**Name: Prachi Shende**

**Roll no. : 381034**

**PRN : 22310945**

# Assignment 9: E-commerce Customer Service Chatbot

## Overview

This document summarizes an AI-powered chatbot designed for e-commerce platforms. Built using Python (Flask) with a web-based interface, it automates customer service tasks such as order tracking, product inquiries, returns, and shipping details. The chatbot enhances customer experience by providing instant, accurate, and context-aware responses while reducing human workload.

## Key Benefits

• 24/7 automated customer support  
• Faster response times and consistent information  
• Reduces workload for human agents  
• Scalable, cross-platform, and easy to use

## Core Components

Technology Stack: Python 3.7+, Flask, HTML5, CSS3, JavaScript, and JSON-based storage.

Architecture: Three-tier system consisting of Presentation Layer (Web UI), Application Layer (Flask API), and Data Layer (chatbot logic and mock data).

## Main Features

• Intent recognition for handling queries like order status, product info, returns, and shipping.  
• Context awareness for natural conversations.  
• Error handling for unrecognized inputs.  
• E-commerce integration with catalog and tracking.  
• User interface with responsive design, quick action buttons, and accessibility support.

## Installation & Usage

1. Install dependencies: pip install -r requirements.txt  
2. Run tests: python test\_chatbot.py  
3. Start server: python app.py  
4. Access via browser: http://localhost:5000  
Alternative: Run CLI version using python chatbot.py

## User Manual Highlights

The chatbot interface includes a chat header, message area, quick action buttons, and an input field. Common quick actions include Order Status, Returns, Shipping, and Products. Users can ask questions such as “Track my order 12345,” “Tell me about wireless headphones,” or “What’s your return policy?”

## API Endpoints

• POST /chat — Sends user message and returns chatbot response.  
• GET /health — Returns service health status.

## Customization

Developers can add new intents in chatbot.py, update sample data in sample\_data.json, and customize the web interface in index.html for branding and layout preferences.

## Troubleshooting

Includes fixes for port conflicts, missing dependencies, Flask startup issues, and Python version compatibility errors.

## Future Enhancements

Technical: Database integration, ML-based intent recognition, multi-language and voice support.  
Business: Product recommendations, automated order processing, CRM integration, and real-time shipping tracking.

## Conclusion

The chatbot demonstrates a practical AI solution for e-commerce. By combining NLP and web technologies, it enhances efficiency and customer satisfaction. Its modular, scalable design and comprehensive documentation make it suitable for both educational and commercial applications.