Q & A Chatbot

24/7 Insurance Customer Support

1. Improved User Experience

Provides **instant answers** to fraud-related questions, reducing the need for customers or agents to sift through documents.

Offers **24/7 availability**, ensuring users can get help anytime.

Supports **natural language queries**, making it easy for non-technical users to interact.

2. Enhanced Fraud Detection & Prevention

Educates users about **fraud indicators** and preventive measures.

Guides claimants and policyholders to **report suspicious activities** effectively.

Helps internal teams quickly **access fraud policies and case precedents**

3. Operational Efficiency

Reduces the workload on **human agents**, freeing them for complex fraud cases.

Automates repetitive **FAQs** about fraudulent claims, investigations, and reporting.

Improves response consistency, ensuring accurate information every time

4. Better Decision-Making with AI & NLP

Can be integrated with your **fraud risk prediction model**, providing contextual responses based on claim risk scores.

Leverages **BERT embeddings and mT5/mBART** to generate precise and context-aware answers.

Helps **investigators** by summarizing fraud trends and past cases

5. Scalability & Cost Savings

Handles multiple queries simultaneously, reducing customer support costs.

Easily updated with **new fraud patterns and policies**, keeping information relevant

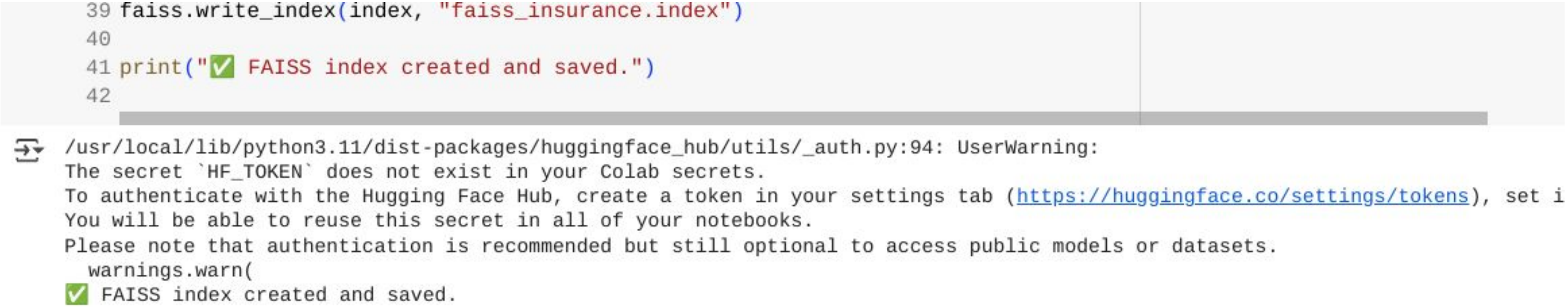
1. Use a Retrieval-Based Approach (BERT + FAISS)

This method works best if you want **fact-based answers from structured data** (e.g., "What is the average Claim Amount?").

Steps:

1. Convert important columns into **text embeddings** using **BERT**.
2. Store embeddings in a **FAISS** index for fast retrieval.
3. When a user asks a question, find the most relevant data points and return an answer.

Step 1: Create & Save FAISS Index (Run This Once)

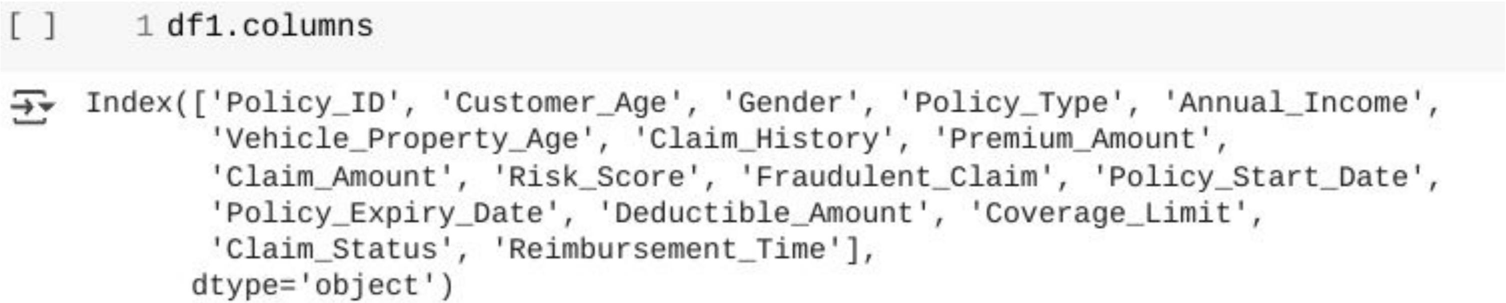


```python
faiss.write_index(index, "faiss_insurance.index")

print("✅ FAISS index created and saved.")

