


```
import pandas as pd # For handling data
import numpy as np # For numerical operations
import matplotlib.pyplot as plt # For plotting
import seaborn as sns # For visualizations
```

1. Data Collection & Preprocessing

```
df = pd.read_csv(r'/content/hotel_bookings.csv')
```


```
df
```



	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	s
0	Resort Hotel	0	342	2015	July	27		1
1	Resort Hotel	0	737	2015	July	27		1
2	Resort Hotel	0	7	2015	July	27		1
3	Resort Hotel	0	13	2015	July	27		1
4	Resort Hotel	0	14	2015	July	27		1
...
119385	City Hotel	0	23	2017	August	35		30
119386	City Hotel	0	102	2017	August	35		31
119387	City Hotel	0	34	2017	August	35		31
119388	City Hotel	0	109	2017	August	35		31
119389	City Hotel	0	205	2017	August	35		29

119390 rows × 32 columns

```
print(type(df)) # Should print <class 'pandas.DataFrame'>
print(df.head()) # Preview the first few rows
```



```
<class 'pandas.core.frame.DataFrame'>
   hotel is_canceled lead_time arrival_date_year arrival_date_month \
0  Resort Hotel      0      342          2015          July
1  Resort Hotel      0      737          2015          July
2  Resort Hotel      0       7          2015          July
3  Resort Hotel      0      13          2015          July
4  Resort Hotel      0      14          2015          July

   arrival_date_week_number arrival_date_day_of_month \
0                          27                        1
1                          27                        1
2                          27                        1
3                          27                        1
4                          27                        1

   stays_in_weekend_nights stays_in_week_nights adults ... deposit_type \
0                        0                      0    2  ...  No Deposit
1                        0                      0    2  ...  No Deposit
2                        0                      1    1  ...  No Deposit
3                        0                      1    1  ...  No Deposit
4                        0                      2    2  ...  No Deposit

   agent company days_in_waiting_list customer_type adr \
0   NaN      NaN                    0      Transient  0.0
1   NaN      NaN                    0      Transient  0.0
2   NaN      NaN                    0      Transient  75.0
3  304.0     NaN                    0      Transient  75.0
4  240.0     NaN                    0      Transient  98.0

   required_car_parking_spaces total_of_special_requests reservation_status \
0                             0                        0      Check-Out
1                             0                        0      Check-Out
2                             0                        0      Check-Out
```

3	0	0	Check-Out
4	0	1	Check-Out

	reservation_status_date
0	01-07-15
1	01-07-15
2	02-07-15
3	02-07-15
4	03-07-15

[5 rows x 32 columns]

df.shape

(119390, 32)

df.isnull().sum()

	0
hotel	0
is_canceled	0
lead_time	0
arrival_date_year	0
arrival_date_month	0
arrival_date_week_number	0
arrival_date_day_of_month	0
stays_in_weekend_nights	0
stays_in_week_nights	0
adults	0
children	4
babies	0
meal	0
country	488
market_segment	0
distribution_channel	0
is_repeated_guest	0
previous_cancellations	0
previous_bookings_not_canceled	0
reserved_room_type	0
assigned_room_type	0
booking_changes	0
deposit_type	0
agent	16340
company	112593
days_in_waiting_list	0
customer_type	0
adr	0
required_car_parking_spaces	0
total_of_special_requests	0
reservation_status	0
reservation_status_date	0

dtype: int64

Handling missing values in the dataset

```
# 1. Fill missing values in the 'children' column with 0 (assuming no children were present)
df['children'] = df['children'].fillna(0)
```

```
# 2. Fill missing values in the 'country' column with the most frequently occurring country
```

```

most_frequent_country = df['country'].mode()[0] # Get the most repeated country
df['country'] = df['country'].fillna(most_frequent_country)

# 3. Fill missing values in the 'agent' column with 0 (indicating no agent was used)
df['agent'] = df['agent'].fillna(0).astype(int) # Convert to integer for consistency

# 4. Fill missing values in the 'company' column with 0 (indicating no company was involved)
df['company'] = df['company'].fillna(0).astype(int) # Convert to integer for consistency

# Final check: Print missing values count to confirm all are handled
print(df.isnull().sum())

```

```

hotel 0
is_canceled 0
lead_time 0
arrival_date_year 0
arrival_date_month 0
arrival_date_week_number 0
arrival_date_day_of_month 0
stays_in_weekend_nights 0
stays_in_week_nights 0
adults 0
children 0
babies 0
meal 0
country 0
market_segment 0
distribution_channel 0
is_repeated_guest 0
previous_cancellations 0
previous_bookings_not_canceled 0
reserved_room_type 0
assigned_room_type 0
booking_changes 0
deposit_type 0
agent 0
company 0
days_in_waiting_list 0
customer_type 0
adr 0
required_car_parking_spaces 0
total_of_special_requests 0
reservation_status 0
reservation_status_date 0
dtype: int64

```

2. Analytics & Reporting

Revenue trends over time: Total Revenue=adr×(stays in weekend nights+stays in week nights)

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Ensure adr is numeric
df['adr'] = pd.to_numeric(df['adr'], errors='coerce')

