Project Title	**AI E-commerce Assistant**
Technologies	Large Language Model
Domain	Generative AI
Project Difficulties level	Advance

Problem Statement:

The current online shopping landscape lacks the interactive and dynamic nature found in traditional brick-and-mortar stores, where customers can negotiate prices with shopkeepers and receive personalized recommendations based on their preferences. Existing e-commerce platforms predominantly rely on fixed prices and recommendation systems that often overlook specific product attributes and descriptions. To address these limitations, this project aims to develop an innovative e-commerce website that integrates artificial intelligence (AI) and machine learning (ML) techniques, fine-tuning of large language models with the custom dataset to introduce negotiation capabilities and enhance the recommendation system.

Objectives:

- **Introduce Bargaining and Negotiation:**
  - Develop a chatbot acting as a shopkeeper capable of negotiating prices with customers.
  - Implement natural language processing techniques and deep learning algorithms to enable effective communication between the chatbot and customers.
  - Ensure a seamless and user-friendly negotiation process to enhance the online shopping experience.
- **Content-Based Recommendation System:**
  - Create a recommendation system that suggests products based on product attributes and descriptions rather than relying solely on customer preferences and ratings.
  - Utilize natural language processing algorithms to analyze and understand product attributes, enhancing the accuracy of recommendations.
  - Ensure the recommendation system adapts to user behavior and provides relevant suggestions, improving customer satisfaction.

- **User-Friendly Interface:**
    - Design an intuitive and visually appealing user interface for the e-commerce website.
    - Ensure that the negotiation process is straightforward and easily understandable for users.
    - Implement features that enhance user engagement and satisfaction during the online shopping experience.
- **Testing and Evaluation:**
    - Conduct thorough testing of the chatbot's negotiation capabilities and the recommendation system to ensure reliability and accuracy.
    - Gather user feedback to continuously improve the system, addressing any issues or challenges identified during the testing phase.

Dataset:

The dataset sample for this project is given below. You can obtain more datasets like this by scrapping the e-commerce website.

Dataset sample

Project Evaluation metrics:

Code:

- You are supposed to write code in a modular fashion.
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system)
- You must maintain your code on GitHub.
- You must keep your GitHub repo public so that anyone can check your code.
- Proper readme file you must maintain for any project development.
- You should include basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards: https://www.python.org/dev/peps/pep-0008/

Database:

- You can use any Vector or NOSQL Database for this Project.

Cloud:

- You can use any cloud platform for this entire solution hosting like AWS, Azure, or GCP

API Details or User Interface:

- You must expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

Logging:

- Logging is a must for every action performed by your code. Use the Python logging library for this.

Ops Pipeline:

- You can try to use the AI ops pipeline for project delivery Ex. DVC, MLflow, Sage maker, Azure machine learning studio, Jenkins, Circle CI, Azure DevOps, TFX, Travis CI.

System Architecture:

Deployment:

- You can host your model in the cloud platform, edge devices, or maybe local, but with a proper justification of your system design.

Solutions Design:

- You must submit complete solution design strategies in HLD and LLD document.
- You must submit a system architecture design in your wireframe document and architecture document.

Latency for model response:

- You must measure the response time of your model for a particular input of a dataset.

Optimization of solutions:

- Try to optimize your solution on the code level, and architecture level and mention all of these things in your final submission.
- Mention your test cases for your project.

Submission requirements:

High-level Document:

You must create a high-level document design for your project. You can reference the HLD form below the link.

Sample link:

HLD Document Link

Low-level document:

You must create a Low-level document design for your project; you can refer to the LLD from the below link.

Sample link

LLD Document Link

Architecture: You must create an Architecture document design for your project; you can refer to the Architecture from the below link.

Sample link Architecture

sample link

Wireframe: You must create a Wireframe document design for your project; refer to the Wireframe from the below link.

Demo link

Wireframe Document Link

Project code:

You must submit your code GitHub repo in your dashboard when the final submission of your project.

Demo link

Project code sample link :

Detail project report: You must create a detailed project report and submit that document as per the given sample. Demo link DPR sample link

Project demo video:

You must record a project demo video for at least 5 Minutes and submit that link as per the given demo.

Demo link

Project sample link :

The project LinkedIn a post:

You must post your project details on LinkedIn and submit that post link in your dashboard in your respective field.

Demo link

Linkedin post sample link :