```

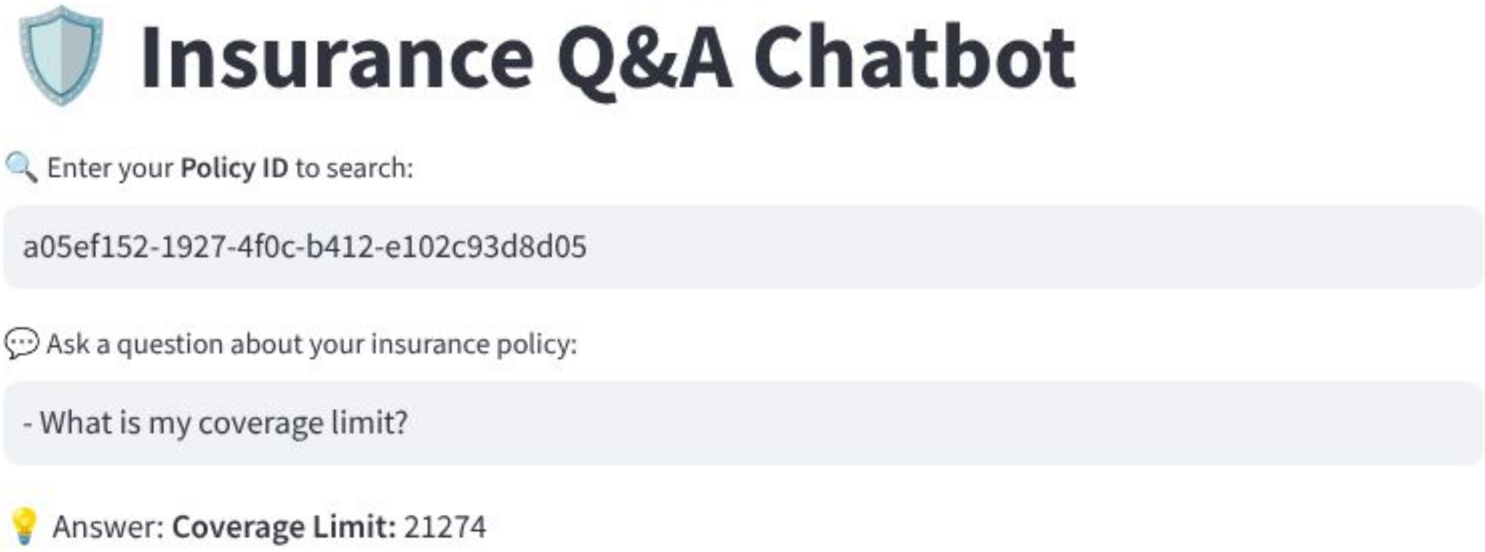
```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set i
You will be able to reuse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