# Calculate revenue per booking
df['revenue'] = df['adr'] * (df['stays_in_week_nights'] + df['stays_in_weekend_nights'])

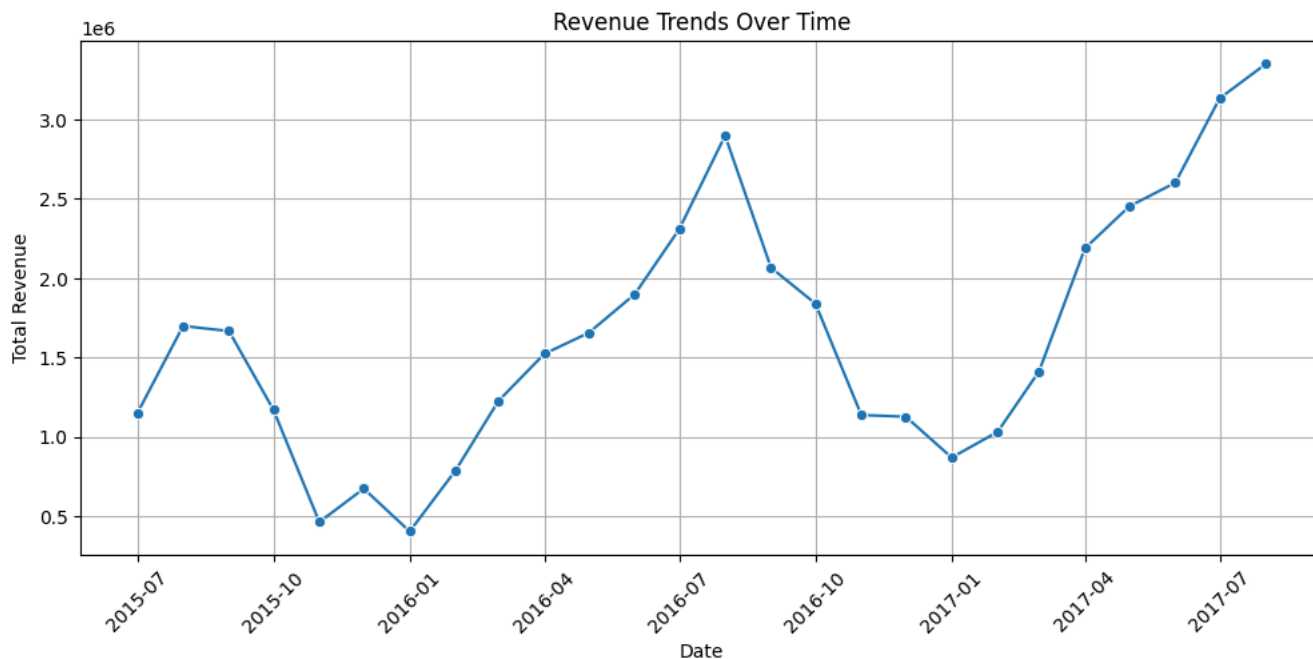
# Aggregate revenue by month & year
revenue_trend = df.groupby(['arrival_date_year', 'arrival_date_month'])['revenue'].sum().reset_index()

# Convert month names to datetime format
revenue_trend['date'] = pd.to_datetime(revenue_trend['arrival_date_year'].astype(str) + '-' + revenue_trend['arrival_date_month'])

# Plot revenue trends over time
plt.figure(figsize=(12, 5))
sns.lineplot(x='date', y='revenue', data=revenue_trend, marker='o')
plt.title("Revenue Trends Over Time")
plt.xlabel("Date")
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.grid()
plt.show()

```

```
<ipython-input-8-5e3731ec347a>:16: UserWarning: Could not infer format, so each element will be parsed individually, falling back to
revenue_trend['date'] = pd.to_datetime(revenue_trend['arrival_date_year'].astype(str) + '-' + revenue_trend['arrival_date_month'])
```



Cancellation Rate as Percentage of Total Bookings

Cancellation rate is calculated as: $\text{Cancellation Rate} = (\text{Total Cancelled Bookings} / \text{Total Bookings}) \times 100$

```
# Compute cancellation rate
cancellation_rate = (df['is_canceled'].sum() / len(df)) * 100


print(f"Cancellation Rate: {cancellation_rate:.2f}%")
```

```
Cancellation Rate: 37.04%
```

Geographical Distribution of Users

```
# Count bookings per country
country_distribution = df['country'].value_counts()

# Plot a bar chart
plt.figure(figsize=(12, 6))
sns.barplot(x=country_distribution.index[:10], y=country_distribution.values[:10], palette='coolwarm')
plt.title("Top 10 Countries by Number of Bookings")
plt.xlabel("Country")
plt.ylabel("Number of Bookings")
plt.xticks(rotation=45)
plt.show()
```

 <ipython-input-9-86e0c2f59e96>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

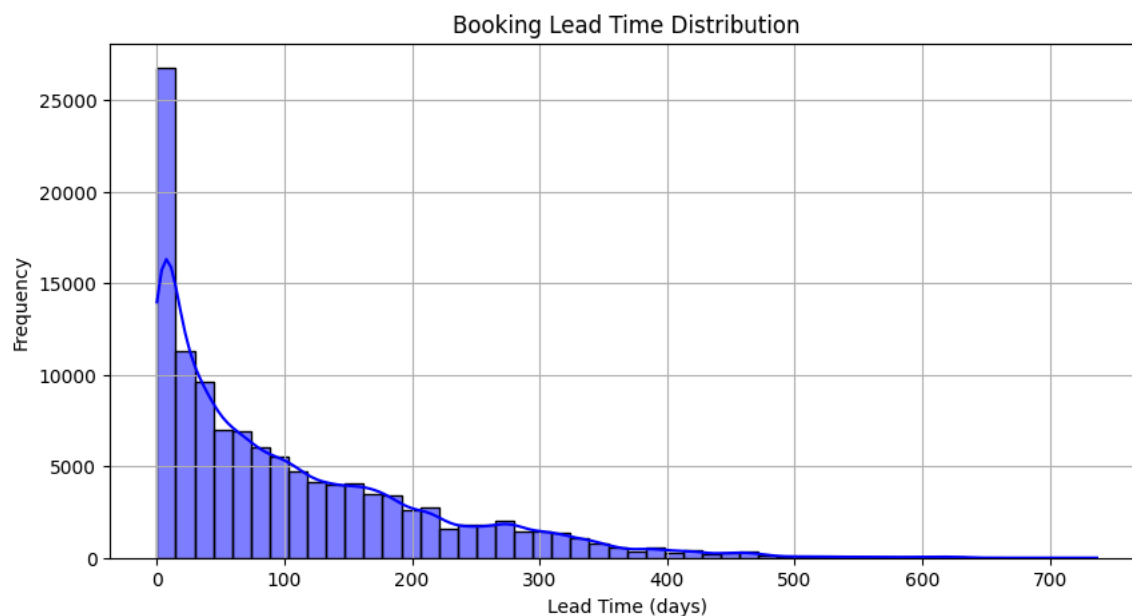
```
sns.barplot(x=country_distribution.index[:10], y=country_distribution.values[:10], palette='coolwarm')
```



Booking Lead Time

