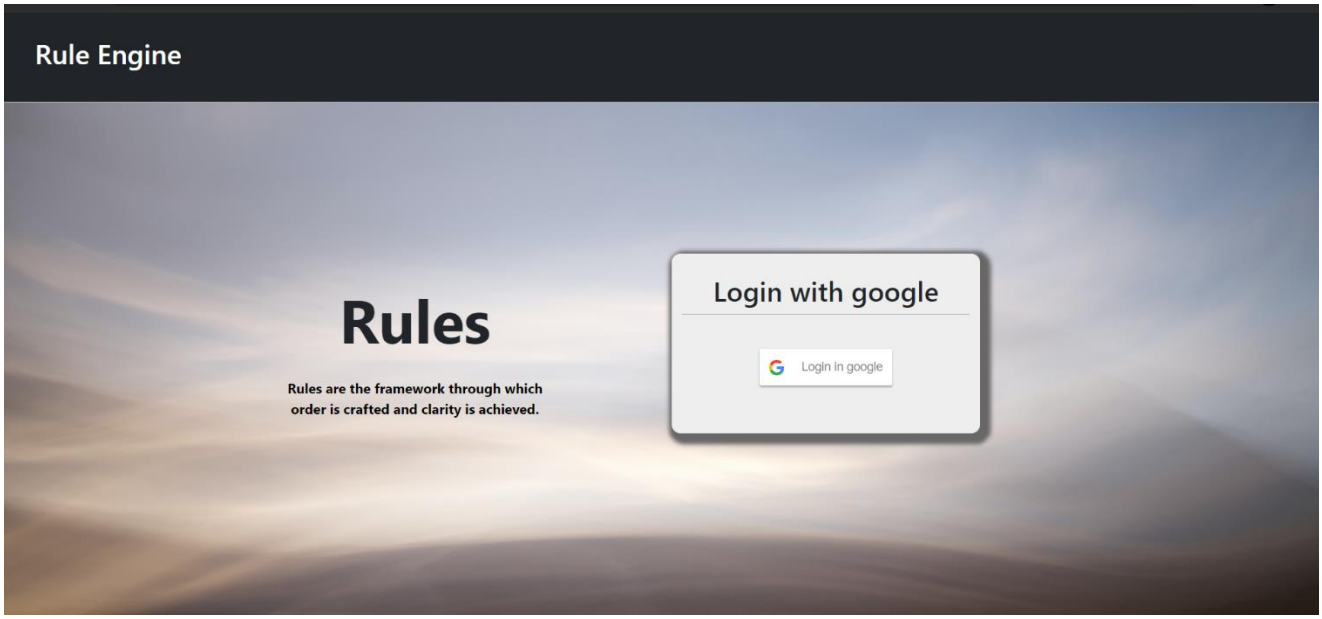
Frontend: http://localhost:3000

Backend API: http://localhost:5556/rule

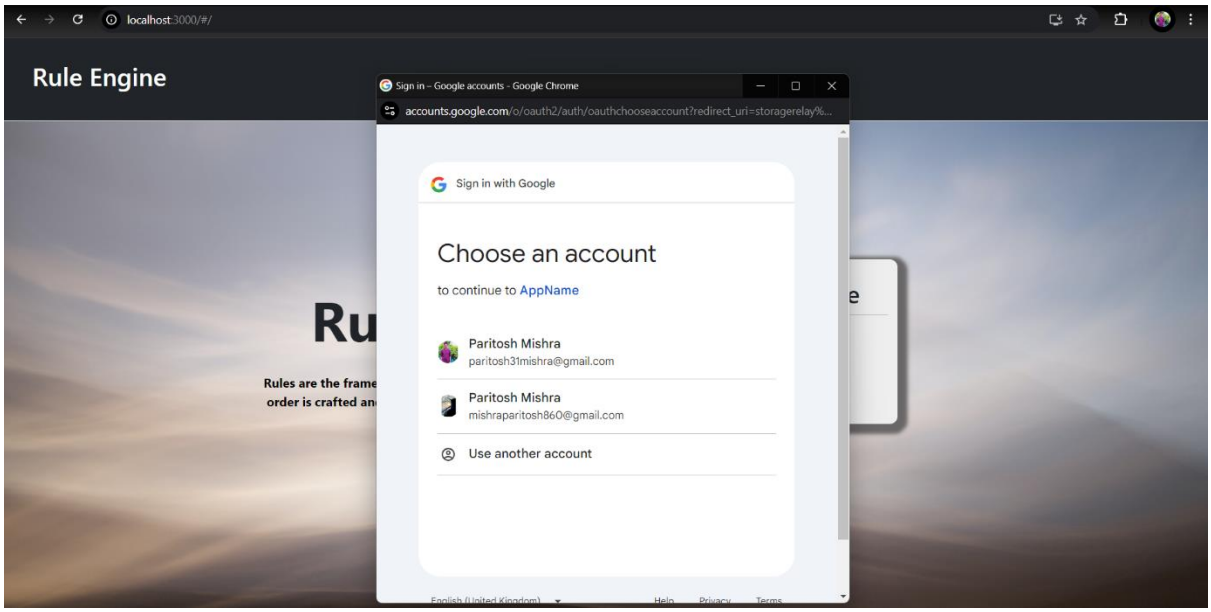
Database: open mongoose portal and see the data

Output of Rule_Engine –

1. **At** http://localhost:3000 (beginning)

