**Assignment details :**

1) Scrap web data / Create a dummy data of a daily weather of multiple cities. (data should be for every second)(if dummy, temp range 30 -50)

2) Publish the generated / scraped data to rabbitMQ (every second --> multiple cities data will be sent)

3) Consume the published data in a python process

4) Use multi-thread and create a thread for each city

5) send mail to user whenever temp crosses 45 degrees ( once mail is sent, no mail should be sent for another hour, and repeat)

P.S : if scraping data , scarp for every 5 minutes and send same data for every second.

**Solution Design :**

DB with the list of Cities

The python function creates the multi threads , a thread for each city.

Consumer picks the temperature data every second and send it to the defined processor.

Producer fetches the city names for every 5mins from the DB and assign the random temperature to each city and send it to the message broker every second.

**Project Package :**

**Thread for city 1** : Verify the conditions and trigger the alert if require.

**Thread for city 3** : Verify the conditions and trigger the alert if required.

**Thread for city 2** : Verify the conditions and trigger the alert if required.

The project has below 4 python files :

1. sqlitedb\_setup.py : This file will create the required database and the tables for this project. The below tables are maintained for this project.
   1. Table <cities> : Lists the cities that need to be monitored.
   2. Table <e\_log>: Captures the log of emails sent with time stamps.
2. producer.py : This python file is responsible for creating the temperature data and push it to the message broker every second.
3. consumer.py : This file will trigger as soon as message arrives the dedicated queue and the send the data to the processor.
4. threads.py : this python module creates threads for each city and each of these threads send the email alerts as per the checks and update the log table in the DB.

**Other insights :**

During the development it has been observed that there is chance of potential optimization to the current approach

* Current approach : Creating a thread for each city irrespective of verifying whether it requires further processing or not (Email trigger).
* Suggest approach : Threads can be created only if they require further processing.
* Ex: Threads can be created only to those cities where the temperature is beyond the threshold and there is no recent email log.