```
plt.figure(figsize=(10, 5))
sns.histplot(df['lead_time'], bins=50, kde=True, color='blue')
plt.title("Booking Lead Time Distribution")
plt.xlabel("Lead Time (days)")
plt.ylabel("Frequency")
plt.grid()
plt.show()
```

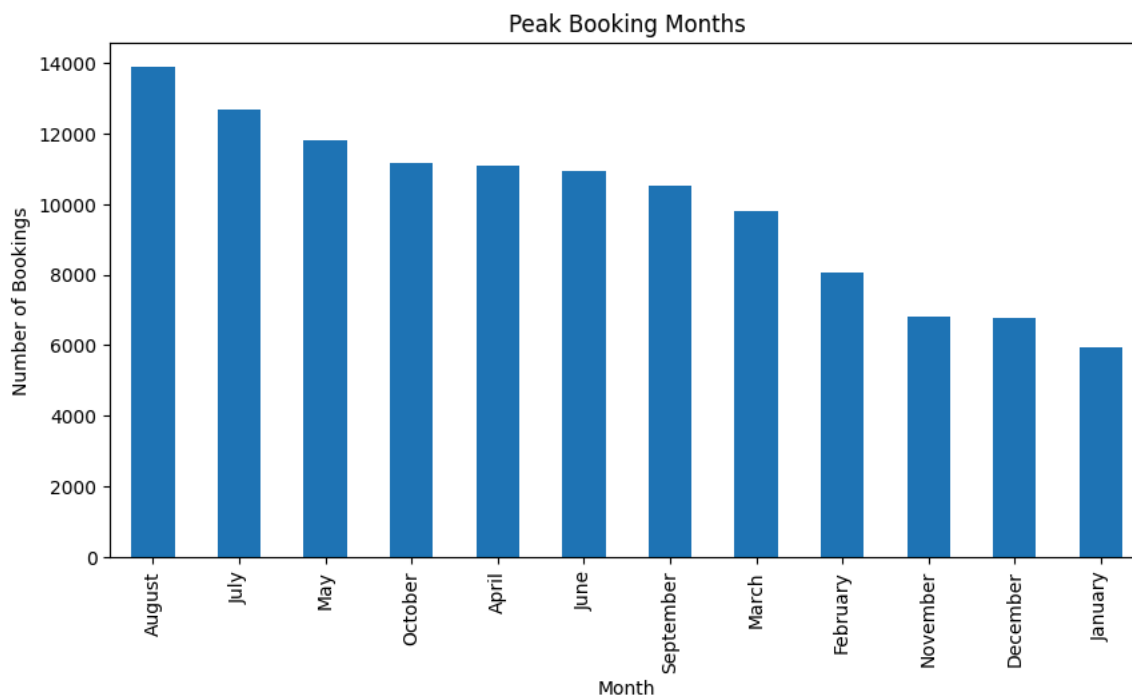




Additional Analytics (Optional)

Key Booking Insights & Trends

```
df['arrival_date_month'].value_counts().plot(kind='bar', figsize=(10,5), title="Peak Booking Months")
plt.xlabel("Month")
plt.ylabel("Number of Bookings")
plt.show()
```



**** Retrieval-Augmented Question Answering (RAG)****

```
!pip install faiss-cpu chromadb transformers langchain
```



```
Collecting faiss-cpu
  Downloading faiss_cpu-1.10.0-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (4.4 kB)
Collecting chromadb
  Downloading chromadb-0.6.3-py3-none-any.whl.metadata (6.8 kB)
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.50.0)
Requirement already satisfied: langchain in /usr/local/lib/python3.11/dist-packages (0.3.21)
Requirement already satisfied: numpy<3.0,>=1.25.0 in /usr/local/lib/python3.11/dist-packages (from faiss-cpu) (2.0.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from faiss-cpu) (24.2)
Collecting build>=1.0.3 (from chromadb)
  Downloading build-1.2.2.post1-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: pydantic>=1.9 in /usr/local/lib/python3.11/dist-packages (from chromadb) (2.10.6)
Collecting chroma-hnswlib==0.7.6 (from chromadb)
  Downloading chroma_hnswlib-0.7.6-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (252 bytes)
Collecting fastapi>=0.95.2 (from chromadb)
  Downloading fastapi-0.115.12-py3-none-any.whl.metadata (27 kB)
Collecting uvicorn>=0.18.3 (from uvicorn[standard]>=0.18.3->chromadb)
  Downloading uvicorn-0.34.0-py3-none-any.whl.metadata (6.5 kB)
Collecting posthog>=2.4.0 (from chromadb)
  Downloading posthog-3.23.0-py3-none-any.whl.metadata (3.0 kB)
Requirement already satisfied: typing_extensions>=4.5.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.12.2)
Collecting onnxruntime>=1.14.1 (from chromadb)
  Downloading onnxruntime-1.21.0-cp311-cp311-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (4.5 kB)
Requirement already satisfied: opentelemetry-api>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
Collecting opentelemetry-exporter-otlp-proto-grpc>=1.2.0 (from chromadb)
  Downloading opentelemetry_exporter_otlp_proto_grpc-1.31.1-py3-none-any.whl.metadata (2.5 kB)
Collecting opentelemetry-instrumentation-fastapi>=0.41b0 (from chromadb)
  Downloading opentelemetry_instrumentation_fastapi-0.52b1-py3-none-any.whl.metadata (2.2 kB)
Requirement already satisfied: opentelemetry-sdk>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
Requirement already satisfied: tokenizers>=0.13.2 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.21.1)
Collecting pypika>=0.48.9 (from chromadb)
  Downloading PyPika-0.48.9.tar.gz (67 kB)
67.3/67.3 kB 5.3 MB/s eta 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: tqdm>=4.65.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.67.1)
Collecting overrides>=7.3.1 (from chromadb)
  Downloading overrides-7.7.0-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.11/dist-packages (from chromadb) (6.5.2)
Requirement already satisfied: grpcio>=1.58.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.71.0)
Collecting bcrypt>=4.0.1 (from chromadb)
  Downloading bcrypt-4.3.0-cp39-abi3-manylinux_2_34_x86_64.whl.metadata (10 kB)
Requirement already satisfied: typer>=0.9.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.15.2)
Collecting kubernetes>=28.1.0 (from chromadb)
  Downloading kubernetes-32.0.1-py2.py3-none-any.whl.metadata (1.7 kB)
```