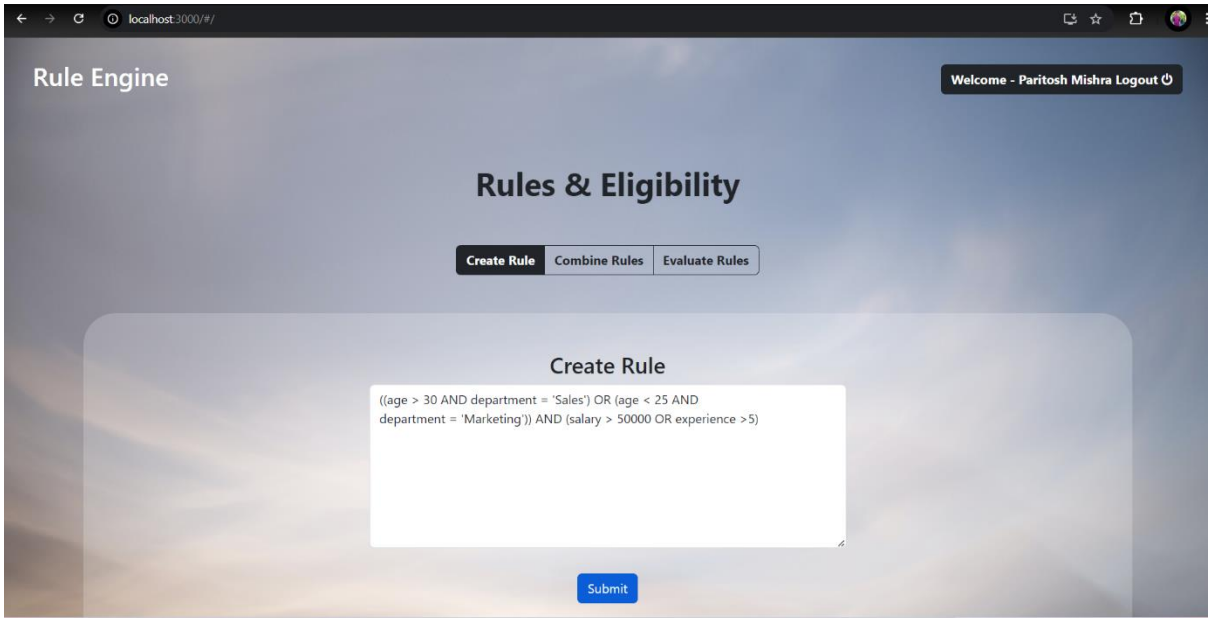


2. **Google Sign in**

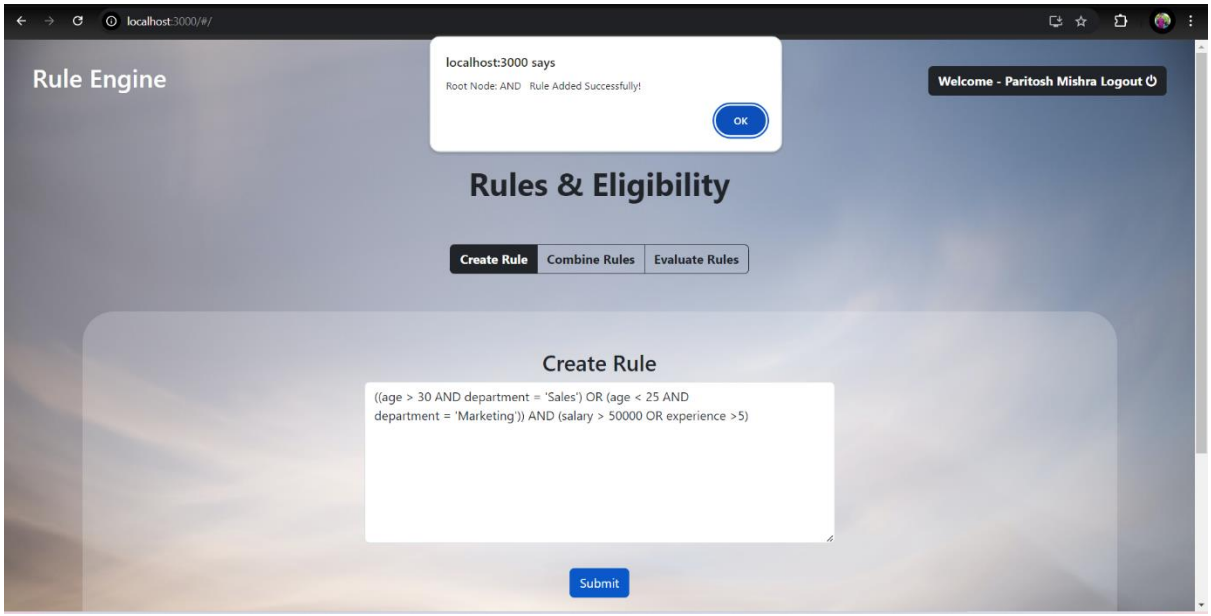


3. Main Page –

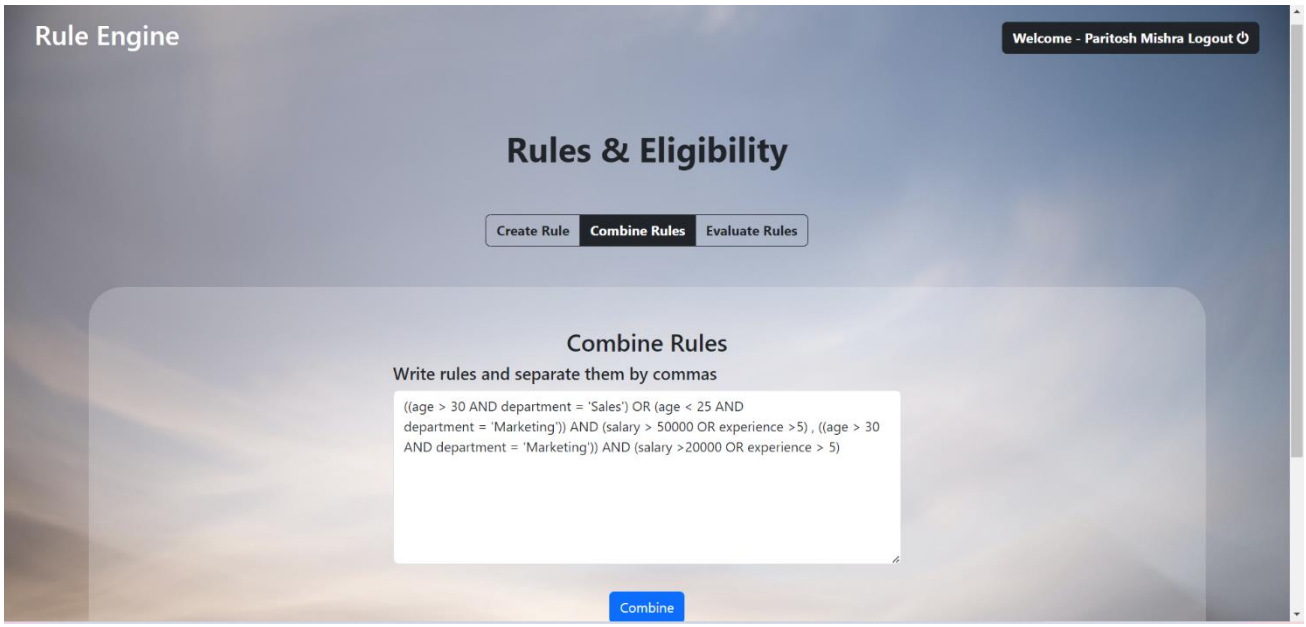
- **Create Rule -**



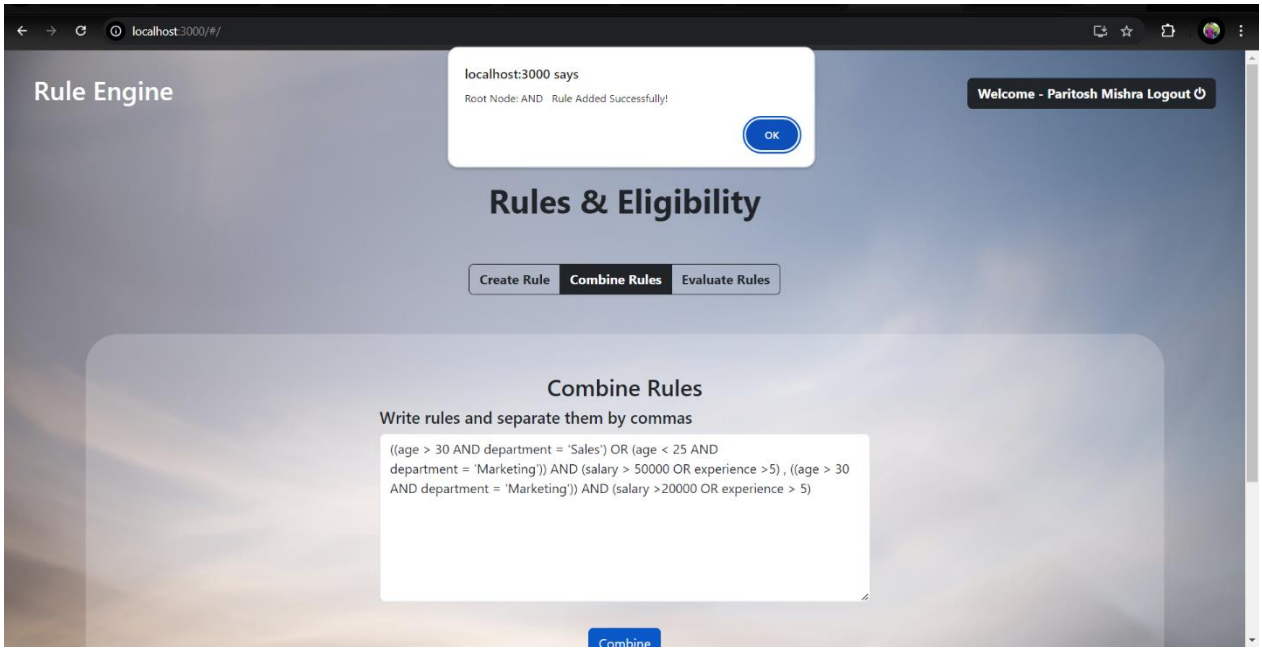
Create Rule output –



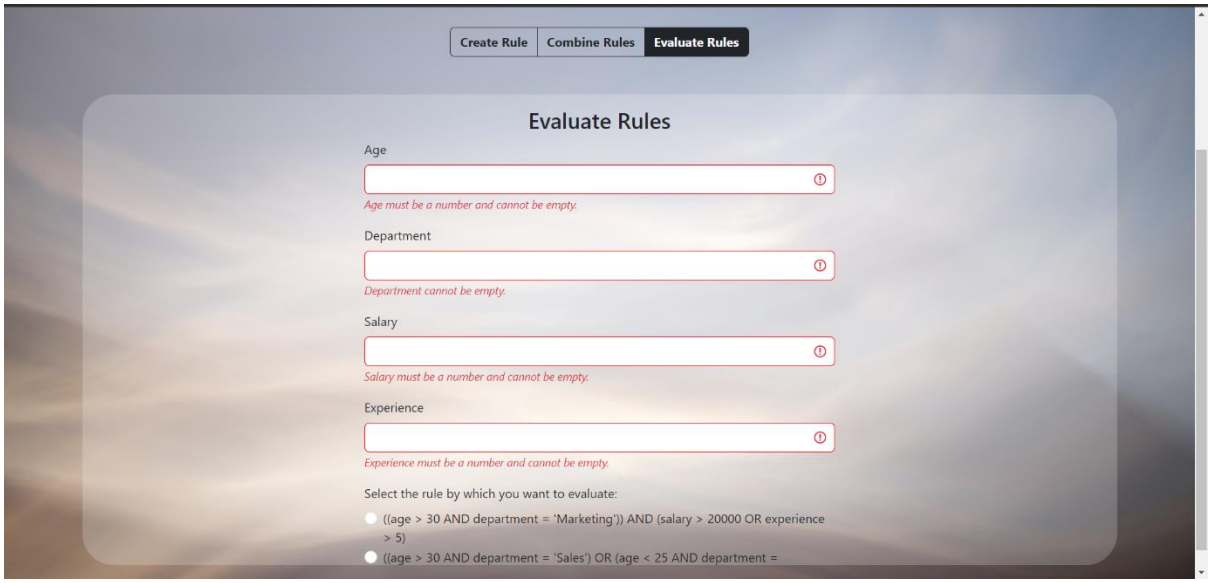
- **Combine Rules-**



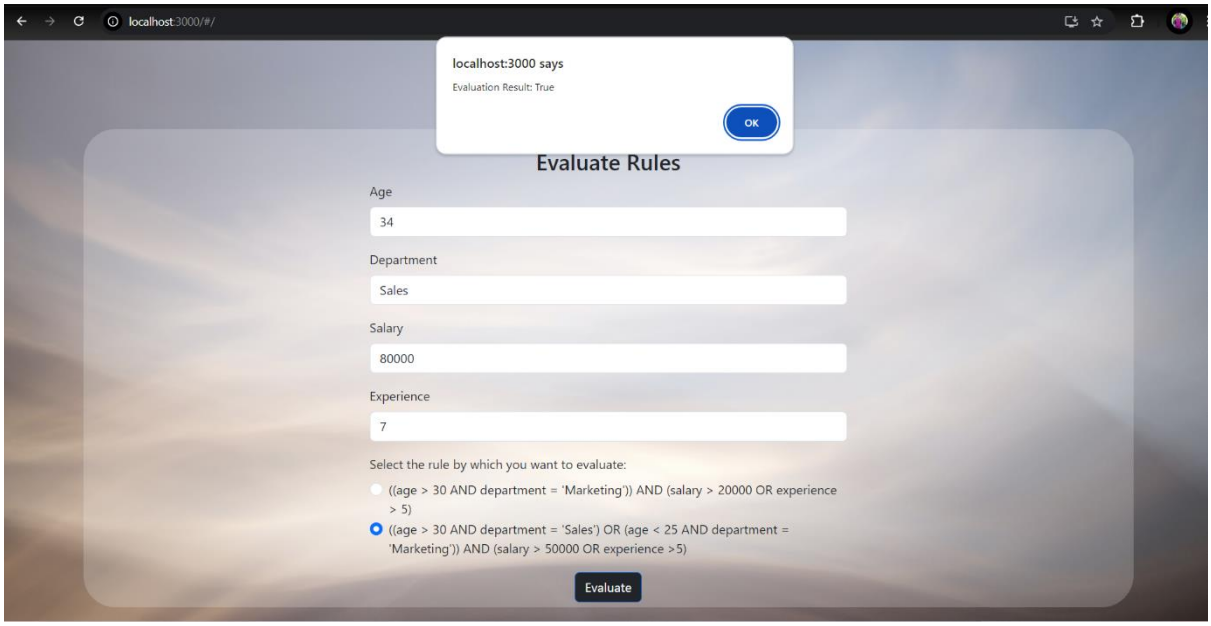
Combine Rule Output –



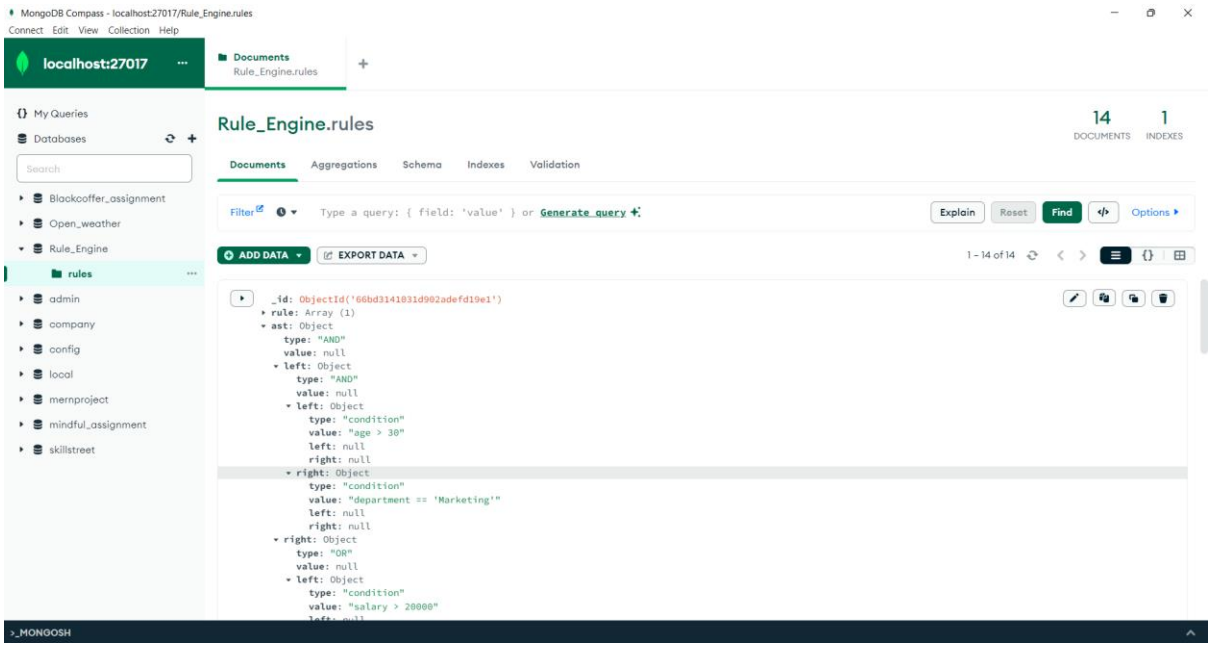
Rules Evalution with Validation -



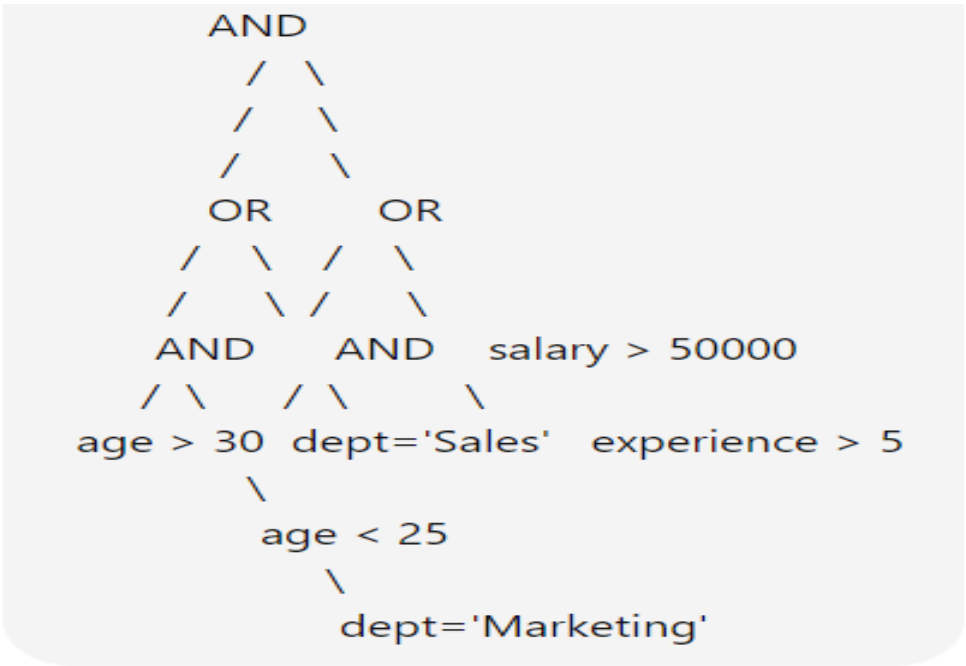
Output of Evalution –



4. AST tree stored in Database –



Ast structure –



Operator as parent node(root node) and condition as Leaf Node

Ast In Object format –

```
const ast = {
  type: 'AND',
  value: null,
  left: {
    type: 'OR',
