

Implementation choices & challenges

The **Question Answering system** in this project is designed to combine two key approaches:

1. **SQL Agent Model:** This model retrieves relevant data from the database (which is based on the provided dataset) using SQL queries.
2. **RAG (Retrieval-Augmented Generation) Model:** This model has prior knowledge of the dataset's overall structure and insights, which were initially provided in a PDF. It generates responses by combining this general knowledge with the specific data retrieved by the SQL Agent.

This dual approach ensures that the system can **effectively utilize tabular data**, leading to more accurate and mathematically sound responses.

Challenges Faced

The system was initially implemented using the **Llama 2 model**, but a major challenge was its **high computational cost**. Additionally, the system's design is not straightforward, making it complex to handle within limited computing resources. Due to time constraints in completing the assignment, I switched to the **Gemini Flash model**, which offered faster performance while still maintaining good response quality.

I have **kept the code for Llama 2** in the repository, so if you have access to a powerful GPU, you can switch back and use it.

API Documentation

Overview

This document provides details about the API endpoints available in the Hotel Booking Analytics application. The APIs facilitate data analysis and querying functionalities, including analytics generation and interaction with the RAG-enabled SQL querying system.

Base URL

The base URL for the API is:

<http://localhost:8000>

1. Result Analytics API

Endpoint:

GET /result_analytics

`POST /result_analytics`

Description:

This endpoint analyzes the hotel booking dataset and returns various plots as JSON-encoded images.

Request Parameters:

None

Response:

- Returns a JSON object containing plot images in Base64 format.

Example Response:

```
{  
  "Booking_patterns_plot": "data:image/png;base64,iVBORw0KGg...",  
  "Cancellation_rate_plot": "data:image/png;base64,iVBORw0KGg..."  
  ....  
}
```

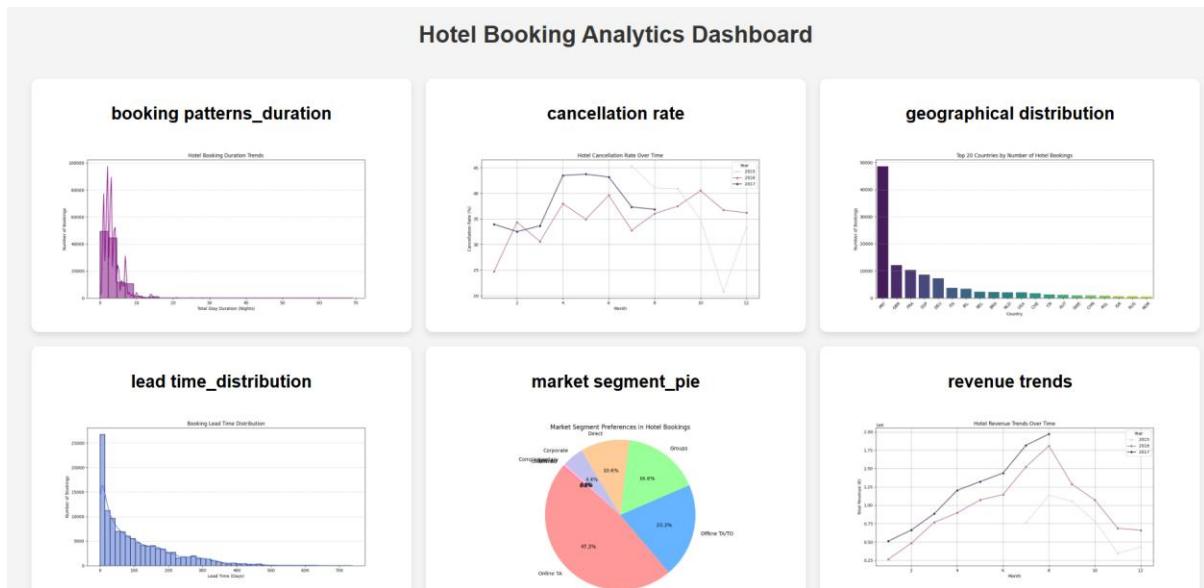


Figure 1: How the charts coming

2. Chat API

Endpoint:

POST /ask

Description:

Processes user queries related to the hotel booking dataset. Uses SQL query execution and a Retrieval-Augmented Generation (RAG) model to generate responses.

Request Parameters:

- **query** (string, required) – The user's question related to the dataset.

Request Example:

```
{  
  "query": "Tell me about the dataset"  
}
```

Response:

- Returns a JSON object containing the query and the AI-generated response.

