

BUILD YOUR OWN **CHATBOT**

with RAG and Vector Database



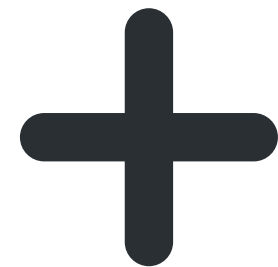
LangChain



HUGGING FACE



Chroma



Streamlit



Like | Share | Subscribe 

Content

- Technologies & Architecture
- Vector Database Implementation
- Application Development
- Prompt Engineering
- Evaluation and Testing

Technologies Used

Tools:

**Retrieval-Augmented Generation
(RAG)**



LangChain



HUGGING FACE



Chroma

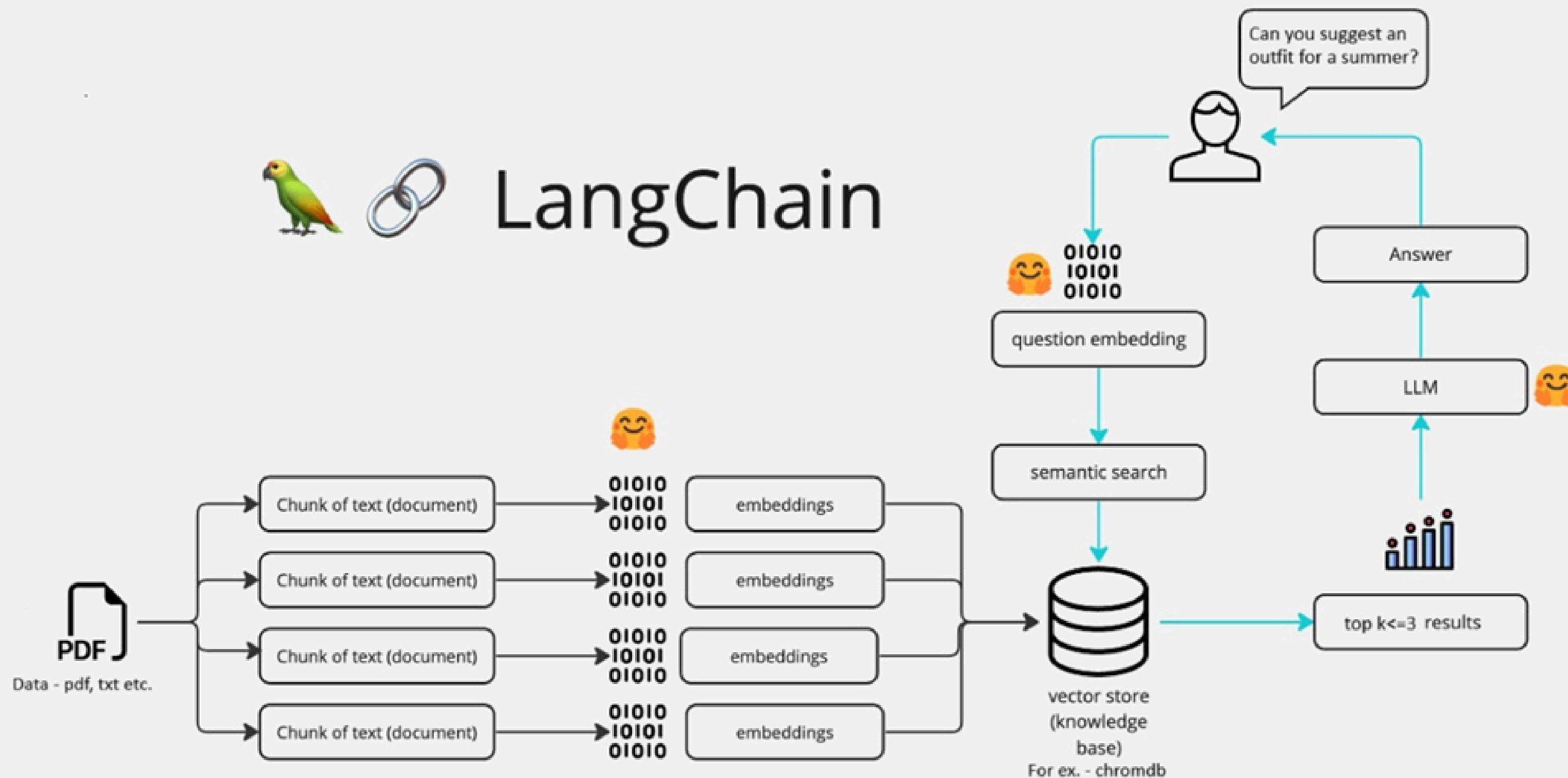


Streamlit

Langchain Architecture



LangChain



Process

- 01 Data Loading:** using **LangChain's PyPDFLoader** to read pdf document and extract text
- 02 Text Chunking:** The extracted text is divided into smaller chunks using **LangChain's RecursiveCharacterTextSplitter**
- 03 LLM Model:** Employs Hugging Face's Transformers, specifically the **sentence-transformers/all-MiniLM-L6-v2 model**, to generate vector representations (embeddings) of the text chunks.
- 04 Similarity Matching:**
When a query is received, the system performs a semantic similarity search using the **Chroma** vector database.
- 05 Response Generation:**
The relevant text chunks are passed to the Hugging Face Hub Language Model - **meta-llama/Meta-Llama-3-8B-Instruct**

Lets Start Building our Chatbot!

FashionBot: Your AI Style Companion

Hi! 🤖 I'm here to help you with your fashion choices. What would you like to know or explore?

Here are some areas you might be interested in:

1. **Fashion Trends** 🌿 👗
2. **Personal Styling Advice** 👠 🗣️
3. **Seasonal Wardrobe Selections** 🌞
4. **Accessory Recommendations** 💍

Feel free to ask me anything about fashion!

Discover the AI Styling Recommendations 👗 🏠

 Hi there! I'm your FashionBot 🤖 Here are some questions you might ask me:

- 💡 What are the top fashion trends this summer?
- 💡 Can you suggest an outfit for a summer wedding?
- 💡 What are some must-have accessories for winter season?
- 💡 What type of shoes should I wear with a cocktail dress?
