

Mynta Fashion Product Search AI Assistant

The image shows a screenshot of the Mynta Fashion Product Assistant web application. On the left, there is a large promotional image of a woman holding several colorful shopping bags, with the Mynta logo overlaid. The main interface has a search bar at the top with the placeholder "What are you looking for today?". Below the search bar is a text input field containing the query "Red ethnic kurta under 2000 with good rating". To the right of the search bar are two buttons: "Search" and "Clear". The search results are displayed in a table format. The first result is for an "Anouk Women Red Floral Print A-Line Kurta" by Anouk, priced at ₹1699.0. It has a rating of 4.08 (N/A ratings). The second result is for a "Libas Women Red & Gold-Toned Ethnic Motifs Printed Kurta".

Name	Anouk Women Red Floral Print A-Line Kurta
Brand	Anouk
Price	₹1699.0
Rating	★ 4.08 (N/A ratings)

Name	Libas Women Red & Gold-Toned Ethnic Motifs Printed Kurta
------	--

Prepared by

Kasi Viswanadh Maddala

Problem Statement

In today's fast-paced digital era, online fashion platforms host **millions of product listings** across categories, brands, sizes, colors, and styles. While traditional search filters (like drop-downs or checkboxes) work to some extent, they often fall short in understanding **natural language queries** and **personalized preferences** from users. Customers are forced to manually refine filters repeatedly to find what they truly want — leading to frustration, abandonment, and poor user experience.

Despite powerful search engines, **modern shoppers demand a more intuitive & intelligent interface** — one that understands their needs just like a **human shopping assistant** would.

Existing Challenges

- How can we enable users to **search fashion products** using **conversational, natural language** like:
 - "Show me blue cotton kurtas under ₹1500 with 4+ rating"
 - "Looking for a festive red anarkali with dupatta and good reviews"
- How can we intelligently **extract relevant filters** from such queries (e.g., color, category, price, rating)?
- Can we present results in a **visually rich, personalized, and interactive format** to improve engagement and user satisfaction?
- How can we further enhance the system with **conversational memory, data analysis, and database interaction** to support agents like:
 - Conversational LLM assistant
 - SQL agent for backend querying
 - CSV/pandas agent for analytics?

Project Goals

To design and build an AI-powered Conversational Fashion Assistant for Myntra that can:

- Accept free-form natural language queries from users.
- Extract structured filters using LLM (e.g., price range, color, occasion).
- Retrieve top matching products using semantic search and similarity scoring.
- Display results in a visually appealing grid format, with product image, price, brand, and rating.
- Enhance customer interaction via a rich, real-time chatbot interface.
- Support conversational agents for SQL, data analytics, and real-time decision support

Project Overview

Myntra Conversational Fashion Search AI Assistant is an AI-powered Streamlit web application that allows users to search fashion products using natural language queries like:

"Show me black cotton kurtas under ₹1000 with 4+ rating"

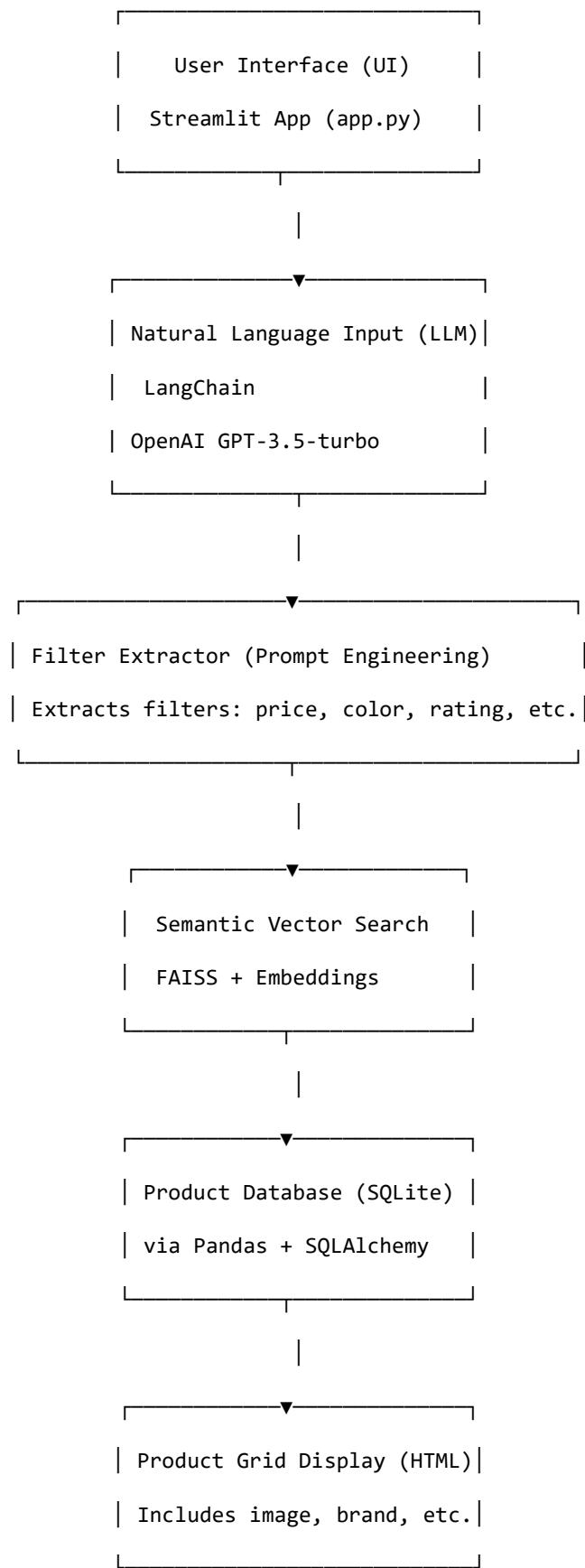
The assistant understands user intent using **LLM-based prompt engineering**, extracts relevant filters, retrieves matching products from a vectorized search index, and displays the top results in a visually rich, interactive UI.

