

GENAI PROJECT REPORT

1. Requirements Document

Introduction

This project involves developing an AI-powered customer support chatbot tailored specifically for e-commerce and service-based businesses. It is designed to provide continuous (24/7), intelligent, multilingual customer support utilizing a locally hosted large language model (LLM), Mistral 7B, facilitated through Ollama. This solution aims to significantly enhance customer satisfaction and reduce operational expenses.

Functional Requirements

- Chatbot must interpret natural language customer inputs.
- Chatbot must provide predefined responses to common customer queries (e.g., return policies, order status).
- Must escalate complex or unrecognized queries to human customer service representatives.
- Must run locally via Ollama, enabling offline functionality.
- Interaction logging for analysis and continuous system enhancement.

Non-Functional Requirements

- Response times within 2 seconds.
- Operationally independent, no external API dependencies.
- Local data handling to guarantee privacy.
- Optimized for performance on standard laptops with at least 8GB RAM.

User Interface Requirements

- Clean and user-friendly chat interface.
- Built using Streamlit, compatible with modern browsers.
- Responsive design for desktop and mobile use.

2. Prioritized User Stories (MoSCoW Method)

User stories below follow the format: "As a , I want , so that ." Must-Haves:

- As a customer, I want to interact with the chatbot using natural language, so that I can get support for my straightforward queries easily without needing specific commands.
- As a customer, I want to receive accurate and predefined responses to frequent queries (like return policies or refund status), so that I can quickly get the standard information I need.
- As a customer, I want the chatbot to function locally without relying on external APIs, so that I can access support reliably even with intermittent internet and my data interaction remains private.
- As a customer, I want the chatbot to escalate my complex or unrecognized queries to a human representative, so that I can get my issue resolved even when the chatbot cannot handle it.

Should-Haves:

- As a customer, I want to see visual feedback (e.g., a typing indicator) while the chatbot is processing my request, so that I know it's working and haven't missed my input.
- As a developer/maintainer, I want clear project file structures and comprehensive documentation, so that I can easily understand, maintain, and enhance the chatbot codebase efficiently.

Could-Haves:

- As a customer who prefers a language other than the default, I want to interact with the chatbot in my preferred language, so that I can receive support effectively regardless of language barriers.
- As a manager/analyst, I want to access anonymized logs of customer interactions, so that I can analyze usage patterns and identify opportunities for chatbot and service improvement.

Won't-Haves (for MVP):

- (No user story needed, but confirms scope): Voice interaction capabilities will not be developed in this iteration.

- (No user story needed, but confirms scope): Deep user-specific personalization based on past interaction history will not be implemented in this iteration.

3. Roadmap

Phase	Timeline	Key Deliverables & Use Cases	Milestones
Research & Planning	April 1 - 3	Requirements & Technical Exploration; High-Level Architecture; LLM (Mistral) Core Logic Approach; Roles & Permissions; Relevance & Selectiveness	Approval of Roadmap
Backend Development	April 4 - 8	Core Logic Implementation (Mistral-based LLM); Functional & Technical Integration; Chatbot Parser (Offline Handling); Role-Based Access & Security Layers	MVP by April 8
Frontend	April 9 - 11	UI Prototyping; Frontend Integration; Responsive & Interactive Design	Prototype Ready by April 11
Refinement & UX Testing	Starting April 11 (ongoing)	Integration (Frontend & Backend); Edge Case Tests & UX Feedback; Performance/Usability Improvements	User Readiness

Deployment	April 18	Streamlined Code; Integrated Logs Module; Production Launch Preparations	Production Release
Documentation & Final Setup	April 30	Project Documentation (GitHub); Live Demo Access; Final Testing & Configuration	Project Completion

4. Tasks

Research & Planning:

- Finalize e-commerce/service sector use cases (Harshal) — Leverages expertise in AI and customer service domains.
- Benchmark local LLMs including Mistral 7B (Harshal) — Utilizes deep knowledge of LLMs and optimization.

Backend Development:

- Develop natural language parser and predefined query handler (Harshal) — Applies NLP and AI integration experience.
- Integrate local LLM execution via Ollama API (Harshal) — Expertise in local deployments and APIs.
- Implement interaction logging system in JSON format (Harshal) — Backend skills for robust logging.

Frontend Development:

- Design Streamlit UI wireframes (Figma) (Vishnu) — Full-stack and UI/UX design skills.
- Develop responsive chat interface with real-time updates (Vishnu) — Expertise in frontend technologies.
- Incorporate visual feedback indicators (Vishnu) — Knowledge of UI components for enhanced clarity.

- Optimize responsiveness with CSS media queries (Vishnu) — Experience in responsive design.

Testing & UX Refinement:

- Create edge case test scripts (Abhay) — UX and testing expertise.
- Conduct user testing with minimum 5 participants (Abhay) — Skills in feedback collection and analysis.
- Analyze logs for frequent unrecognized queries (Abhay) — Analytical skills for improvement.
- Optimize interaction flow based on feedback (Abhay) — UX design experience.

Deployment:

- Configure Streamlit Cloud environment (Vishnu & Harshal) — Deployment and backend collaboration.
- Set up GitHub Actions for CI/CD (Harshal) — CI/CD and repository management skills.
- Draft detailed user guide and documentation (Harshal) — Technical writing abilities.

5. Responsibilities and Roles

Harshal -- Project Lead / LLM Engineer

- Chatbot logic using Mistral 7B.
- Local LLM integration via Ollama API.
- Interaction logging system.
- GitHub repository and technical documentation.
- Alignment: Core functionality and technical excellence.

Vishnu -- Full Stack Developer

- Streamlit UI and responsive design.
- Backend integration.
- Streamlit Cloud deployment and CI/CD pipeline.

- Alignment: Polished interface and streamlined deployment.

Abhay — UX / Feedback & Testing Lead

- Chatbot conversation design.
- User testing and feedback collection.
- Interaction logs analysis.
- UX optimization recommendations.
- Alignment: Intuitive, user-centric experience.

Collaboration Framework:

- Weekly meetings every Tuesday at 10 AM via Zoom.
- Integration checkpoints every Friday.
- Team consensus for major decisions; Harshal as final decision-maker.
- Documentation in GitHub wiki.
- Bi-weekly user feedback reports reviewed for UX improvements.