AI-Powered Retail Chatbot: Conversation, Recommendations, Analysis and Edge Support

Welcome to our innovative AI chatbot solution for retail. This cutting-edge system offers personalized recommendations, and customer support, all optimized for edge devices. Our approach combines advanced tool calling language models, synthetic data generation, and a user-friendly interface to create a dynamic, responsive retail assistant.

By Team Elif

Github Repo: https://github.com/payne19/Lab_Lab_AI_Chatbot

Data Generation and Storage

LLaMA 3.1 70B Model

We utilized the powerful LLaMA 3.1 70B model via an API to generate several hundred rows of synthetic data. This data was carefully curated and saved in a CSV file for further processing.

Dummy Data Creation

We generated dummy data using Kaggle as a source, which includes key retail information such as customer IDs, purchased items, quantities, purchase dates, and costs. This data will later be utilized to interact with the model.

PostgreSQL Storage

The dummy data was stored in a PostgreSQL database, ensuring efficient retrieval and management of the information for our AI chatbot tool calling system.

Data Preparation for Training

1

Processing Synthetic Data

We processed the synthetic data to create training examples for our model. This crucial step involved formatting the data in a specific structure to optimize learning.

2

Formatting Structure

The format used was row["text"] = row["query"] + row["tools"] + row["answers"] + tokenizer.eos_token

3

Component Breakdown

In this format, 'query' represents user questions, 'tools' are functions written for the model to call, and 'answers' are the expected outputs indicating which functions to use.

Model Training: Salesforce xLAM

xLAM Model Download

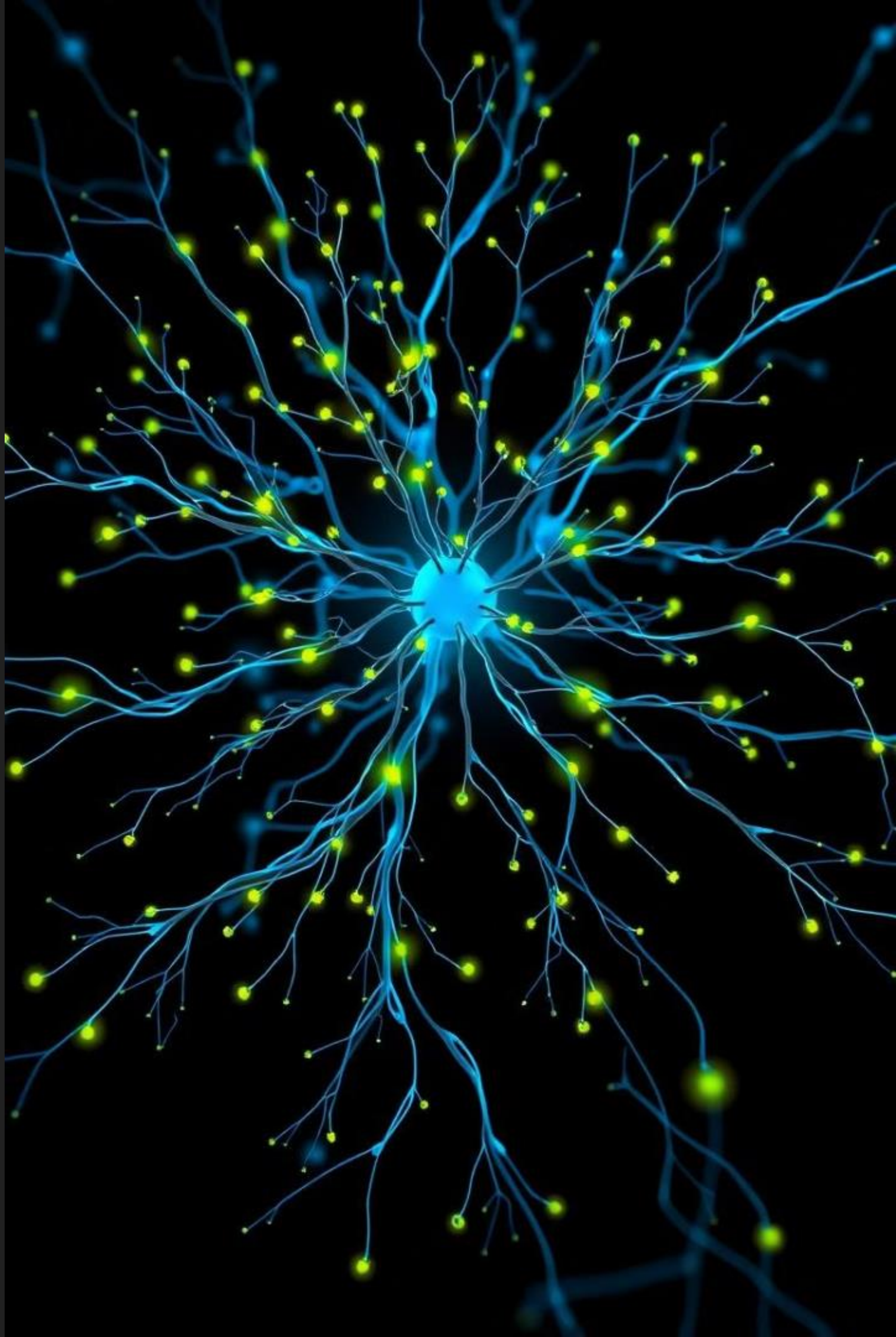
We began by downloading the Salesforce xLAM tool-calling model, known for its effectiveness in handling complex queries and calling write function name for it and it is only 1.35B parameters making it one of the best model to run it on edge devices.