The system is designed to simulate a **smart, real-time personal shopping assistant** using advanced NLP, vector search, and LLM agents.

Key Features of Myntra Fashion Product AI Assistant

Feature	Description
Natural Language Search	Users can search with phrases like "Blue denim jacket under 2000"
LLM-Powered Filter Extraction	Extract filters like price, rating, color, category using LangChain & OpenAI
Product Retrieval Engine	Retrieves top matching products using FAISS + cosine similarity
Rich Visual Display	Products displayed with images, names, prices, brands, and ratings
Conversational Chat Interface	Built-in chat interface for user interaction
Data Analytics Agent	Pandas-based agent to analyze product data
SQL Agent	LangChain SQL agent to query product DB
Beautiful UI with CSS	Styled interface with custom buttons, dark mode, and grid layout
Clear Query Button	Instantly resets the query and clears results

Tech Stack & Architecture



Why LangChain is an Ideal Framework?

Feature	Why It's Ideal for Fashion Assistant
Prompt Management	Templates for extracting filters, summarizing results, handling follow-ups
LLM Chains	Easily connect user inputs → prompts → LLM → response
Tool Integration	Add custom tools (like Pandas or SQL agents) for analytics
Memory & Context	Allow multi-turn conversation for refining searches or asking follow-ups
Agents	You can create an autonomous fashion analyst or recommender that picks tools
Streaming & Async	Support smooth, responsive user experiences
Vector Search Integration	Easily plug in FAISS or Chroma to enable semantic search over product descriptions
Built-in Observability	Debug, trace, and optimize your query chains

Real Use Cases Made Easy with LangChain

Capability	How LangChain Helps
Convert Natural Language to Filters	Use LLMChain or PromptTemplate + ChatOpenAI to extract filters (price, rating, brand, gender)
Semantic Product Search	Use FAISS + OpenAIEmbeddings to semantically retrieve relevant products
Data Analytics Agent	Use PandasAgent or CSVAgent to let LLM run analytics on the dataset
Conversational Agent	Use ConversationChain or ChatPromptTemplate + Memory for contextual chat
SQL Agent (Advanced)	LangChain's SQLDatabaseChain or SQLAgent lets the LLM run SQL queries directly
Tool Orchestration	Agents can decide when to search, summarize, analyze, or clarify
Summarization / Ranking	Use a summarization chain to re-rank or describe the top 3 products

User Queries & Responses using Conversational Agent

User Query1:

```
user_query1 = "Show me ethnic motif kurtas under ₹3000."  
fashion_search_ai(user_query1, top_k=2)  
{'product_type': 'kurta', 'max_price': 3000}
```



Name	AHIKA Women Black & Green Printed Straight Kurta
Brand	AHIKA
Price	₹1350.0
Rating	⭐ 3.98 (N/A ratings)



Name	Anouk Women Yellow & White Printed Kurta with Palazzos
Brand	Anouk
Price	₹1999.0
Rating	⭐ 4.25 (N/A ratings)