    value: null,
    left: {
      type: 'AND',
      value: null,
      left: { type: 'condition', value: 'age > 30' },
      right: { type: 'condition', value: 'department == \'Sales\'' }
    },
    right: {
      type: 'AND',
      value: null,
      left: { type: 'condition', value: 'age < 25' },
      right: { type: 'condition', value: 'department == \'Marketing\'' }
    }
```

Open Weather

Github Repo - https://github.com/paritosh31mishra/Open_Weather

Overview

Open Weather is a web application that provides real-time weather data. The project consists of a React frontend and a Node.js backend with MongoDB for data storage. The application allows users to view weather information for various cities.

Technologies

MERN stack (MongoDB, Express.js, React.js, Node.js)

Project Structure

Project Structure

graphql Copy code

```
weather/
│
├── reactapp/           # React frontend application
│
├── backend/            # Node.js backend application
│   ├── api/            # API files for communication with MongoDB
│   │   └── weatherapi.js
│   ├── schema/         # Mongoose schemas for MongoDB
│   │   └── weatherSchema.js
│   ├── myserver.js     # Server configuration and middleware
│
└── README.md
```

Prerequisites

Ensure you have the following installed:

- **Node.js** (v14.x or later)
- **MongoDB** (Community Edition)
- **npm** (Comes with Node.js)
- **Git** (For cloning the repository)

Dependencies

Backend

- **express**
- **mongoose**
- **cors**

Frontend

- **react**
- **react-dom**
- **react-scripts**

Getting Started -

Clone the Repository

git clone https://github.com/paritosh31mishra/Open_Weather.git

cd Open_Weather

Setting Up the Backend

1. Navigate to the Backend Directory

cd backend

2. Install Dependencies

Install the required npm packages:

npm install

3. Install Mongodb

npm install mongodb

4. Run the Backend Server

nodemon myserver.js / node myserver.js

5. Install Mongodb compass (community version) from internet

And create database Open_weather

Setting Up the Frontend

1. Navigate to the React Application Directory

cd ../reactapp

2. Run the React Application

Start the React application:

npm start

The frontend will be available at http://localhost:3000.

