ChatGPT

E-commerce Chatbot Project Report

Project Title: Full-Stack E-commerce Sales Chatbot

Internship: Uplyft Full Stack Intern Case Study – June 2025

---

1. Objective

To design and implement a full-stack e-commerce chatbot system that facilitates interactive product search and purchase functionality using modern frontend and backend technologies.

---

2. Project Architecture

Frontend: React.js - Pages: Chatbot Interface (ChatbotPage.js) - Axios used to make API requests - Responsive design using Tailwind CSS (or plain CSS)

Backend: Flask (Python) - RESTful API endpoints - SQLite as the relational database - CORS enabled

Database: SQLite3 - Contains mock inventory of 100+ products (name, category, price, description, image URL)

---

3. Features

- Chatbot-like interface for querying products
- User input handled through a React form
- Products matched from the backend and displayed dynamically
- Timestamps and user messages tracked
- Session-based display of past queries and responses

---

4. API Endpoints

- `POST /chat` — Receives user query, returns matched products
- `GET /products` — Returns all products (for admin/testing)

---

5. Technologies Used

- **Frontend:** React, Axios
- **Backend:** Flask, Flask-CORS, Flask-SQLAlchemy
- **Database:** SQLite3
- **Other Tools:** Faker (for mock data), Postman (for testing)

---

6. Sample Query Flow

User: "Mouse" **Bot:** "Found 1 product" - Wireless Mouse - ₹499 - Image + description rendered in card layout

---

7. Challenges Faced & Solutions

- **CORS issues:** Solved by enabling `flask-cors`
- **Data Seeding:** Used a script to seed the DB with CSV product data
- **Cross-platform layout:** React with flex/grid layout ensured responsiveness

---

8. Future Enhancements

- Integrate NLP for better query understanding
- Add authentication/login functionality
- Allow real-time order placement and cart management
- Admin dashboard to manage products

---

9. Setup Instructions

1. Run Flask backend:
    - `cd backend`
    - `python3 -m venv venv && source venv/bin/activate`
    - `pip install -r requirements.txt`
    - `python models.py` to seed database
    - `python app.py`
2. Run React frontend:
    - `cd frontend`
    - `npx create-react-app .`
    - Replace `src` folder with provided chatbot files
    - `npm install axios`
    - `npm start`

---

10. Conclusion

This project demonstrates a full-stack implementation of an e-commerce chatbot that allows users to interactively search for products. It emphasizes modular design, clean UI, and scalable architecture — fulfilling the core expectations of a full-stack intern challenge.