- 💡 What's the best look for a professional photo shoot?

Clear Chat

Your message

Fork  



Install all the dependencies

01

Create virtual environment and activate it

```
python -m venv .venv
```

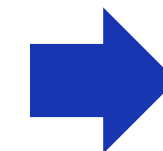
```
.venv\Scripts\activate.bat
```

01

Install all dependencies using pip command in the virtual environment

```
pip install -r requirements.txt
```

requirements.txt



```
chromadb==0.5.3  
pysqlite3-binary  
protobuf==3.20.*  
streamlit==1.36.0  
pypdf==4.2.0  
langchain==0.2.5  
langchain-community==0.2.5  
langchain-huggingface==0.0.3  
python-dotenv
```

Vector Database Implementation



Vector Database Implementation

01

Importing dependencies and API key

```
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import Chroma
import os
from dotenv import load_dotenv
from collections import OrderedDict

# Load environment variables from .env file
load_dotenv()

os.environ["HUGGINGFACEHUB_API_TOKEN"] = os.getenv("HUGGINGFACEHUB_API_TOKEN")
```

Sample Dataset

Here is my sample dataset

```
{
  "Item": "Twilight Velvet Blazer",
  "Type": "Jacket",
  "Style": "Formal",
  "Season": "Fall/Winter",
  "Material": "Velvet",
  "Color": "Deep Navy",
  "Sizes Available": ["S", "M", "L", "XL"],
  "Features": "Tailored fit, satin lapels, single-button closure, with decorative cuff buttons and inner pocket."
},
{
  "Item": "Coastal Linen Shorts",
  "Type": "Pants",
  "Style": "Casual",
  "Season": "Summer",
  "Material": "Linen",
  "Color": "Beige",
  "Sizes Available": ["S", "M", "L", "XL"],
  "Features": "Relaxed fit, elastic waistband, drawstring, with rolled cuffs and deep pockets."
},
```

Vector Database Implementation

02

PDF Data Loading

```
# Load the PDF
loader = PyPDFLoader("fashion_data.pdf") # Provide your PDF path here
documents = loader.load()
```

03

Text Splitting

```
# Split the text
text_splitter = RecursiveCharacterTextSplitter(chunk_size=2000, chunk_overlap=300)
texts = text_splitter.split_documents(documents)
```