Access the Application:

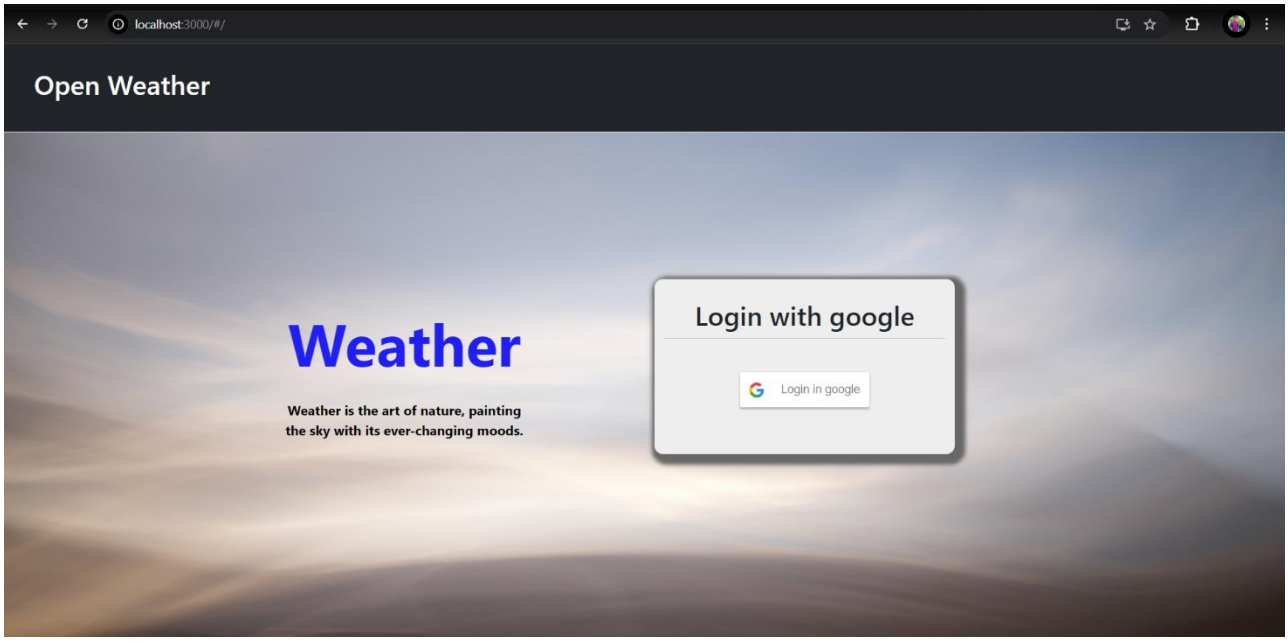
Frontend: http://localhost:3000

Backend API: http://localhost:5557/weather

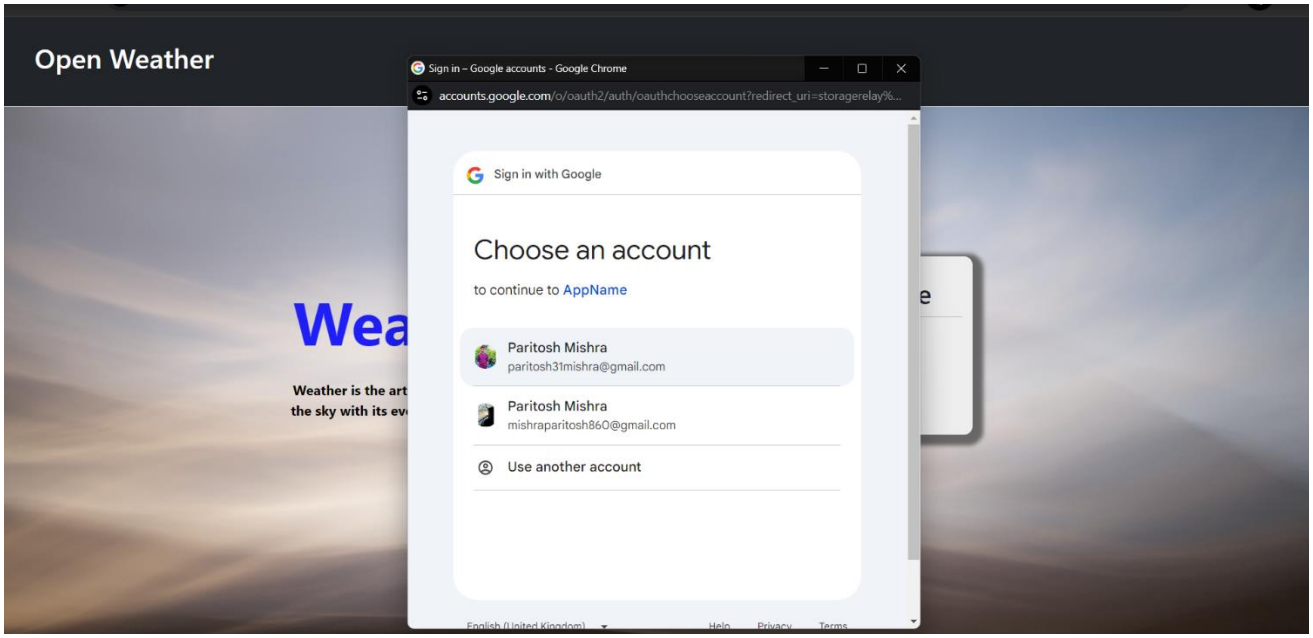
Database: open mongoose portal and see the data