User Query2:

```
user_query2 = "Sleeveless kurtas in orange color for casual wear"  
fashion_search_ai(user_query2, top_k=1)  
{'product_type': 'kurta', 'colour': 'orange'}
```



Name	InWeave Women Orange Solid Kurta with Palazzos & Floral Print Dupatta
Brand	InWeave
Price	₹5899.0
Rating	⭐ 4.12 (N/A ratings)

User Query3:

```
user_query3 = "I'm searching for sarees in elegant colors like blue"
fashion_search_ai(user_query3, top_k=1)

{'product_type': 'saree', 'colour': 'blue'}
```



Name	Zainab chottani Blue Beads and Stones Pure Georgette Block Print Saree
Brand	Zainab chottani
Price	₹2999.0
Rating	⭐ nan (N/A ratings)

User Query4:

```
user_query4 = "I'm searching for blue jeans with top ratings"
fashion_search_ai(user_query4, top_k=2)

{'product_type': 'jeans', 'colour': 'blue', 'min_rating': 5}
```



Name	Levis Women Blue 714 Straight Fit High-Rise Light Fade Stretchable Jeans
Brand	Levis
Price	₹3699.0
Rating	⭐ 5.0 (N/A ratings)



Name	People Women Blue Skinny Fit Jeans
Brand	People
Price	₹1299.0
Rating	⭐ 5.0 (N/A ratings)

User Query5:

```
user_query5 = "I'm looking for Yoga Trousers under price 5000 with Green color"  
fashion_search_ai(user_query5, top_k=1)
```

```
{'product_type': 'Yoga Trousers', 'colour': 'Green', 'max_price': 5000}
```



Name	EVERDION Women Green Flared High-Rise Yoga Trousers
Brand	EVERDION
Price	₹1399.0
Rating	⭐ 4.29 (N/A ratings)

SQL Agent Query Example:

```
class SQLAgent:
    def __init__(self, db_path=SQLITE_DB_PATH):
        self.db = SQLDatabase.from_uri(f"sqlite:///{{db_path}}")
        self.llm = ChatOpenAI(
            temperature=0,
            model=LLM_MODEL,
            api_key=OPENAI_API_KEY
        )
        toolkit = SQLDatabaseToolkit(db=self.db, llm=self.llm)
        self.agent = create_sql_agent(llm=self.llm, toolkit=toolkit, verbose=True)

    def run_query(self, query: str):
        # Run the query through SQLDatabase directly for structured output
        result = self.db.run(query)
        return result

if __name__ == "__main__":
    sql_agent = SQLAgent()

    # Agent natural language interface
    nl_response = sql_agent.agent.run("List the top 3 products with the highest average rating")
    print("\n Agent Response:")
    print(format_agent_response(nl_response))

y Shape ID': '333,424', 'Body or Garment Size': 'Garment Measurements in', 'Bo Anubhutee Women Navy Blue Ethnic Motifs Embroidered Thread Work Kurta with Trousers & With Dupatta K
I need to query the top 3 products with the highest average rating from the products table.
Action: sql_db_query
I now know the final answer.M products ORDER BY avg_rating DESC LIMIT 3[('Anouk Women Black & White Bandhani Printed Halter Neck Kurta', 5.0), ('Sangria Women Black Pure Cotton Kurta with Trousers', 5.0), ('Varanga Women Sea Green & White Bandhani Printed Georgette Kurta', 5.0)]  
Final Answer: The top 3 products with the highest average rating are 'Anouk Women Black & White Bandhani Printed Halter Neck Kurta', 'Sangria Women Black Pure Cotton Kurta with Trousers', and 'Varanga Women Sea Green & White Bandhani Printed Georgette Kurta'.  
> Finished chain.  
Agent Response:  
## 🌐 Results  
- **1. Anouk Women Black & White Bandhani Printed Halter Neck Kurta**  
- **2. Sangria Women Black Pure Cotton Kurta with Trousers**  
- **3. Varanga Women Sea Green & White Bandhani Printed Georgette Kurta**
```