Vector Database Implementation

04

Text Embedding

```
# Initialize the embedding model
embedding_model = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

# Convert texts to embeddings
try:
    embeddings = embedding_model.embed_documents([doc.page_content for doc in texts])
    print("Vector Embeddings created successfully")
except Exception as e:
    print(f"Error creating vector embeddings: {e}")
```

05

Storing in Vector Database

```
# Initialize Chroma vector store
vector_store = Chroma(embedding_function=embedding_model, persist_directory="data")

# Add documents to the vector store
vector_store.add_documents(documents=texts)
```

Vector Database Implementation

06

Validate the setup

```
# Validate the setup
try:
    # Test query to validate data retrieval
    test_query = "What are some popular items for winter?"
    results = vector_store.search(query=test_query, search_type='similarity')

    # Deduplicate results
    unique_results = OrderedDict()
    for doc in results:
        if doc.page_content not in unique_results:
            unique_results[doc.page_content] = doc

    # Convert unique results to a list and limit to top 3
    final_results = list(unique_results.values())[:3]
    print(f"Unique query results: {final_results}")
except Exception as e:
    print(f"Error during test query: {e}")
```

Vector Database Implementation

07

Result of the test query

Vector Embeddings created successfully

```
Unique query results: [Document(metadata={'page': 14, 'source': 'fashion_data.pdf'}, page_content='"Style": "Formal", \n      "Season": "All -Season", \n      "Material": "Stainless steel", \n      "Color": "Silver", \n      "Sizes Available": ["One Size"], \n      "Features": "Water -resistant, nautical inspired, with compass bezel and luminous hands." \n    }, \n    { \n      "Item": "Crimson Velvet Heels", \n      "Type": "Accessory", \n      "Style": "Formal", \n      "Season": "All -Season", \n      "Material": "Velvet", \n      "Color": "Deep Red", \n      "Sizes Available": ["6", "7", "8", "9"], \n      "Features": "Stiletto heel, pointed toe, with cushioned insole and ankle strap." \n    }, \n    { \n      "Item": "Frost Beanie", \n      "Type": "Accessory", \n      "Style": "Casual", \n      "Season": "Winter", \n      "Material": "Wool", \n      "Color": "Snow White", \n      "Sizes Available": ["One Size"], \n      "Features": "Knit pattern, warm fabric, with pom -pom top and fold -over cuff." \n    }, \n    { \n      "Item": "Trailblazer Hiking Boots",')]
```

Application Development



Application Development

01

Initialization of Environment and Dependencies

```
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import Chroma
import os
from dotenv import load_dotenv
from collections import OrderedDict

# Load environment variables from .env file
load_dotenv()

os.environ["HUGGINGFACEHUB_API_TOKEN"] = os.getenv("HUGGINGFACEHUB_API_TOKEN")
```


Application Development

02

Setup of the Language Model and Vector Store

```
21
22 # Load the vector store from disk
23 embedding_model = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")
24 vector_store = Chroma(embedding_function=embedding_model, persist_directory=data_directory)
25
```

03

Preparation of the Retrieval System

```
26 # Initialize the Hugging Face Hub LLM
27 hf_hub_llm = HuggingFaceHub(
28     repo_id="meta-llama/Meta-Llama-3-8B-Instruct",
29     # repo_id="mistralai/Mistral-7B-Instruct-v0.3",
30     model_kwargs={"temperature": 1, "max_new_tokens":1024},
31 )
32
```

Application Development

04

Configure the retriever to fetch top 3 results

```
69 rag_chain = RetrievalQA.from_chain_type(  
70     llm=hf_hub_llm,  
71     chain_type="stuff",  
72     retriever=vector_store.as_retriever(top_k=3), # retriever is set to fetch top 3 results  
73     chain_type_kwargs={"prompt": custom_prompt})  
74
```