```
Requirement already satisfied: tenacity>=8.2.3 in /usr/local/lib/python3.11/dist-packages (from chromadb) (9.0.0)
Requirement already satisfied: PyYAML>=6.0.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (6.0.2)
Collecting mmh3>=4.0.1 (from chromadb)
  Downloading mmh3-5.1.0-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl.metadata
Requirement already satisfied: orjson>=3.9.12 in /usr/local/lib/python3.11/dist-packages (from chromadb) (3.10.15)
Requirement already satisfied: httpx>=0.27.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.28.1)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (13.9.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.26.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.29.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
```

```
#Convert Booking Data into Text
import pandas as pd
```

```
# Load your processed DataFrame
df['text'] = df.apply(lambda row: f"Booking ID: {row.name}, Hotel: {row['hotel']}, ADR: {row['adr']}, Country: {row['country']}, Canceled: {row['canceled']},
```

```
# Convert DataFrame to a list of text records
documents = df['text'].tolist()
```

```
from sentence_transformers import SentenceTransformer
```

```
# Load a pre-trained model for text embeddings
model = SentenceTransformer('all-MiniLM-L6-v2')
```

```
# Convert text data into embeddings
embeddings = model.encode(documents, convert_to_tensor=True)
```

```
⚠ /usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as :
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
```

```
warnings.warn(
modules.json: 100% 349/349 [00:00<00:00, 31.4kB/s]
config_sentence_transformers.json: 100% 116/116 [00:00<00:00, 8.07kB/s]
README.md: 100% 10.5k/10.5k [00:00<00:00, 804kB/s]
sentence_bert_config.json: 100% 53.0/53.0 [00:00<00:00, 3.12kB/s]
config.json: 100% 612/612 [00:00<00:00, 53.5kB/s]
model.safetensors: 100% 90.9M/90.9M [00:00<00:00, 179MB/s]
tokenizer_config.json: 100% 350/350 [00:00<00:00, 23.2kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 1.85MB/s]
tokenizer.json: 100% 466k/466k [00:00<00:00, 3.40MB/s]
special_tokens_map.json: 100% 112/112 [00:00<00:00, 8.24kB/s]
config.json: 100% 190/190 [00:00<00:00, 19.4kB/s]
```

```
import faiss
import numpy as np
```

```
# Convert embeddings to a NumPy array
embeddings_np = np.array(embeddings.cpu())
```

```
# Initialize FAISS index
dimension = embeddings_np.shape[1]
index = faiss.IndexFlatL2(dimension)
```

```
# Add embeddings to FAISS
index.add(embeddings_np)
```

```
print("FAISS Index built successfully!")
```

```
⚡ FAISS Index built successfully!
```

```
from transformers import pipeline
import numpy as np
from sentence_transformers import SentenceTransformer
import faiss
```

```
# Load sentence transformer model for embeddings
model = SentenceTransformer("all-MiniLM-L6-v2")

# Load a small LLM for testing (change to a stronger model if needed)
qa_model = pipeline("text-generation", model="EleutherAI/gpt-neo-1.3B")

# Sample FAISS index setup (replace with your real index)
dimension = 384 # Depends on embedding model
index = faiss.IndexFlatL2(dimension)
documents = [
    "The average price of a hotel booking is $120.",
    "Hotel bookings in July 2017 generated $1M in revenue.",
    "Portugal had the highest number of bookings."
]

# Encode and store embeddings in FAISS
doc_embeddings = model.encode(documents)
index.add(np.array(doc_embeddings))

def rag_qa(question, top_k=3):
    # Convert question to embedding
    query_embedding = model.encode([question])

    # Search in FAISS index
    distances, indices = index.search(np.array(query_embedding), top_k)

    # Retrieve the top results
    retrieved_texts = [documents[i] for i in indices[0]]

    # Combine retrieved context
    context = " ".join(retrieved_texts)

    # Generate answer using LLM
    prompt = f"Context: {context} \n Question: {question} \n Answer:"
    response = qa_model(prompt, max_new_tokens=50)[0]['generated_text']