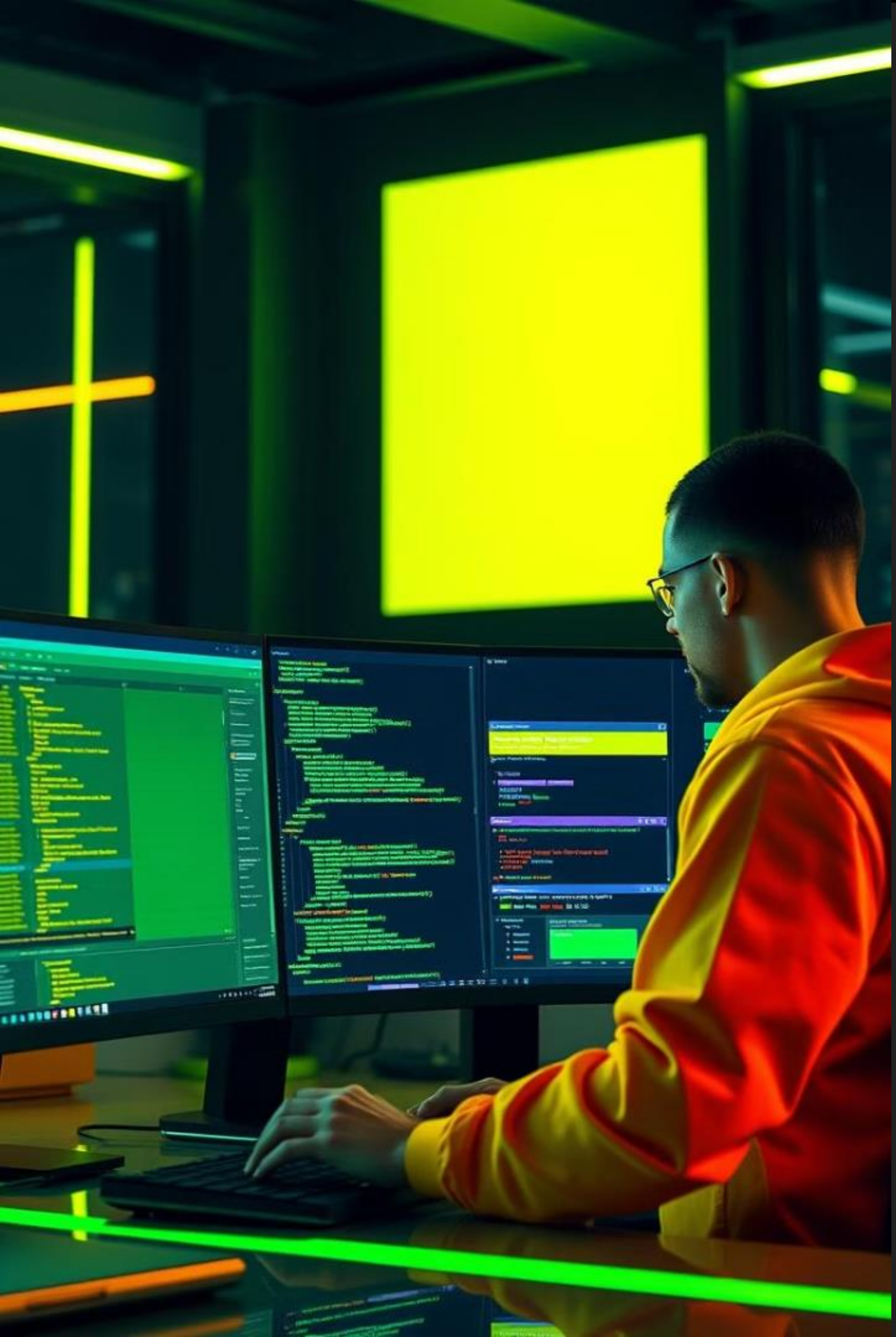
Fine-tuning Process

The xLAM model was then fine-tuned using our processed data, allowing it to learn from our specific database-focused dataset and improves performance in our use case.

Adapter Saving

After fine-tuning, we saved the model as an adapter on Hugging Face, making it easily accessible and deployable for our chatbot application.





Streamlit Application Development

1

UI Creation

We developed a user-friendly Streamlit application to serve as the interface for our AI chatbot. This ensures a smooth and intuitive experience for retail customers.

2

Model Integration

The fine-tuned xLAM adapter was integrated with the xLAM model using PEFT, combining the adapter with the xLAM model to create a powerful and responsive chatbot system tailored to our use case.

3

Prompt Design

We designed a prompt that includes function descriptions. When a user inputs a question, both the prompt and the question are sent to the model for processing.

Function Execution and Result Display

1 **Model Output**

The model processes the user's question and outputs a specific function name based on the query and available tools.

2 **Database Query**

The resulting function will be executed, containing SQL code used to run queries on the dummy data from Kaggle stored in our PostgreSQL database.

3 **Result Display**

The results of the database query are then displayed in the Streamlit UI, providing the user with relevant and accurate information.

Dynamic System Benefits

1 Real-time Responses

This approach allows you to create a dynamic system where user questions trigger specific functions, providing real-time, relevant responses to customer queries.

2 Database Interaction

The system interacts with a database of dummy data to provide relevant results, ensuring that responses are grounded in actual retail data.

3 No Hallucination

By using this method of tool calling, we rely on predefined functions instead of the AI model, ensuring that the AI chatbot operates without hallucinations and provides accurate and reliable information to retail customers.

