Project Report: E-commerce Sales Chatbot

# 1. Introduction

This project involves the design and implementation of a sales chatbot that enhances the shopping experience by enabling efficient search, exploration, and purchase processes on an e-commerce platform. The project integrates a chatbot interface with a simulated e-commerce server that processes user queries and returns relevant product data.

# 2. Objectives

- Develop a chatbot capable of understanding user queries.

- Integrate with a simulated backend to retrieve relevant product information.

- Allow user registration, authentication, and chat history storage.

- Display product listings dynamically based on user input.

# 3. System Architecture

The system consists of the following components:  
- Frontend: React-based UI for login, registration, and chat interaction.  
- Backend: Flask server with RESTful APIs handling authentication, product retrieval, and chat history.  
- Database: SQLite for storing users, products, and chat messages.

# 4. Backend Implementation

The backend is built using Flask and SQLAlchemy. Key features include:

* - JWT-based authentication system.
* - User and product data models.
* - REST API routes for user registration, login, product retrieval, and chat history.

# 5. Frontend Implementation

The frontend is developed using React. Key components include:

* - App.jsx for routing between login, registration, and chatbot interface.
* - Chatbot.jsx for handling user messages, displaying responses, and showing products.
* - Login.jsx and Register.jsx for user authentication UI.

# 6. Chatbot Logic

The chatbot processes user input to identify keywords like 'phone', 'laptop', or 'accessory'. It responds with product listings matching the category or with predefined replies for greetings or invalid inputs.

# 7. Sample Data and Initialization

The `seed\_data.py` script initializes the database with a mock product inventory including phones, laptops, and accessories.

# 8. Security Features

* - Passwords are hashed using Werkzeug.
* - JWTs are used to protect routes and verify user identity.

# 9. Conclusion

This chatbot-based e-commerce interface offers a user-friendly shopping experience. It showcases integration between modern web technologies like React and Flask, highlighting the feasibility of intelligent and responsive user interfaces for retail.