Output of Weather Application –

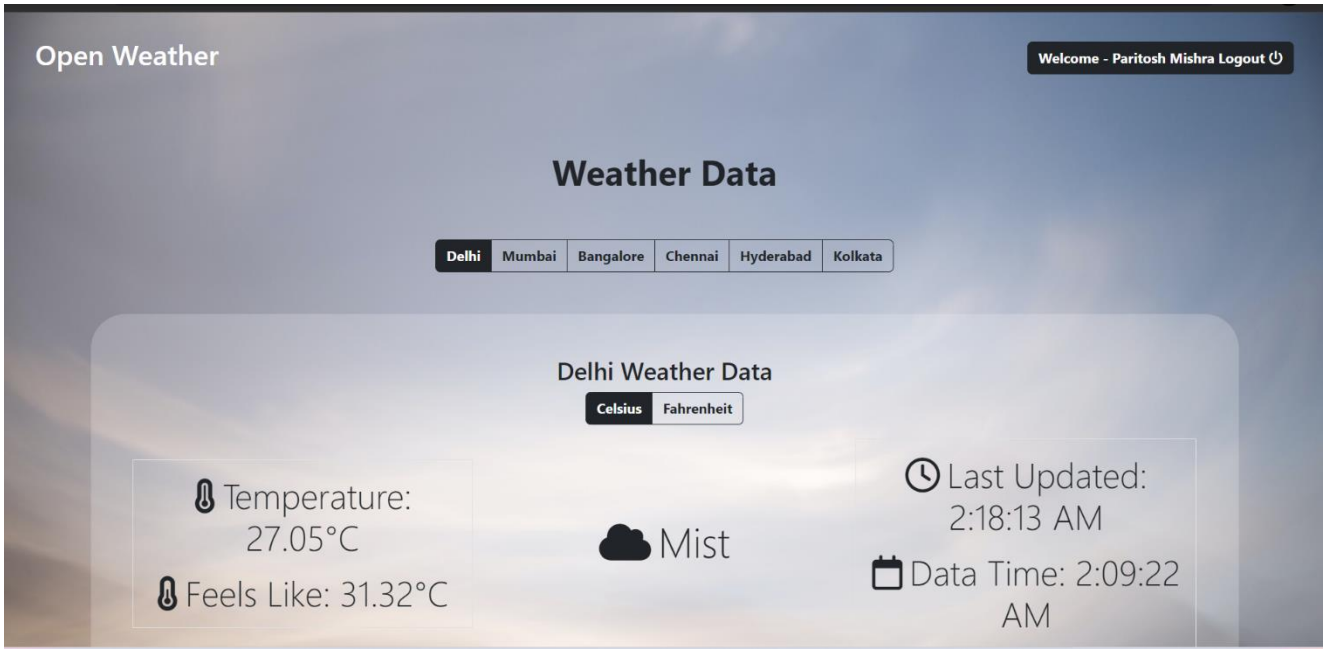
1. **At** http://localhost:3000 (beginning)



