**Hotel Recommendation AI with Langchain**

**TABLE OF CONTENTS**

1. Objectives
2. Design
3. Implementation
4. Challenges
5. Lessons learned

**Objectives**

The main objective of this project is to provide guidance to the user by recommending best hotels around Bangalore. This is built using the langchain framework.

The chatbot gets the location from the user with Bangalore region and suggests best hotel based on their preference, location and budget.

The chatbot recommends 3 hotels to the user and answers any follow-up questions related to the same when user has concerns.

If the user is searching for hotels other than Bangalore region, proper responses will be shown to the user.

**Design**

**Retrieve/build the Dataset**

1. Downloaded the bangalore hotels dataset from Kaggle.
2. The dataset had around 68000 records.
3. Filtered and retrieved only few dataset for different locations [around 200]
4. Added the description field based on few columns in the dataset. The descriptions are generated by GPT 4
5. The same fields that are used to generate the description were taken into account to collect the user preferences.
6. A simple dataset has been kept ready for the chatbot to provide recommendation to the customer.

The langchain APIs were used to retrieve the relevant output from the chat model.

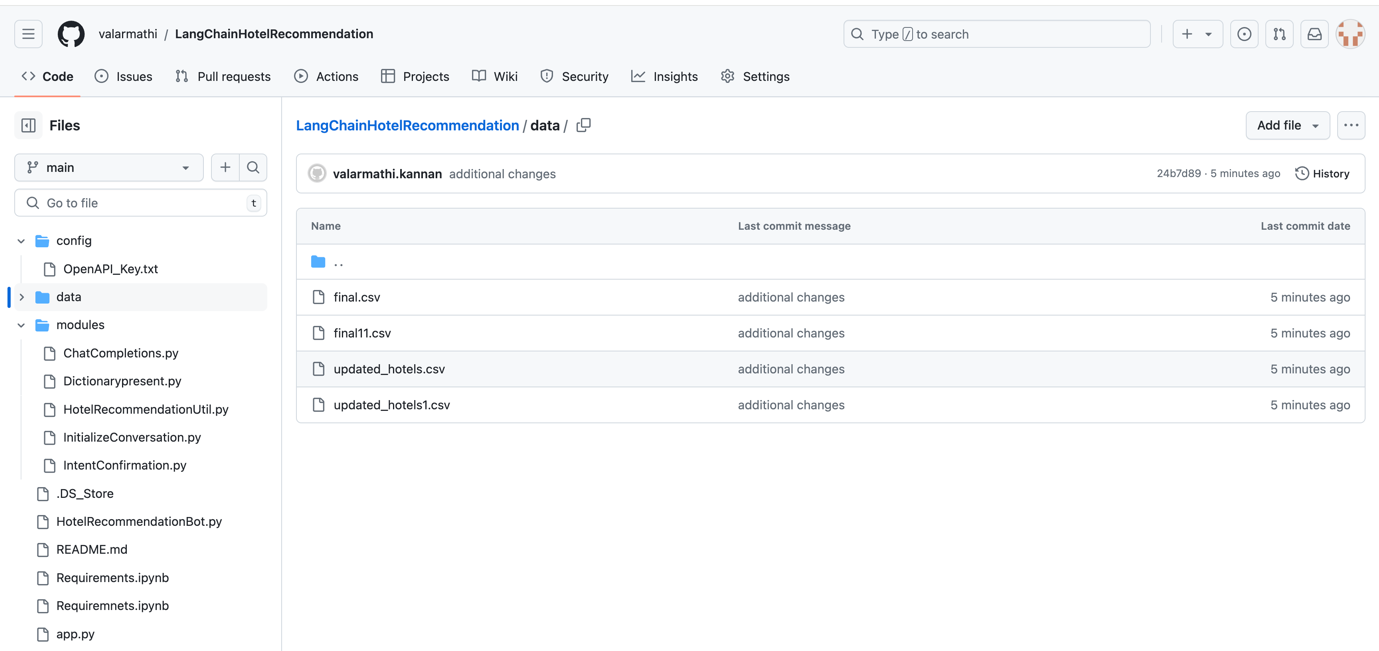
**Implementation**

The code repository for this bot can be seen here

<https://github.com/valarmathi/LangChainHotelRecommendation>