Prompt Engineering

05

Integration of Custom Prompt Template

```
33 prompt_template = """
34 As a highly knowledgeable fashion assistant, your role is to accurately interpret fashion queries and
35 provide responses using our specialized fashion database. Follow these directives to ensure optimal user interactions:
36 1. Precision in Answers: Respond solely with information directly relevant to the user's query from our fashion database.
37    Refrain from making assumptions or adding extraneous details.
38 2. Topic Relevance: Limit your expertise to specific fashion-related areas:
39    - Fashion Trends
40    - Personal Styling Advice
41    - Seasonal Wardrobe Selections
42    - Accessory Recommendations
43 3. Handling Off-topic Queries: For questions unrelated to fashion (e.g., general knowledge questions like "Why is the sky blue?"),
44    politely inform the user that the query is outside the chatbot's scope and suggest redirecting to fashion-related inquiries.
45 4. Promoting Fashion Awareness: Craft responses that emphasize good fashion sense, aligning with the latest trends and
46    personalized style recommendations.
47 5. Contextual Accuracy: Ensure responses are directly related to the fashion query, utilizing only pertinent
48    information from our database.
49 6. Relevance Check: If a query does not align with our fashion database, guide the user to refine their
50    question or politely decline to provide an answer.
51 7. Avoiding Duplication: Ensure no response is repeated within the same interaction, maintaining uniqueness and
52    relevance to each user query.
53 8. Streamlined Communication: Eliminate any unnecessary comments or closing remarks from responses. Focus on
54    delivering clear, concise, and direct answers.
55 9. Avoid Non-essential Sign-offs: Do not include any sign-offs like "Best regards" or "FashionBot" in responses.
56 10. One-time Use Phrases: Avoid using the same phrases multiple times within the same response. Each
57    sentence should be unique and contribute to the overall message without redundancy.
58
59 Fashion Query:
60 {context}
61
62 Question: {question}
63
64 Answer:
65 """
66
67 custom_prompt = PromptTemplate(template=prompt_template, input_variables=["context", "question"])
68
```

Application Development

06

Response Generation Mechanism

```
68
69 ✓ rag_chain = RetrievalQA.from_chain_type(
70     llm=hf_hub_llm,
71     chain_type="stuff",
72     retriever=vector_store.as_retriever(top_k=3), # retriever is set to fetch top 3 results
73     chain_type_kwargs={"prompt": custom_prompt})
74
75 ✓ def get_response(question):
76     result = rag_chain({"query": question})
77     response_text = result["result"]
78     answer_start = response_text.find("Answer:") + len("Answer:")
79     answer = response_text[answer_start:].strip()
80     return answer
```

Application Development

07

Streamlit Interface Configuration - Customize the Streamlit layout to remove unnecessary whitespace and add aesthetic elements like headers and sidebar messages

```
82 # Streamlit app
83 # Remove whitespace from the top of the page and sidebar
84 st.markdown(
85     """
86     <style>
87     .appview-container .main .block-container {{
88         padding-top: {padding_top}rem;
89         padding-bottom: {padding_bottom}rem;
90     }}
91
92     </style>""".format(
93         padding_top=1, padding_bottom=1
94     ),
95     unsafe_allow_html=True,
96 )
97
98 # st.header("### Discover the AI styling recommendations :dress:", divider='grey')
99 st.markdown("""
100     <h3 style='text-align: left; color: black; padding-top: 35px; border-bottom: 3px solid red;'>
101         Discover the AI Styling Recommendations 👗👠
102     </h3>""", unsafe_allow_html=True)
103
```

Application Development

08

Create interactive elements such as chat input and buttons to clear the chat history, enhancing user interaction.

```
104
105 side_bar_message = """
106 Hi! 🙋 I'm here to help you with your fashion choices. What would you like to know or explore?
107 \nHere are some areas you might be interested in:
108 1. **Fashion Trends** 👕 👖
109 2. **Personal Styling Advice** 👠 👢
110 3. **Seasonal Wardrobe Selections** 🌞
111 4. **Accessory Recommendations** 💍
112
113 Feel free to ask me anything about fashion!
114 """
115
116 with st.sidebar:
117     st.title('🤖 FashionBot: Your AI Style Companion')
118     st.markdown(side_bar_message)
119
120 initial_message = """
121 Hi there! I'm your FashionBot 🤖
122 Here are some questions you might ask me:\n
123 🧡 What are the top fashion trends this summer?\n
124 🧡 Can you suggest an outfit for a summer wedding?\n
125 🧡 What are some must-have accessories for winter season?\n
126 🧡 What type of shoes should I wear with a cocktail dress?\n
127 🧡 What's the best look for a professional photo shoot?
128 """
```