    return response

# Test Example
question = "What is the average price of a hotel booking?"
answer = rag_qa(question)
print(answer)
```

Device set to use cuda:0
 Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
 Context: The average price of a hotel booking is \$120. Hotel bookings in July 2017 generated \$1M in revenue. Portugal had the highest number of bookings.
 Question: What is the average price of a hotel booking?
 Answer: The average price of a hotel booking is \$130.

B.
 Question: How many hotels did Portugal have in July 2017? For the most part, they are listed below with the number of rooms. Are there any other hotels in Portugal that are not listed below?

!pip install groq chromadb sentence-transformers

Collecting groq
 Downloading groq-0.20.0-py3-none-any.whl.metadata (15 kB)
 Requirement already satisfied: chromadb in /usr/local/lib/python3.11/dist-packages (0.6.3)
 Requirement already satisfied: sentence-transformers in /usr/local/lib/python3.11/dist-packages (3.4.1)
 Requirement already satisfied: anyio<5,>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from groq) (4.9.0)
 Requirement already satisfied: distro<2,>=1.7.0 in /usr/local/lib/python3.11/dist-packages (from groq) (1.9.0)
 Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from groq) (0.28.1)
 Requirement already satisfied: pydantic<3,>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from groq) (2.10.6)
 Requirement already satisfied: sniffio in /usr/local/lib/python3.11/dist-packages (from groq) (1.3.1)
 Requirement already satisfied: typing-extensions<5,>=4.10 in /usr/local/lib/python3.11/dist-packages (from groq) (4.12.2)
 Requirement already satisfied: build<1.0.3 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.2.2.post1)
 Requirement already satisfied: chroma-hnswlib<0.7.6 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.7.6)
 Requirement already satisfied: fastapi<0.95.2 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.115.12)
 Requirement already satisfied: uvicorn<0.18.3 in /usr/local/lib/python3.11/dist-packages (from uvicorn[standard]>=0.18.3->chroma) (0.18.3)
 Requirement already satisfied: numpy<1.22.5 in /usr/local/lib/python3.11/dist-packages (from chromadb) (2.0.2)
 Requirement already satisfied: posthog<2.4.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (3.23.0)
 Requirement already satisfied: onnxruntime<1.14.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.21.0)
 Requirement already satisfied: opentelemetry-api<1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
 Requirement already satisfied: opentelemetry-exporter-otlp-proto-grpc<1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
 Requirement already satisfied: opentelemetry-instrumentation-fastapi<0.41b0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.41b0)
 Requirement already satisfied: opentelemetry-sdk<1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
 Requirement already satisfied: tokenizers<0.13.2 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.21.1)
 Requirement already satisfied: pypika<0.48.9 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.48.9)
 Requirement already satisfied: tqdm<4.65.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.67.1)
 Requirement already satisfied: overrides<7.3.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (7.7.0)
 Requirement already satisfied: importlib-resources in /usr/local/lib/python3.11/dist-packages (from chromadb) (6.5.2)
 Requirement already satisfied: grpcio<1.58.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.71.0)


```
Requirement already satisfied: bcrypt>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.3.0)
Requirement already satisfied: typer>=0.9.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.15.2)
Requirement already satisfied: kubernetes>=28.1.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (32.0.1)
Requirement already satisfied: tenacity>=8.2.3 in /usr/local/lib/python3.11/dist-packages (from chromadb) (9.0.0)
Requirement already satisfied: PyYAML>=6.0.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (6.0.2)
Requirement already satisfied: mmh3>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (5.1.0)
Requirement already satisfied: orjson>=3.9.12 in /usr/local/lib/python3.11/dist-packages (from chromadb) (3.10.15)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (13.9.4)
Requirement already satisfied: transformers<5.0.0,>=4.41.0 in /usr/local/lib/python3.11/dist-packages (from sentence-transformers) (2.6.0+cu124)
Requirement already satisfied: torch>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from sentence-transformers) (2.6.0+cu124)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (from sentence-transformers) (1.6.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from sentence-transformers) (1.14.1)
Requirement already satisfied: huggingface-hub>=0.20.0 in /usr/local/lib/python3.11/dist-packages (from sentence-transformers) (0.20.0)
Requirement already satisfied: Pillow in /usr/local/lib/python3.11/dist-packages (from sentence-transformers) (11.1.0)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5,>=3.5.0->groq) (3.10)
Requirement already satisfied: packaging>=19.1 in /usr/local/lib/python3.11/dist-packages (from build>=1.0.3->chromadb) (24.2)
Requirement already satisfied: pyproject_hooks in /usr/local/lib/python3.11/dist-packages (from build>=1.0.3->chromadb) (1.2.0)
Requirement already satisfied: starlette<0.47.0,>=0.40.0 in /usr/local/lib/python3.11/dist-packages (from fastapi>=0.95.2->chroma) (0.47.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->groq) (2025.1.31)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->groq) (1.0.7)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->groq) (0.14.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.20.0->sentence-transformers) (3.16.0)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.20.0->sentence-transformers) (2025.10.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.20.0->sentence-transformers) (2.32.0)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (1.17.0)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.9.0)
Requirement already satisfied: google-auth>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.32.0)
Requirement already satisfied: websocket-client!=0.40.0,!0.41.*,!0.42.*,>=0.32.0 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (1.7.0)
Requirement already satisfied: requests-oauthlib in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.0.0)
Requirement already satisfied: oauthlib>=3.2.2 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (3.3.1)
```