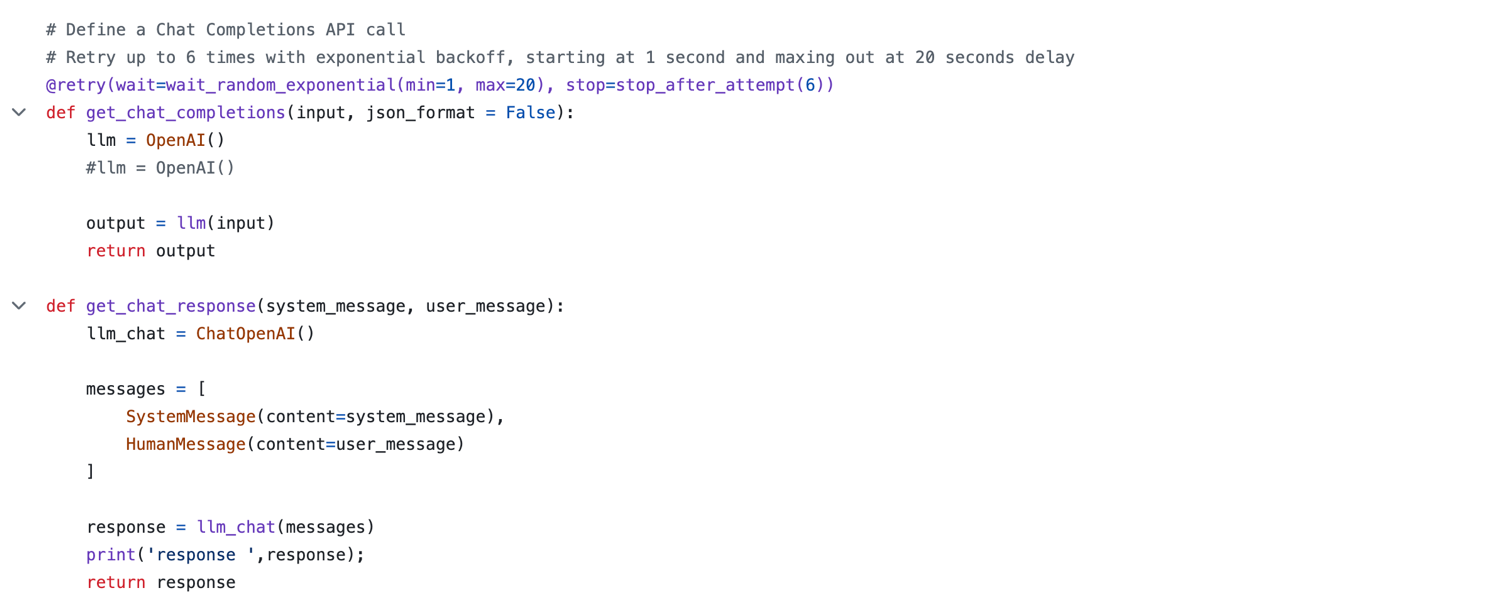
The project is modularized and the needed utilities are placed in the appropriate directory

Here is the code structure



All the prerequisites had been placed in Requirements.ipynb and on the 1st deployment one can run the cells to install the needed packages.

Had 2 basic methods for llm with langchain and this did most of the job of retrieving the relevant response to the user.



**Challenges**

1. Initially struggled to get a exact dataset and filtering the dataset was also a problem.
2. Since the Zomato.csv had around 65000 records, it took very long time to load.
3. Finding the properties to build the description for the hotel dataset took some time.

There were many properties to consider like online\_booking, votes, rating etc.., Not sure which ones to consider. So, I picked up some random ones which are meaningful to me.

1. Most of the langchain APIs used in the live discussions were deprecated and finding an appropriate package and api took time.
2. Felt that interaction made with langchain was very good and we need not write much of the prompts for this.

**Lessons learned**

1. Understood on how the realtime dataset looks like and how huge it is.
2. Writing a better prompt to get the better results.
3. Modularizing the code to avoid confusions.
4. Learnt most of the python features
5. Explored different GenAI models and used various properties of it.
6. Understood how the usage and billing is calculated by OpenAI
7. Integration with langchain