**Winter Supplement Calculator Documentation**

**Overview**

The **Winter Supplement Calculator** is a web-based tool designed to determine eligibility for winter supplement benefits based on user inputs. The application utilizes MQTT (Message Queuing Telemetry Transport) for communication between the frontend and backend, ensuring real-time interactions.

**Features**

* **Frontend**: A clean web-based interface built using Bootstrap for responsive design.
* **Backend**: The backend is powered by Python, which handles the eligibility calculations via an MQTT engine.
* **Real-time Communication**: The frontend and backend communicate via MQTT, ensuring real-time updates and results.
* **Scalable**: The application is designed to be easily extendable with additional calculation logic or other functionalities.

**Setup Instructions**

To set up the **Winter Supplement Calculator** on your local machine, follow these steps:

**Prerequisites**

Before you begin, ensure the following are in place:

* Python 3.x installed on your machine.
* A modern web browser (Chrome, Firefox, etc.) to load the frontend.
* Access to a public MQTT broker (e.g., **test.mosquitto.org**) for MQTT communication.

**1. Clone the Repository**

Clone the repository from GitHub and navigate to the project folder:

git clone https://github.com/SnehaPopuri/winter-supplement-calculator.git

cd winter-supplement-calculator

**2. Install Backend Dependencies**

Install the required Python dependencies by running the following command:

pip install -r requirements.txt

**3. Run the Frontend**

You can either open the index.html file directly in your browser or serve it locally:

* Open index.html in your browser.
* Alternatively, run a simple local server with Python:
  + python -m http.server 8000
* Access it at http://localhost:8000 in your browser.
* To log in, use the following credentials:
  + **Username**: user
  + **Password**: password

Run the backend script:

python main.py

**4. Update the MQTT Topic ID (Local Setup)**

If you wish to run the application locally and change the MQTT topic ID, follow these steps:

* **Backend (main.py)**:
  + Open main.py and locate the constants for input and output topics. Update these constants with your desired topic IDs:

TOPIC\_INPUT = "your\_custom\_topic/input/"

TOPIC\_OUTPUT = "your\_custom\_topic/output/"

* **Frontend (index.html)**:
  + Open index.html and find the section where the topic ID is generated. You can either replace crypto.randomUUID() with a static ID or leave it as is to automatically generate a new ID on each page load:

const uniqueId = crypto.randomUUID(); // Or set a custom ID

document.getElementById('mqttTopicId').value = uniqueId;

**5. MQTT Cloud Setup (Optional)**

You can view the results of the MQTT communication in the cloud by setting up a test client:

* Go to the MQTT client website: https://testclient-cloud.mqtt.cool
* Choose the broker: tcp://test.mosquitto.org:1883
* Enter the following **Subscriber IDs**:
* **Input**: BRE/calculateWinterSupplementInput/14a8301b-5999-47cf-914e-a604d82e218b
* **Output**: BRE/calculateWinterSupplementOutput/14a8301b-5999-47cf-914e-604d82e218b

The published output will be displayed on the frontend website.

**Troubleshooting**

If you encounter any issues, here are some steps to resolve common problems:

**MQTT Connection Issues:**

* Ensure that the broker (test.mosquitto.org) is reachable.
* Verify that the topic IDs match between the frontend and backend.
* If using a custom broker, ensure the correct server address and port are configured.

**Frontend Not Displaying Results:**

* Open the browser's console (press F12 or right-click > Inspect > Console) to check for any errors.
* Ensure that the backend is running and correctly connected to the MQTT broker.

**Backend Errors:**

* Confirm that the rules\_engine is properly implemented for eligibility calculations.

**Contributing**

We welcome contributions! If you'd like to propose changes, please follow these steps:

1. Fork the repository.
2. Create a new branch for your changes.
3. Make your changes and test them.
4. Submit a pull request with a description of your changes.