```python
df1.columns
```

```
Index(['Policy_ID', 'Customer_Age', 'Gender', 'Policy_Type', 'Annual_Income',
       'Vehicle_Property_Age', 'Claim_History', 'Premium_Amount',
       'Claim_Amount', 'Risk_Score', 'Fraudulent_Claim', 'Policy_Start_Date',
       'Policy_Expiry_Date', 'Deductible_Amount', 'Coverage_Limit',
       'Claim_Status', 'Reimbursement_Time'],
      dtype='object')
```

Streamlit



🛡️ Insurance Q&A Chatbot

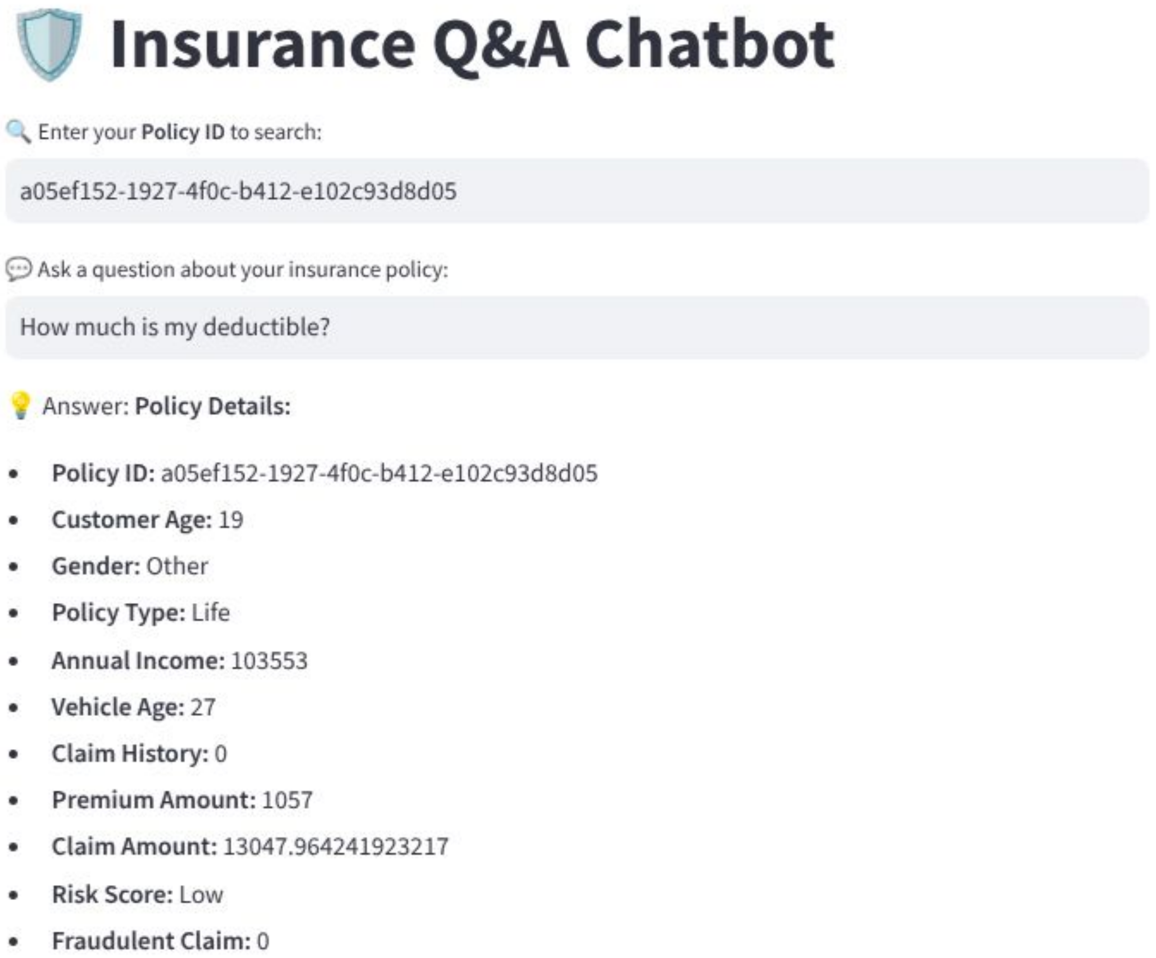
🔍 Enter your **Policy ID** to search:

a05ef152-1927-4f0c-b412-e102c93d8d05

💬 Ask a question about your insurance policy:

- What is my coverage limit?

💡 Answer: **Coverage Limit:** 21274



🛡️ Insurance Q&A Chatbot

🔍 Enter your **Policy ID** to search:

a05ef152-1927-4f0c-b412-e102c93d8d05

💬 Ask a question about your insurance policy:

How much is my deductible?

💡 Answer: **Policy Details:**

- **Policy ID:** a05ef152-1927-4f0c-b412-e102c93d8d05
- **Customer Age:** 19
- **Gender:** Other
- **Policy Type:** Life
- **Annual Income:** 103553
- **Vehicle Age:** 27
- **Claim History:** 0
- **Premium Amount:** 1057
- **Claim Amount:** 13047.964241923217
- **Risk Score:** Low
- **Fraudulent Claim:** 0

Thank You