Application Development

09

User Interaction Flow

```
130 # Store LLM generated responses
131 if "messages" not in st.session_state.keys():
132     st.session_state.messages = [{"role": "assistant", "content": initial_message}]
133
134 # Display or clear chat messages
135 for message in st.session_state.messages:
136     with st.chat_message(message["role"]):
137         st.markdown(message["content"])
138
139 def clear_chat_history():
140     st.session_state.messages = [{"role": "assistant", "content": initial_message}]
141 st.button('Clear Chat', on_click=clear_chat_history)
142
143 # User-provided prompt
144 if prompt := st.chat_input():
145     st.session_state.messages.append({"role": "user", "content": prompt})
146     with st.chat_message("user"):
147         st.markdown(prompt)
148
149 # Generate a new response if last message is not from assistant
150 if st.session_state.messages[-1]["role"] != "assistant":
151     with st.chat_message("assistant"):
152         with st.spinner("Hold on, I'm fetching the latest fashion advice for you..."):
153             response = get_response(prompt)
154             placeholder = st.empty()
155             full_response = response # Directly use the response
156             placeholder.markdown(full_response)
157     message = {"role": "assistant", "content": full_response}
158     st.session_state.messages.append(message)
```


**Lets run our
Chatbot
application!**



Run the application

01

Run vector_database implementation file using below command:
python vector_db.py

```
chatbot>python vector_embeddings.py
```

02

Run our main application using below command
streamlit run filename.py

```
chatbot>streamlit run FashionBot.py
```

Evaluation and Testing

01

Simple Query:

"What are the top fashion trends this summer?"



What are the top fashion trends this summer?



This summer, classic blue denim pants with a relaxed fit, adjustable straps, and front bib pocket are a must-have for casual, effortless style. Pair them with the vibrant Fiesta Maxi Skirt in multicolor print for a statement look. Don't forget to add a touch of rustic charm with the Gladiator Sandals in rustic brown leather. These fashion trends are sure to keep you looking fresh and stylish throughout the summer season.

Evaluation and Testing

02

Simple Query:

"What type of shoes should I wear with a cocktail dress?"



What type of shoes should I wear with a cocktail dress?



Based on your query, I recommend pairing a cocktail dress with a pair of Crimson Velvet Heels. The stiletto heel and pointed toe will complement the dress's style, while the cushioned insole and ankle strap will provide comfort and support. The Deep Red color will also match the dress's color, creating a cohesive and stylish look.

Evaluation and Testing

03

Complex/Nuanced Query:

"i want to visit a museum in afternoon, and then have lunch party at a place , please suggest me some clothing outfit that will suit both the occasion"



i want to visit a museum in afternoon, and then have lunch party at a place , please suggest me some clothing outfit that will suit both the occasion



For a museum visit in the afternoon, I recommend wearing something comfortable and stylish. Considering the "Heritage Wool Sweater" in Burgundy, it would be a great choice. The cable knit and crew neck will add a touch of sophistication, while the wool material will keep you warm and cozy. Pair it with a pair of dark jeans or trousers and loafers or boots for a chic and put-together look.

For the lunch party, you can easily dress up the outfit by adding a blazer or a cardigan. The "Nomad Leather Backpack" in Royal Blue would be a great accessory to add a pop of color and a casual touch. You can also add a statement piece of jewelry, such as a bold necklace or earrings, to elevate the outfit.

Remember to dress according to the dress code specified by the party host, if any. Overall, this outfit combination will help you look stylish and put-together for both occasions.

Evaluation and Testing

04

Complex/Nuanced Query:

"What are some ways to style a white t-shirt for different occasions?"



What are some ways to style a white t-shirt for different occasions



For a casual look, you can pair a white t-shirt with distressed denim jeans and sneakers for a relaxed, everyday outfit. For a more formal occasion, you can dress up a white t-shirt with a blazer and trousers for a stylish and sophisticated look. You can also add a pop of color with a statement necklace or a bold scarf to elevate the outfit. For a more bohemian look, you can layer a white t-shirt under a flowy cardigan or a kimono for a cozy and eclectic outfit. Ultimately, the key to styling a white t-shirt is to balance it with other pieces that complement its simplicity and versatility.