Example Response:

```
{  
  "query": "Tell me about the dataset",  
  "response": "The Hotel Booking Demand dataset contains detailed information about hotel bookings, including customer demographics, booking details, stay duration, and cancellations, and is widely used for predictive modelling, customer segmentation, and business intelligence in the hospitality industry. The dataset has 119,390 rows and 32 columns with numerical and categorical attributes. It includes information from two types of hotels: City Hotel (Lisbon, Portugal) and Resort Hotel (Algarve, Portugal). The data was extracted from the hotels' Property Management System (PMS) SQL databases covering bookings from July 1, 2015, to August 31, 2017  
}
```

Error Handling:

- If no query is provided, returns:

```
{  
  "error": "Query is required"  
}
```

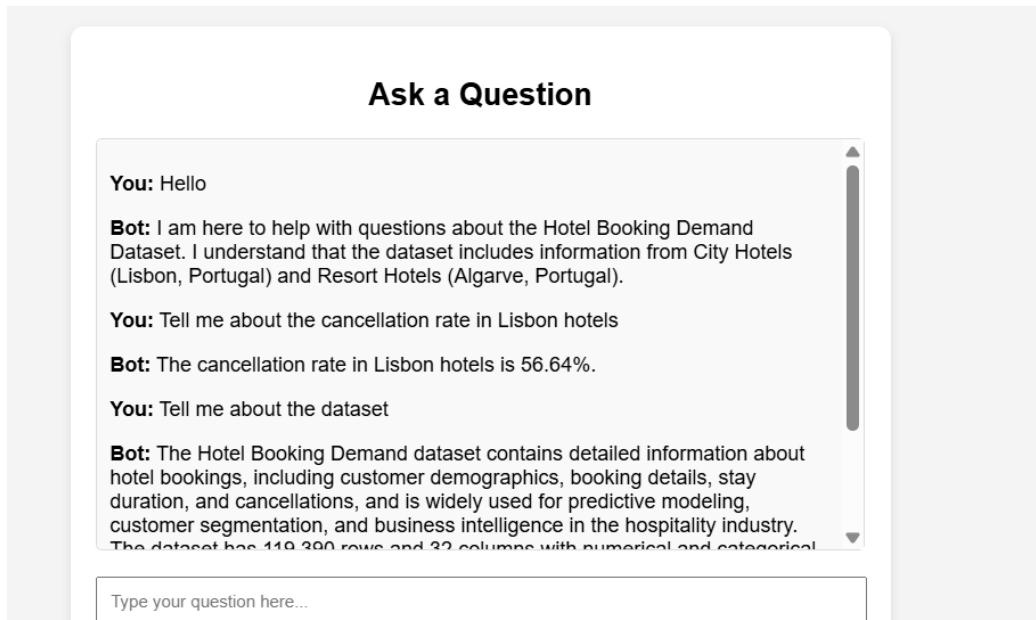


Figure 2: In index.html webpage

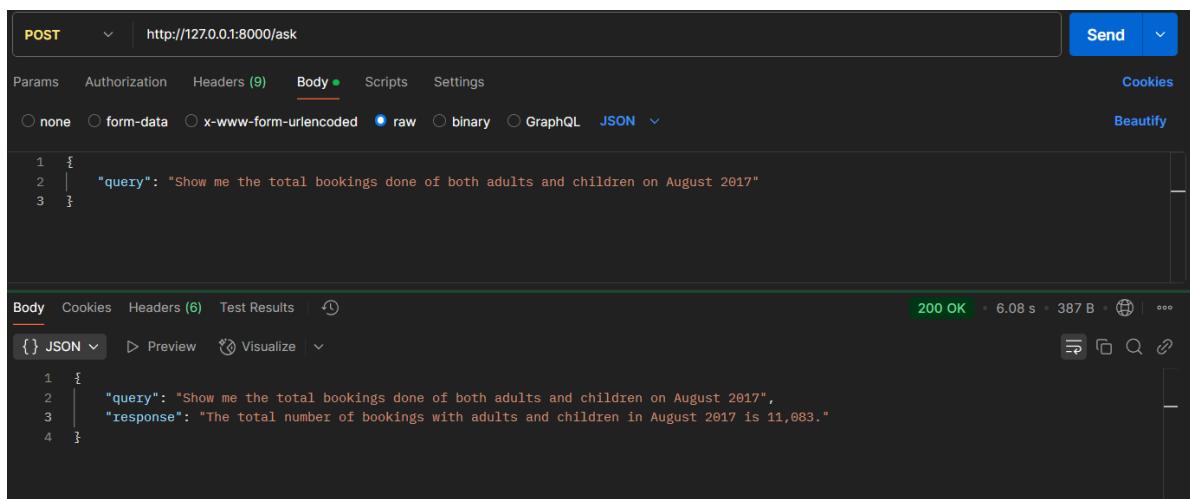


Figure 3: Responses in postman

Notes

- Ensure that the required dependencies and environment variables are set up before using these APIs.
- The **SQL Query Execution** is handled internally, and responses depend on the available dataset and retrieval mechanism.

For further details, refer to the project repository.