```
!pip install chromadb groq
```

```
Requirement already satisfied: chromadb in /usr/local/lib/python3.11/dist-packages (0.6.3)
Requirement already satisfied: groq in /usr/local/lib/python3.11/dist-packages (0.20.0)
Requirement already satisfied: build>=1.0.3 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.2.2.post1)
Requirement already satisfied: pydantic>=1.9 in /usr/local/lib/python3.11/dist-packages (from chromadb) (2.10.6)
Requirement already satisfied: chroma-hnswlib==0.7.6 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.7.6)
Requirement already satisfied: fastapi>=0.95.2 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.115.12)
Requirement already satisfied: uvicorn>=0.18.3 in /usr/local/lib/python3.11/dist-packages (from uvicorn[standard]>=0.18.3->chroma) (0.30.1)
Requirement already satisfied: numpy>=1.22.5 in /usr/local/lib/python3.11/dist-packages (from chromadb) (2.0.2)
Requirement already satisfied: posthog>=2.4.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (3.23.0)
Requirement already satisfied: typing_extensions>=4.5.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.12.2)
Requirement already satisfied: onnxruntime>=1.14.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.21.0)
Requirement already satisfied: opentelemetry-api>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
Requirement already satisfied: opentelemetry-exporter-otlp-proto-grpc>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
Requirement already satisfied: opentelemetry-instrumentation-fastapi>=0.41b0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.41b0)
Requirement already satisfied: opentelemetry-sdk>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
Requirement already satisfied: tokenizers>=0.13.2 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.21.1)
Requirement already satisfied: pypika>=0.48.9 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.48.9)
Requirement already satisfied: tqdm>=4.65.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.67.1)
Requirement already satisfied: overrides>=7.3.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (7.7.0)
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.11/dist-packages (from chromadb) (6.5.2)
Requirement already satisfied: grpcio>=1.58.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.71.0)
Requirement already satisfied: bcrypt>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.3.0)
Requirement already satisfied: typer>=0.9.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.15.2)
Requirement already satisfied: kubernetes>=28.1.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (32.0.1)
Requirement already satisfied: tenacity>=8.2.3 in /usr/local/lib/python3.11/dist-packages (from chromadb) (9.0.0)
Requirement already satisfied: PyYAML>=6.0.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (6.0.2)
Requirement already satisfied: mmh3>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (5.1.0)
Requirement already satisfied: orjson>=3.9.12 in /usr/local/lib/python3.11/dist-packages (from chromadb) (3.10.15)
Requirement already satisfied: httpx>=0.27.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.28.1)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (13.9.4)
Requirement already satisfied: anyio<5,>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from groq) (4.9.0)
Requirement already satisfied: distro<2,>=1.7.0 in /usr/local/lib/python3.11/dist-packages (from groq) (1.9.0)
Requirement already satisfied: sniffio in /usr/local/lib/python3.11/dist-packages (from groq) (1.3.1)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5,>=3.5.0->groq) (3.10)
Requirement already satisfied: packaging>=19.1 in /usr/local/lib/python3.11/dist-packages (from build>=1.0.3->chromadb) (24.2)
Requirement already satisfied: pyproject_hooks in /usr/local/lib/python3.11/dist-packages (from build>=1.0.3->chromadb) (1.2.0)
Requirement already satisfied: starlette<0.47.0,>=0.40.0 in /usr/local/lib/python3.11/dist-packages (from fastapi>=0.95.2->chroma) (0.47.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx>=0.27.0->chromadb) (2025.1.31)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.27.0->chromadb) (1.0.7)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.27.0->chromadb) (0.14.0)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (1.17.0)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chroma) (2.9.0)
Requirement already satisfied: google-auth>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.32.0)
Requirement already satisfied: websocket-client!=0.40.0,!0.41.*,!0.42.*,>=0.32.0 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (1.7.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.32.3)
Requirement already satisfied: requests-oauthlib in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.0.0)
Requirement already satisfied: oauthlib>=3.2.2 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (3.3.1)
Requirement already satisfied: urllib3>=1.24.2 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.2.3)
Requirement already satisfied: durationpy>=0.7 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (0.9.0)
Requirement already satisfied: coloredlogs in /usr/local/lib/python3.11/dist-packages (from onnxruntime>=1.14.1->chromadb) (15.0.1)
Requirement already satisfied: flatbuffers in /usr/local/lib/python3.11/dist-packages (from onnxruntime>=1.14.1->chromadb) (25.2.1)
Requirement already satisfied: protobuf in /usr/local/lib/python3.11/dist-packages (from onnxruntime>=1.14.1->chromadb) (5.29.4)
Requirement already satisfied: sympy in /usr/local/lib/python3.11/dist-packages (from onnxruntime>=1.14.1->chromadb) (1.13.1)
Requirement already satisfied: deprecated>=1.2.6 in /usr/local/lib/python3.11/dist-packages (from opentelemetry-api>=1.2.0->chroma) (1.2.15)
```

Requirement already satisfied: importlib-metadata<8.7.0,>=6.0 in /usr/local/lib/python3.11/dist-packages (from opentelemetry-api>
 Requirement already satisfied: googleapis-common-protos~=1.52 in /usr/local/lib/python3.11/dist-packages (from opentelemetry-expo

```
import os
import chromadb
from groq import Groq

# Set the API key directly (⚠ Not recommended for security reasons)
GROQ_API_KEY = "gsk-GnvIMBaJ7ibAHG7c0p52WGdyb3FYVj0Y1IHZfdT6G2pRgwL917vD"

# Initialize the Groq LLaMA 2 model
groq_client = Groq(api_key=GROQ_API_KEY)

# Initialize ChromaDB for storing vector embeddings
chroma_client = chromadb.PersistentClient(path="./chroma_db")
collection = chroma_client.get_or_create_collection(name="hotel_bookings")

# Function to insert data into ChromaDB
def insert_data(documents):
    for i, doc in enumerate(documents):
        collection.add(
            ids=[str(i)], # Unique ID for each document
            documents=[doc] # The actual text data
        )
    print("Documents inserted into ChromaDB.")

# Function to perform retrieval-augmented generation (RAG)
def rag_qa(question, top_k=3):
    # Search for relevant documents in ChromaDB
    results = collection.query(query_texts=[question], n_results=top_k)


    # Extract relevant context from retrieved documents
    retrieved_texts = results["documents"][0] if results["documents"] else []
    context = " ".join(retrieved_texts)

    # Generate an answer using the LLaMA 2 model
    response = groq_client.chat.completions.create(
        model="llama3-70b-8192",
        messages=[
            {"role": "system", "content": "You are an AI assistant providing hotel booking insights."},
            {"role": "user", "content": f"Context: {context}\nQuestion: {question}\nAnswer:"}
        ]
    )

    return response.choices[0].message.content

# Insert sample data (replace with actual hotel booking data)
sample_documents = [
    "The total revenue for July 2017 was $1 million.",
    "The highest booking cancellations were in Portugal.",
    "The average price of a hotel booking is $120."
]
insert_data(sample_documents)


# Example query
question = "What is the average price of a hotel booking?"
answer = rag_qa(question)
print("Answer:", answer)
```

 /root/.cache/chroma/onnx_models/all-MiniLM-L6-v2/onnx.tar.gz: 100%|██████████| 79.3M/79.3M [00:02<00:00, 36.4MiB/s]
 Documents inserted into ChromaDB.
 Answer: The average price of a hotel booking is \$120.