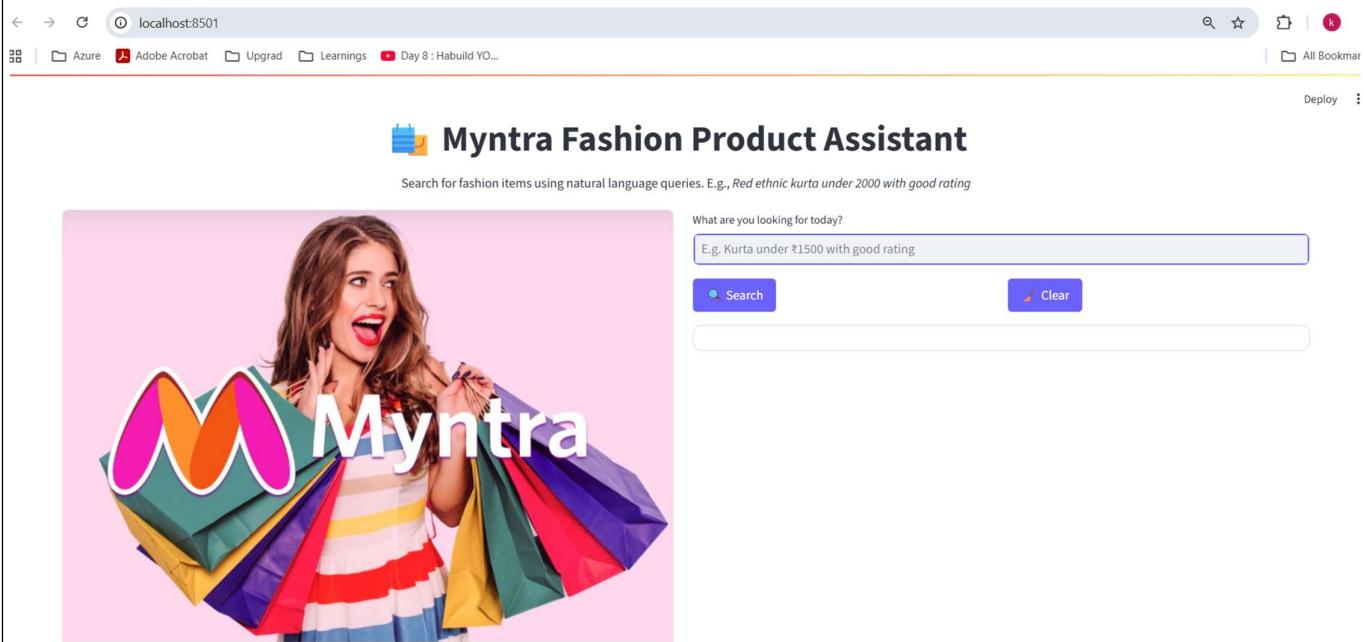
Data Analytics Agent Query Example

```
print("\n Agent Response:")
print(format_agent_response(nl_response))

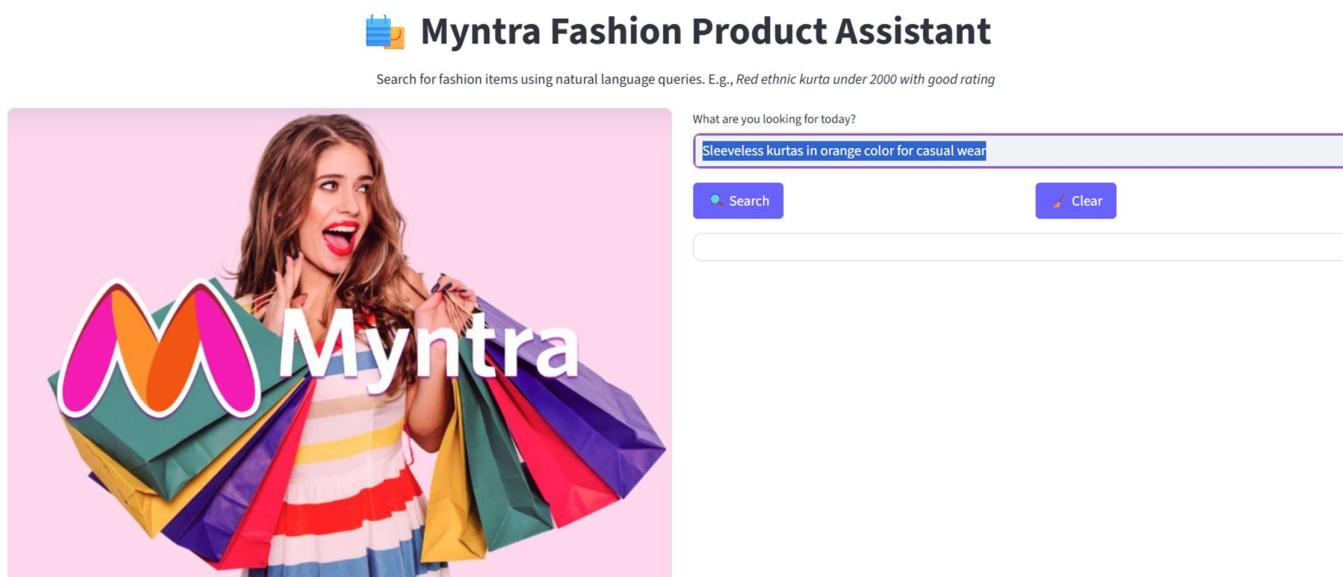
C:\Users\Admin\anaconda3\Lib\site-packages\langchain_experimental\agents\agent_toolkits\pandas\base.py:283: UserWarning: Received additional kwargs {'handle_parsing_errors': True} which are no longer supported.
  warnings.warn(
C:\Users\Admin\AppData\Local\Temp\ipykernel_14784\296216571.py:39: LangChainDeprecationWarning: The method `Chain.run` was deprecated in langchain 0.1.0 and will be removed in 1.0. Use :meth:`~invoke` instead.
  nl_response = pandas_agent.run("List the top 5 product categories by number of purchases")  
The top 5 product categories by number of purchases are:  
1. Dupatta: 1920 purchases  
2. Top: 1221 purchases  
3. Trousers: 1146 purchases  
4. Jeans: 986 purchases  
5. Saree: 903 purchases  
> Finished chain.  
Agent Response:  
## 🌐 Results  
- **1. Dupatta: 1920 purchases**  
- **2. Top: 1221 purchases**  
- **3. Trousers: 1146 purchases**  
- **4. Jeans: 986 purchases**  
- **5. Saree: 903 purchases**
```

