🧠 Uplyft Full Stack Intern Case Study — Documentation

---

1. Architecture Overview

The application is a full-stack web platform that enables users to search products using a natural language chatbot interface.

High-Level Architecture:

Frontend (React) → Flask API Backend → SQLite Database

Application Workflow:

1. Users log in via /login, receiving a JWT token
2. Authenticated users send queries via /chat
3. The server parses the query into filters (like price/category)
4. Matching products are fetched from the database and returned
5. The frontend renders product cards for the user

---

2. Frameworks & Tools Used

Frontend:

Tool	Purpose
React (Vite)	Component-based UI
Tailwind CSS	Utility-first styling
Axios	HTTP request handling
JWT (localStorage)	Auth token storage

Backend:

Tool	Purpose
Flask	REST API and routing
Flask-JWT-Extended	Authentication via JWT

Tool	Purpose
SQLAlchemy	ORM to interact with SQLite
Pandas	Data loading from CSV
SQLite	Lightweight embedded database

---

3. Mock Data Generation

Product data is generated using a script (generate_products.py) that outputs a CSV of fake product records.

Key Features:

- Random product names across categories: books, shoes, laptops, tshirts
- Random pricing (category-specific)
- Placeholder images (https://picsum.photos)
- Output saved to mock_data/products.csv
- The script creates the file and loads it into the database using db_init.py

---

4. Challenges & Solutions

✅ JWT Format Error

- **Problem:** JWT creation raised an error due to integer user ID
- **Solution:** Converted user.id to a string when creating token using create_access_token(identity=str(user.id))

✅ No Results from Chat Queries

- **Problem:** Price filters were too strict, and mock product prices were too high
- **Solution:** Modified data generation to assign more realistic pricing ranges

✅ Frontend Chat Not Sending Messages

- **Problem:** No response after clicking "Send" in the chat UI
- **Solution:** Identified browser ad blocker as the issue and whitelisted localhost

---

✅ Project Features

Feature	Status
User Authentication	✅
JWT Integration	✅
Natural Language Chat	✅
Product Filtering	✅
Mock Data + DB Setup	✅
Frontend Integration	✅

🚀 Optional Next Steps

- Add logout and token expiry handling
- Add persistent chat history (localStorage/sessionStorage)
- Improve query parsing with NLP or LLMs
- Deploy frontend on Netlify/Vercel and backend on Render/Fly.io
- Add product detail pages with routing