```
from groq import Groq

GROQ_API_KEY = "my_groq_api_key"
groq_client = Groq(api_key=GROQ_API_KEY)

# Print the response structure
models = groq_client.models.list()
print(models)
```

 ModelListResponse(data=[Model(id='mistral-saba-24b', created=1739996492, object='model', owned_by='Mistral AI', active=True, context

```
!pip install fastapi uvicorn chromadb groq requests
```

```

Requirement already satisfied: fastapi in /usr/local/lib/python3.11/dist-packages (0.115.12)
Requirement already satisfied: uvicorn in /usr/local/lib/python3.11/dist-packages (0.34.0)
Requirement already satisfied: chromadb in /usr/local/lib/python3.11/dist-packages (0.6.3)
Requirement already satisfied: groq in /usr/local/lib/python3.11/dist-packages (0.20.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (2.32.3)
Requirement already satisfied: starlette<0.47.0,>=0.40.0 in /usr/local/lib/python3.11/dist-packages (from fastapi) (0.46.1)
Requirement already satisfied: pydantic!=1.8,!1.8.1,!2.0.0,!2.0.1,!2.1.0,<3.0.0,>=1.7.4 in /usr/local/lib/python3.11/dist-packages (from fastapi) (2.10.6)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.11/dist-packages (from fastapi) (4.12.2)
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.11/dist-packages (from uvicorn) (8.1.8)
Requirement already satisfied: h11>=0.8 in /usr/local/lib/python3.11/dist-packages (from uvicorn) (0.14.0)
Requirement already satisfied: build>=1.0.3 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.2.2.post1)
Requirement already satisfied: chroma-hnswlib==0.7.6 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.7.6)
Requirement already satisfied: numpy>=1.22.5 in /usr/local/lib/python3.11/dist-packages (from chromadb) (2.0.2)
Requirement already satisfied: posthog>=2.4.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (3.23.0)
Requirement already satisfied: onnxruntime>=1.14.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.21.0)
Requirement already satisfied: opentelemetry-api>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
Requirement already satisfied: opentelemetry-exporter-otlp-proto-grpc>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
Requirement already satisfied: opentelemetry-instrumentation-fastapi>=0.41b0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.51b0)
Requirement already satisfied: opentelemetry-sdk>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.31.1)
Requirement already satisfied: tokenizers>=0.13.2 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.21.1)
Requirement already satisfied: pypika>=0.48.9 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.48.9)
Requirement already satisfied: tqdm>=4.65.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.67.1)
Requirement already satisfied: overrides>=7.3.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (7.7.0)
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.11/dist-packages (from chromadb) (6.5.2)
Requirement already satisfied: grpcio>=1.58.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (1.71.0)
Requirement already satisfied: bcrypt>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (4.3.0)
Requirement already satisfied: typer>=0.9.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.15.2)
Requirement already satisfied: kubernetes>=28.1.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (32.0.1)
Requirement already satisfied: tenacity>=8.2.3 in /usr/local/lib/python3.11/dist-packages (from chromadb) (9.0.0)
Requirement already satisfied: PyYAML>=6.0.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (6.0.2)
Requirement already satisfied: mmh3>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from chromadb) (5.1.0)
Requirement already satisfied: orjson>=3.9.12 in /usr/local/lib/python3.11/dist-packages (from chromadb) (3.10.15)
Requirement already satisfied: httpx>=0.27.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (0.28.1)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from chromadb) (13.9.4)
Requirement already satisfied: anyio<5,>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from groq) (4.9.0)
Requirement already satisfied: distro<2,>=1.7.0 in /usr/local/lib/python3.11/dist-packages (from groq) (1.9.0)
Requirement already satisfied: sniffio in /usr/local/lib/python3.11/dist-packages (from groq) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests) (2025.1.31)
Requirement already satisfied: packaging>=19.1 in /usr/local/lib/python3.11/dist-packages (from build>=1.0.3->chromadb) (24.2)
Requirement already satisfied: pyproject_hooks in /usr/local/lib/python3.11/dist-packages (from build>=1.0.3->chromadb) (1.2.0)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.27.0->chromadb) (1.0.7)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (1.17.0)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.9.0.post0)
Requirement already satisfied: google-auth>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.38.0)
Requirement already satisfied: websocket-client!=0.40.0,!0.41.*,!0.42.*,>=0.32.0 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (1.7.0)
Requirement already satisfied: requests-oauthlib in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (2.0.0)
Requirement already satisfied: oauthlib>=3.2.2 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (3.2.2)
Requirement already satisfied: durationpy>=0.7 in /usr/local/lib/python3.11/dist-packages (from kubernetes>=28.1.0->chromadb) (0.9.0)
Requirement already satisfied: coloredlogs in /usr/local/lib/python3.11/dist-packages (from onnxruntime>=1.14.1->chromadb) (15.0.1)
Requirement already satisfied: flatbuffers in /usr/local/lib/python3.11/dist-packages (from onnxruntime>=1.14.1->chromadb) (25.2.1)
Requirement already satisfied: protobuf in /usr/local/lib/python3.11/dist-packages (from onnxruntime>=1.14.1->chromadb) (5.29.4)
Requirement already satisfied: sympy in /usr/local/lib/python3.11/dist-packages (from onnxruntime>=1.14.1->chromadb) (1.13.1)
Requirement already satisfied: deprecated>=1.2.6 in /usr/local/lib/python3.11/dist-packages (from opentelemetry-api>=1.2.0->chromadb) (1.2.15)
Requirement already satisfied: importlib-metadata<8.7.0,>=6.0 in /usr/local/lib/python3.11/dist-packages (from opentelemetry-api>=1.2.0->chromadb) (8.5.0)

```

```
!pip install pyngrok
```

```

Collecting pyngrok
  Downloading pyngrok-7.2.3-py3-none-any.whl.metadata (8.7 kB)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.11/dist-packages (from pyngrok) (6.0.2)
  Downloading pyngrok-7.2.3-py3-none-any.whl (23 kB)
Installing collected packages: pyngrok
Successfully installed pyngrok-7.2.3

```