Mynta Fashion Product Search AI Assistant Bot using Streamlit

Fashion Search AI Bot User Interface:



User asking Query in Chatbox:



Fashion Search AI Bot response for user query:

The screenshot shows a web browser window with the URL localhost:8501. The page title is "Myntra Fashion Product Assistant". A search bar contains the query "Sleeveless kurtas in orange color for casual wear". Below the search bar is a "Search" button and a "Clear" button. To the right of the search bar, there is a product card for an "InWeave Women Orange Solid Kurta with Palazzos & Floral Print Dupatta". The product card includes details: Name (InWeave Women Orange Solid Kurta with Palazzos & Floral Print Dupatta), Brand (InWeave), Price (₹5899.0), and Rating (4.12 (N/A ratings)).

User Query & Response using Conversational Agent:

The screenshot shows a web browser window with the URL localhost:8501. The page title is "Myntra Fashion Product Assistant". A search bar contains the query "I'm looking for Yoga Trousers under price 5000 with Green color". Below the search bar is a "Search" button and a "Clear" button. To the right of the search bar, there is a product card for "EVERDION Women Green Flared High-Rise Yoga Trousers". The product card includes details: Name (EVERDION Women Green Flared High-Rise Yoga Trousers), Brand (EVERDION), Price (₹1399.0), and Rating (4.29 (N/A ratings)).

Future Enhancements

Enhancement	Description
 Conversational Memory	Remember user preferences in ongoing sessions
 Personalized Recommendations	Suggest products based on user history or profile
 Image-Based Search	Let users upload images to find similar products
 Add-to-Cart Integration	Link results with Myntra cart / checkout
 Voice Input Support	Enable voice-based product search
 Advanced Analytics Dashboard	Use Streamlit Charts to analyze product trends
 Multi-language Support	Understand queries in Hindi, Tamil, etc.
 AI Chatbot Agent Integration	Fully conversational chatbot with memory, sentiment, etc.
 User Authentication	Secure login and personalized view
 Dark Mode Toggle	Improve user experience for night use

Deployment Guide

1. Download or clone the repo:

```
git clone https://github.com/mkviswanadh/Fashion\_Search\_AI.git
```

2. Navigate into the **fashion_product_search_bot** folder where fashion_search_bot.py, dependencies, configs, pipeline files reside.

3. Create & Activate a Python Virtual Environment • In PowerShell or CMD: cd path\to\chatbot_app
python -m venv venv .\venv\Scripts\activate

4. Install Dependencies with the command in the venv : pip install -r requirements.txt

5. Set Environment variables for OPENAI_API_KEY

6. Run Pre-process pipeline: python pipelines/preprocess_data.py

7. Create FAISS index : python pipelines/build_vector_store.py

8. Ingest the data into Sqllite table : python pipelines/init_db.py

9. Run the Streamlit App with below specified command:

```
streamlit run app.py --server.port 8501 --server.address 0.0.0.0
```

10. Check in Browser Open your browser: <http://127.0.0.1:8501> or <http://localhost:8501> You should see the chatbot interface or default home page rendered.