Conclusion: Revolutionizing Retail Support

Personalized Recommendations

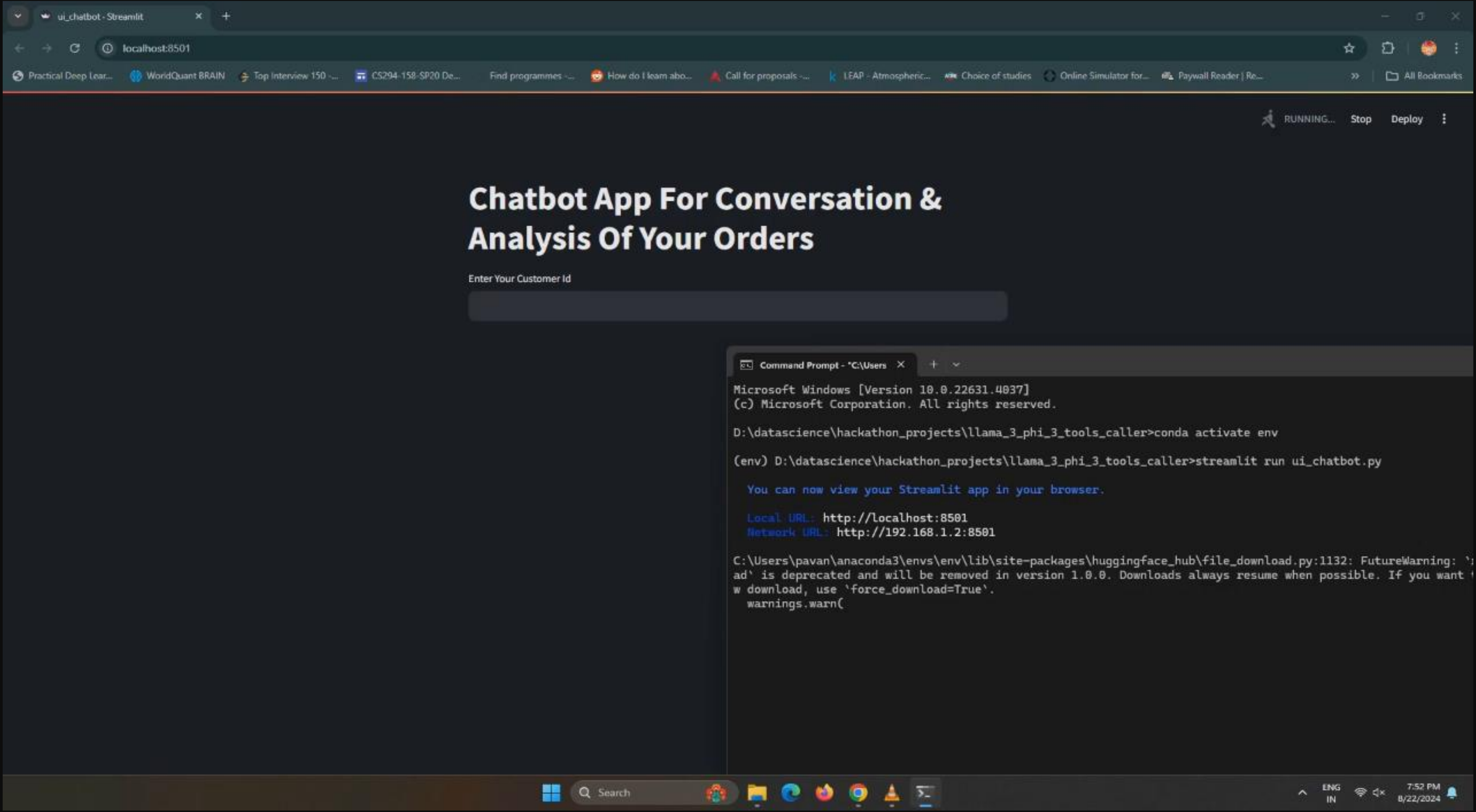
Our AI chatbot offers tailored product recommendations based on customer data and preferences.

Edge Device Support

Optimized for edge devices, ensuring seamless support across various platforms and locations.

Continuous Innovation

Our system paves the way for ongoing improvements in AI-powered retail support.



Chatbot App For Conversation & Analysis Of Your Orders

Enter Your Customer Id

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\datascience\hackathon_projects\llama_3_phi_3_tools_caller>conda activate env

(env) D:\datascience\hackathon_projects\llama_3_phi_3_tools_caller>streamlit run ui_chatbot.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.2:8501

C:\Users\pavan\anaconda3\envs\env\lib\site-packages\huggingface_hub\file_download.py:1132: FutureWarning: `r
ad` is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want t
w download, use `force_download=True`.
  warnings.warn(
```