COMP9322 Assignment 2

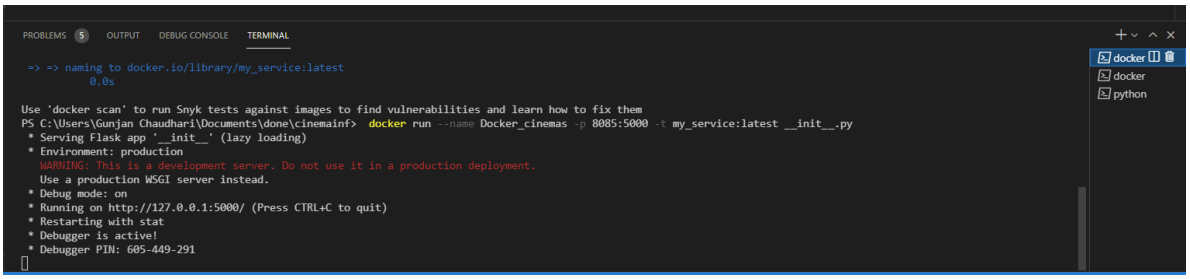
Zid: Z5302729

Steps to run the code,

1. install: conda install flask-CORS

2. in folder Cinemainf, run command : build -t my_service:latest .

3. run command: docker run --name Docker_cinema -p 8085:5000 -t my_service:latest __init__.py



4. in folder timslot, run command : build -t my_service:latest .

5. run command: docker run --name Docker_booker -p 8080:5000 -t my_service:latest __init__.py

6. Go to bot/chatbot/demo and run command : python __init__.py

This will start the environment by running both the dockers, containing booking and cinema information services, which are called in the 'bot' folder connected to the 'wit.ai'

To connect to a chatbot navigate to the folder 'jake' and

Run: python -m http.server

Using a browser, Go to http://localhost:8000

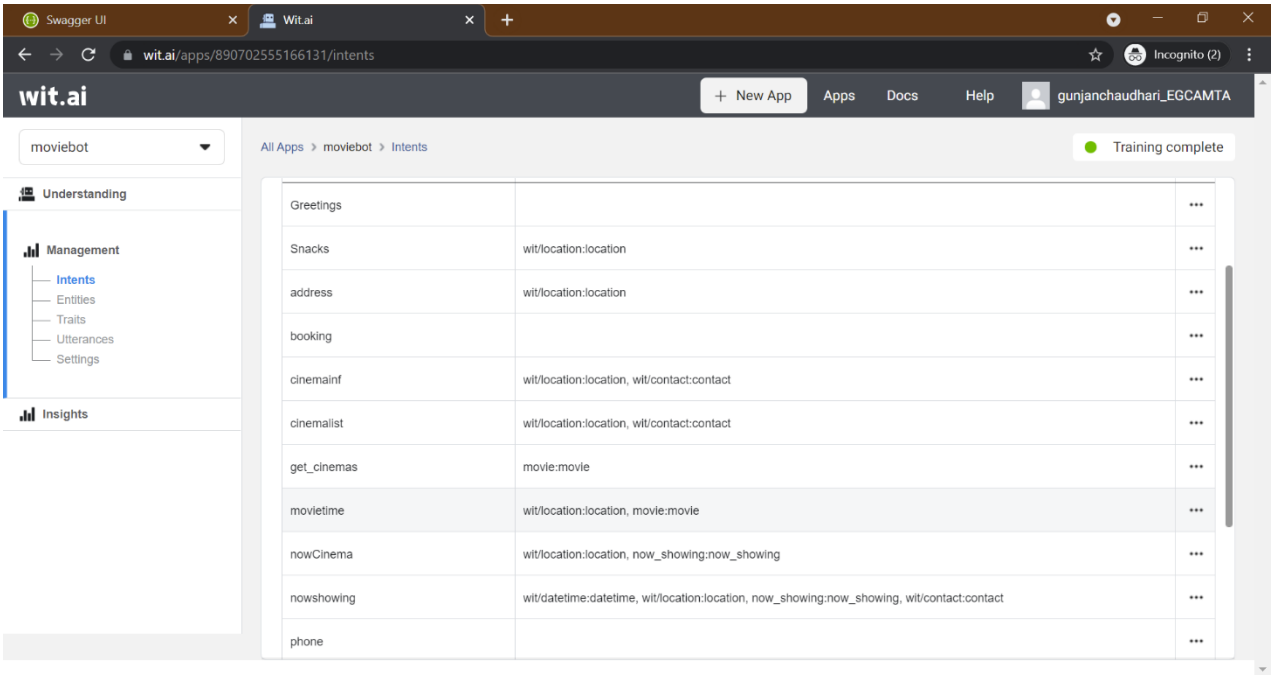
WIT usage:

Facebook Credentials:

Email address: gunjanunsw@gmail.com

Password: Gunjan1234

Intents Page:



Chatbot Working:

Flow:

The chatbot ideally starts by ‘Greetings’ after which it askes the user to enter their name. This name is stored in a sample.txt file for remembering the username.

The user can then ask multiple questions related to cinema, movies, and the timeslots. The respective intents are created on WIT. Depending on which intent is called the chatbot searches for the respective ‘if ‘ statement.

e.g., questions,

Which can I watch a movie?

What is address of Ritz?

When can I watch No Time To Die at HOYTS Broadway?

I want to book tickets for Halloween Kills

Conversation:

The sample.txt stores conversation dictionary which contains initially:

conversations = { "name": 0, "tickets": 0,"timeslot": 0,"ttype": 0,"booking_id": 0,"intent": 0}

As the conversation goes ahead each of the fields is filled when everything is filled the chatbot asks user if the want to confirm the booking. If the user says 'Yes' the booking is registered in the database.

CinemaInf:

The cinemaInf contains code, which returns cinema and movie related queries, to the swagger.

Timeslot:

The timeslot contains code, which returns timeslot and booking related queries and insertion.

Bot:

The bot contains the main conversational logic of the chatbot which is imports data from the 2 dockers

Jake:

Jake contains the frontend chatbot files.

Documents:

Contains yaml files, and the installation instruction