```
!ngrok authtoken 2uqlra8OV2bZAVqL8Jbyy9EbFE0_4B1KTAUhYgWgajRdMdq
```

```
Auth token saved to configuration file: /root/.config/ngrok/ngrok.yml
```

```
from pyngrok import ngrok
```

```

# Start a tunnel to expose FastAPI
public_url = ngrok.connect(8000).public_url
print(f"Public API URL: {public_url}")

```

```
Public API URL: https://a998-34-53-13-17.ngrok-free.app
```

```
!kill -f ngrok
```

```
!pip install flask flask-cors
```

```

Requirement already satisfied: flask in /usr/local/lib/python3.11/dist-packages (3.1.0)
Requirement already satisfied: flask-cors in /usr/local/lib/python3.11/dist-packages (5.0.1)
Requirement already satisfied: Werkzeug>=3.1 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.6)
Requirement already satisfied: itsdangerous>=2.2 in /usr/local/lib/python3.11/dist-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.11/dist-packages (from flask) (8.1.8)
Requirement already satisfied: blinker>=1.9 in /usr/local/lib/python3.11/dist-packages (from flask) (1.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from Jinja2>=3.1.2->flask) (3.0.2)

```

```

from flask import Flask, request, jsonify
from flask_cors import CORS
import threading

```

```

app = Flask(__name__)
CORS(app) # Enable CORS for cross-origin requests

```

```

@app.route("/", methods=["GET"])
def home():
    return jsonify({"message": "Flask API is running!"})

```

```

@app.route("/ask", methods=["POST"])
def ask():
    data = request.json
    question = data.get("question", "")

    # Dummy response (Replace with your Llama 2 API logic)
    response = {"answer": f"You asked: {question}, but I don't have a real answer yet."}

    return jsonify(response)

```

```

def run():
    app.run(host="0.0.0.0", port=5000)

```

```

# Run Flask in a background thread
threading.Thread(target=run).start()

```

```

!pip install pyngrok
from pyngrok import ngrok

```

```

# Authenticate with your ngrok token (replace 'your-authtoken' with your actual token)
!ngrok authtoken 2uqLra8OV2bZAVqL8Jbyy9EbFE0_4B1KTACuhUgYGwajRdMdq

```

```

# Expose Flask API
public_url = ngrok.connect(5000).public_url
print(f"Public API URL: {public_url}")

```

```

Requirement already satisfied: pyngrok in /usr/local/lib/python3.11/dist-packages (7.2.3)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.11/dist-packages (from pyngrok) (6.0.2)
Auth token saved to configuration file: /root/.config/ngrok/ngrok.yml
Public API URL: https://056a-34-53-13-17.ngrok-free.app

```

```

from flask import Flask, request, jsonify
import chromadb
import requests

```

```
app = Flask(__name__)
```

```

# Initialize ChromaDB Client
chroma_client = chromadb.PersistentClient(path="./chroma_db")
collection = chroma_client.get_or_create_collection("hotel_data")

```

```

# Insert Sample Data into ChromaDB
sample_documents = [
    "The total revenue for July 2017 was $1 million.",
    "Portugal had the highest number of booking cancellations.",
    "The average price of a hotel booking is $120."
]
for idx, doc in enumerate(sample_documents):
    collection.add(ids=[str(idx)], documents=[doc])

```

```

# Groq API Configuration
GROQ_API_KEY = "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" # Replace with your API key
LLAMA_MODEL = "llama2-70b-chat"

```

```

# Function to call Groq API
def get_llama2_answer(question, context):

```

```

url = "https://api.groq.com/v1/chat/completions"
headers = {"Authorization": f"Bearer {GROQ_API_KEY}", "Content-Type": "application/json"}
data = {
    "model": LLAMA_MODEL,
    "messages": [
        {"role": "system", "content": "You are an AI that answers questions based on hotel booking data."},
        {"role": "user", "content": f"Question: {question}\nRelevant Info: {context}"}
    ],
    "temperature": 0.7
}
response = requests.post(url, headers=headers, json=data)
return response.json()["choices"][0]["message"]["content"]

# API Endpoint to Ask Questions
@app.route("/ask", methods=["POST"])
def ask():
    data = request.get_json()
    question = data.get("question", "")


    # Retrieve relevant data from ChromaDB
    results = collection.query(query_texts=[question], n_results=1)
    relevant_info = results["documents"][0][0] if results["documents"] else "No relevant information found."

    # Get answer from Llama 2
    answer = get_llama2_answer(question, relevant_info)


    return jsonify({"answer": answer})

# Run API
if __name__ == "__main__":
    app.run(port=8000, debug=True)

```

 WARNING:chromadb.segment.impl.metadata.sqlite:Insert of existing embedding ID: 0
 WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 0
 WARNING:chromadb.segment.impl.metadata.sqlite:Insert of existing embedding ID: 1
 WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 1
 WARNING:chromadb.segment.impl.metadata.sqlite:Insert of existing embedding ID: 2
 WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 2
 * Serving Flask app '__main__'
 * Debug mode: on
 INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
 * Running on <http://127.0.0.1:8000>
 INFO:werkzeug:Press CTRL+C to quit
 INFO:werkzeug: * Restarting with stat

```
!pip install flask flask-ngrok
```

 Requirement already satisfied: flask in /usr/local/lib/python3.11/dist-packages (3.1.0)
 Collecting flask-ngrok
 Downloading flask_ngrok-0.0.25-py3-none-any.whl.metadata (1.8 kB)
 Requirement already satisfied: Werkzeug>=3.1 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.3)
 Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.6)
 Requirement already satisfied: itsdangerous>=2.2 in /usr/local/lib/python3.11/dist-packages (from flask) (2.2.0)
 Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.11/dist-packages (from flask) (8.1.8)
 Requirement already satisfied: blinker>=1.9 in /usr/local/lib/python3.11/dist-packages (from flask) (1.9.0)