Evaluation and Testing

05

Scenario-Based Query:

"I have a travel plan this weekend that includes visiting historical sites, shopping, and attending a concert. Could you please suggest outfits and accessories to match styles for all these occasions?"



"I have a travel plan this weekend that includes visiting historical sites, shopping, and attending a concert. Could you please suggest outfits and accessories to match styles for all these occasions?"



Based on your travel plan, I recommend the following outfits and accessories:

For historical sites:

- Wear the Safari Linen Vest in Khaki, which is lightweight and suitable for warm weather. Pair it with a white or light-colored shirt and comfortable pants or a flowy skirt.

For shopping:

- The Celestial Silk Nightgown in Sky Blue is a versatile and elegant option for a casual shopping trip. You can wear it with sandals or sneakers and add a statement necklace for a stylish touch.

For the concert:

- The clutch in Royal Purple is a great accessory for a night out. You can wear it with a simple white or black top and high-waisted pants or a flowy skirt. Add some statement earrings and a bold lip color to complete the look.

Remember to dress according to the venue's dress code and the weather forecast. Have a great trip!

Evaluation and Testing

06

Scenario-Based Query:

"I'm going on a first date and want to make a good impression. What should I wear and what are some styling tips?"



I'm going on a first date and want to make a good impression. What should I wear and what are some styling tips?



For a first date, you want to make a good impression, and fashion plays a significant role in that. Considering the items in our database, I recommend the "Celestial Silk Nightgown" in Sky Blue. It's a floor-length dress with adjustable straps, lace trim, and a soft, flowing silhouette. This will make you look elegant and sophisticated.

Styling tips:

- Pair the dress with some statement jewelry, like a bold necklace or earrings, to add a touch of glamour.
- Keep your hair simple and sleek, perhaps in a low bun or a ponytail.
- Opt for a pair of heeled sandals or pumps to elongate your legs and add height.
- Keep your makeup natural and subtle, focusing on enhancing your natural features.

Remember, confidence is key, so own that dress and have fun on your date!

Evaluation and Testing

07

Irrelevant/Out-of-Scope Queries:
"Do you have any movie collection?"



Do you have any movie collection?



"Sorry, I'm a fashion assistant, and I don't have information on movie collections. I can help you with fashion-related queries, though! What's your fashion concern or question?"

Evaluation and Testing

08

Irrelevant/Out-of-Scope Queries:
"Why is the sky blue?"



Why is the sky blue?"



I'm happy to help you with your fashion-related questions! However, the query "Why is the sky blue?" is outside the scope of our fashion database. If you'd like to know more about the latest fashion trends or receive personalized styling advice, I'd be delighted to assist you. Please feel free to rephrase your question or ask something related to fashion.

Evaluation and Testing

09

Misspelled Query:
"men cliothin recommenations?"



• "men cliothin recommenations?"



As a fashion assistant, I'd be happy to help you with some men's clothing recommendations! Based on your query, I'll provide you with some suggestions that cater to various styles and preferences.

For a casual look, I recommend the "Silhouette Slim Jeans" in Dark Wash. They're a versatile and comfortable option that can be paired with a variety of tops and shoes.

If you're looking for something more formal, the "Executive Winter Coat" in Charcoal Grey would be an excellent choice. Its slim fit and double-breasted design make it perfect for a professional setting.

For a sporty vibe, the "Alpine Trek Pants" in Black are a great option. They're waterproof, breathable, and feature reinforced knees for added durability.

Lastly, if you're looking for a stylish accessory, the "Leather Ankle Boots" in Brown would be a great addition to your wardrobe. They're waterproof, ankle-supportive, and feature a rugged sole and lace-up closure.

I hope these recommendations help you find the perfect outfit for your needs! If you have any further questions or would like more suggestions, feel free to ask!

LLM Model Performance Comparison



LLM Model Performance Comparison

01

compares the performance of the two Hugging Face models

Model Name	Response Quality	Scope Relevance	Notes
Meta-Llama-3-8B-Instruct	High	High	Consistently relevant and accurate.
Mistral-7B-Instruct-v0.3	Moderate	Low	Often vague and out of scope.

Thank you!

