# Case Study 1- Development of an E-commerce Sales Chatbot

In the competitive sphere of e-commerce, deploying advanced, interactive, and user-centric sales chatbots is essential for elevating customer experience and streamlining business operations. This case study presents the challenge of developing a comprehensive sales chatbot tailored for an e-commerce platform specialising in a specific product category (such as electronics, books, or textiles). The aim is to facilitate customer interactions from product search to purchase.

**Objective:**
Design and implement a sales chatbot that enhances the shopping experience by enabling efficient search, exploration, and purchase processes on an e-commerce platform. The deliverables include:

- The chatbot interface and logic.
- A simulated e-commerce server that processes user queries from the chatbot and returns relevant product data. This server should handle a mock inventory via RESTful interactions *(The backend data can be a mock e-commerce inventory)*

**Requirements:**

**1. User Interface / Front End:**

- Develop a responsive user interface compatible with desktop, tablet, and mobile devices, utilizing modern JavaScript frameworks alongside HTML5 and CSS.
- Implement a login and authentication module to secure user sessions.
- Manage session continuity to maintain user state throughout interactions.
- Design a simple, intuitive chatbot interface with features such as conversation reset buttons and session tracking with timestamps.
- Ensure all chat interactions are stored effectively for later retrieval and analysis.

**2. Backend**

- Create an API-driven backend system using Python with frameworks like Flask or Django, capable of processing search queries and fetching relevant product data from a database.
- Populate a relational database management system (RDBMS) with at least 100 mock e-commerce product entries.

**3. Technical Documentation:**

- Document the entire process, including architecture , choice of tools/frameworks, and mock data creation should be documented
- Bonus if the candidate can include a section on potential challenges faced and how they were handled

**4. Code Quality and Best Practices:**

- The code should be clean, readable, and well-commented, adhering to industry standards for maintainability and scalability. Code should be implemented in python
- Ensure the codebase is modular and fault-tolerant, with clear separation of concerns and robust error handling.
- Provide rational for  the choice of frameworks, libraries, and design patterns used in the project.

**Evaluation Criteria:**

**1. UI User Experience:**

- Creativity applied while visualising the products that are fetched without compromising the UX
- Innovative ways considered and applied to enable to customer interact , filter and explore the products

**2. Technical Implementation:**

- Quality of code, including readability, structure, and adherence to best practices for both the client (chatbot and server module)
- Modular architecture and fault tolerance capabilities of the architecture

**3. Innovation and Problem-Solving:**
- Creativity in approach and solutions to challenges encountered during the project.
- Ability to leverage advanced UI design / web development techniques catering to seamless customer product search experience.

**4. Documentation and Presentation:**
- Clarity and completeness of technical documentation, including code comments, project setup, and execution instructions.
- Effectiveness in communicating the project's objectives, methodology, results, and learnings.

**Deliverables:**

- A GitHub repository containing all source code, complete with a detailed README.md outlining project setup, execution instructions, and a comprehensive project summary.
- A detailed project report showcasing the technology stack used, sample queries, and the results obtained.
- A presentation to the recruitment panel, detailing the project approach, technologies utilized, and key learnings.

## Case Study 2- Enhancing Job Search with Semantic Analysis Using LLM

Develop a system that collects job postings from open job sites of your choice (such as LinkedIn and Indeed, Naukri) within the Information Technology domain, and implements a semantic search functionality that outperforms the site's native search in fetching relevant job posts.

**Requirements:**

**1. Data Collection:**

- Collect at least 1000 job postings from open job sites focusing on the Information Technology domain.
- Ensure data includes job titles, descriptions, requirements, and any other relevant information.
- Provide a detailed description of the data collection process, including tools and methodologies used.

**2. Semantic Search Implementation:**

- Utilise large language models to implement semantic search capabilities.
- The system should understand the context and semantics of the job postings and the search queries to improve match accuracy.
- Compare the effectiveness of the developed semantic search against the native search functionality of the job sites for identical job IDs, showcasing improvements in relevance and accuracy.
- Built a simple front end application to showcase the search using libraries like framework/Gradio or web frameworks of your choice.

**3. Technical Documentation:**

- Document the entire process, including data collection, model training, semantic search implementation, and comparison methodology.
- Include a section on challenges faced and how they were overcome.

**4. Code Quality and Best Practices:**

- The code should be clean, readable, and well-commented, adhering to industry standards for maintainability and scalability. Code should be implemented in python
- Ensure the codebase is modular and fault-tolerant, with clear separation of concerns and robust error handling.
- Provide rational for the choice of frameworks, libraries, and design patterns used in the project.

**Evaluation Criteria:**

**1. Effectiveness of Semantic Search:**

- Accuracy and relevance of search results compared to native job site search functionalities. Apply or come up with metrics to quantitatively measure the quality of the solution. Eg: BLEU scores, BERT Scores, Precision/Recall/MRR and augmenting with Human Evaluation metrics.
- Innovative use of large language models in understanding and matching job descriptions and requirements with search queries.

**2. Technical Implementation:**

- Quality of code, including readability, structure, and adherence to best practices.
- Robustness of the data collection process and efficiency in handling and processing large datasets.
- Modular architecture and fault tolerance of the system.

**3. Innovation and Problem-Solving:**

- Creativity in approach and solutions to challenges encountered during the project.
- Ability to leverage advanced machine learning techniques and natural language processing tools to enhance search functionality.

**4. Documentation and Presentation:**

- Clarity and completeness of technical documentation, including code comments, project setup, and execution instructions.
- Effectiveness in communicating the project's objectives, methodology, results, and learnings.

**Deliverables:**

- GitHub repository containing all source code, with a README.md file outlining the project setup, execution instructions, references and a comprehensive summary of the project.
- Report detailing the project approach, technologies used, comparison results with native search functionalities, and insights gained.
- Good to have but not must – A short presentation summarizing the key aspects of the project, challenges faced, solutions implemented, and results achieved.
- Candidate will be providing a detailed walkthrough to the interviewers post completion of the case study.

Feel free to reach out @ uplyft.avd@gmail.com for any clarifications