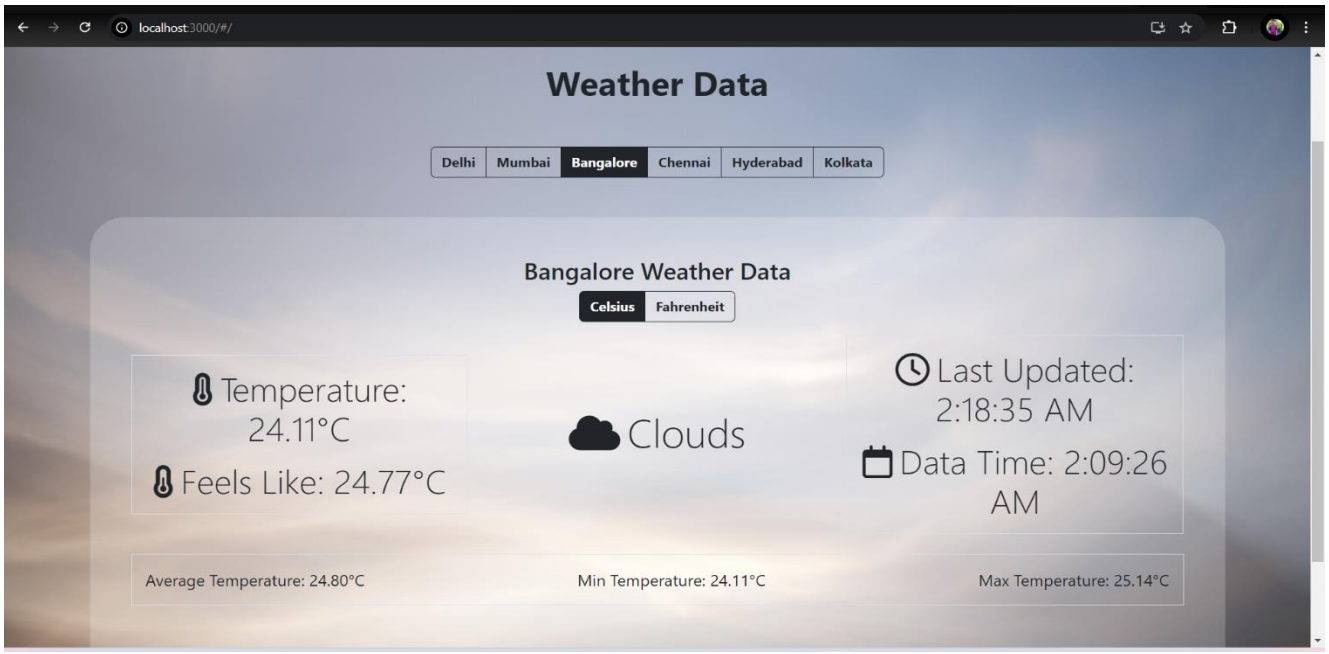
2. **Google Sign in**



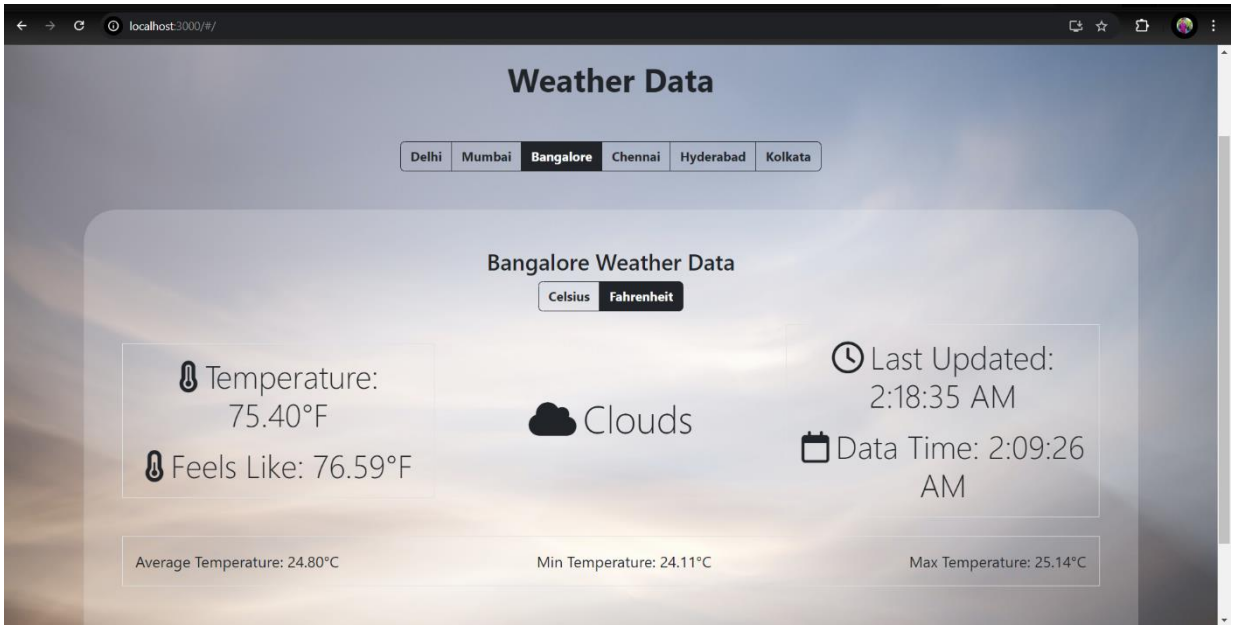
3. Main Page -



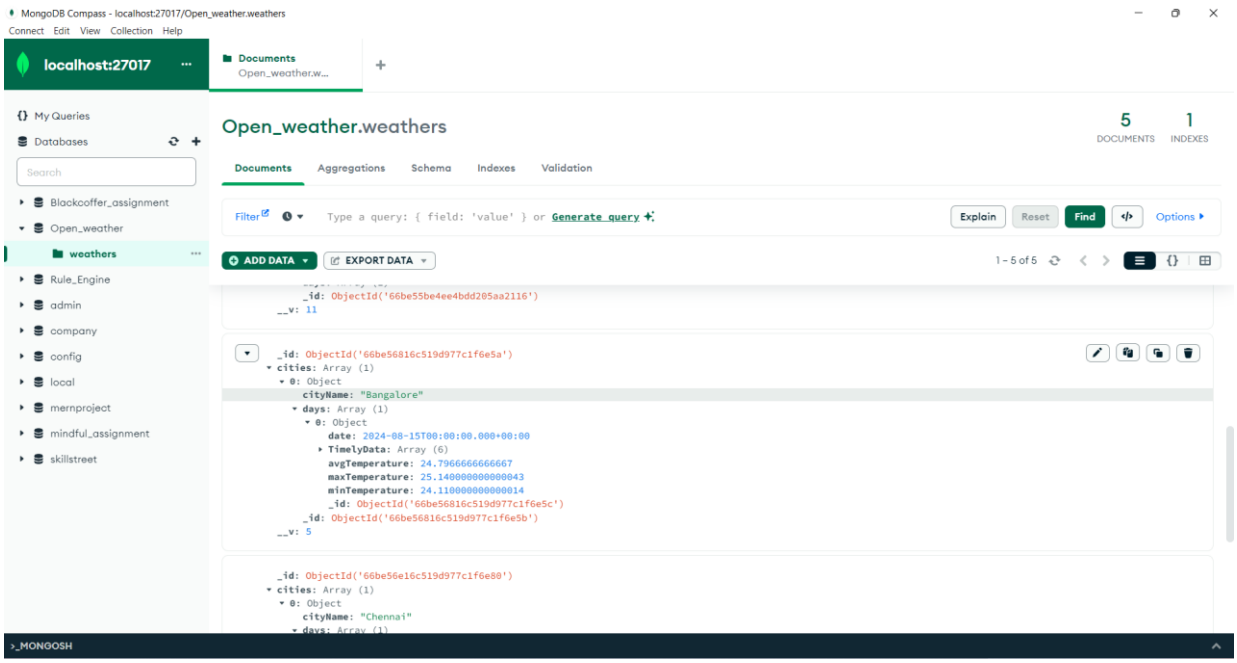
Tempreature in Celcious -



Tempreature in Fahrenheit -



4. Data in Database –



5. Notified when Tempreature cross the threshold value –