Software Requirements Specification (SRS)

Project: College Chatbot Web Application

Technology Stack: Rasa, GPT-2, Python Flask, Render Deployment

---

1. Introduction

The College Chatbot Web Application is designed to assist students, faculty, and visitors by answering queries related to admissions, courses, fees, placements, faculty information, and events. The system uses Rasa for intent detection and GPT-2 for fallback generative responses.

2. Scope

The system provides real-time responses to college-related queries through a web interface. It integrates Rasa for intent classification, GPT-2 for dynamic responses, and Flask as the backend framework. The application is deployed on the Render cloud platform.

3. Overall System Architecture

User → Web Interface → Flask Backend → Rasa (Intent Detection) If intent confidence is high → predefined response If confidence is low → GPT-2 generates response All chat logs are stored in the database.

4. Functional Requirements

FR1: User can send messages through chat interface. FR2: System shall classify user intent using Rasa. FR3: System shall generate fallback response using GPT-2 if intent confidence is low. FR4: Chat logs shall be stored with timestamp. FR5: Admin panel (optional) to manage FAQs and retrain model.

5. Non-Functional Requirements

Performance: Response time should be under 3 seconds. Security: Input validation and HTTPS support. Reliability: 99% uptime on Render. Scalability: Can upgrade to advanced language models.

6. Database Design

Table: chat_logs Fields: - id (Primary Key) - user_message (Text) - bot_response (Text) - timestamp (DateTime)

7. Deployment Requirements

- Code hosted on GitHub - Deployed as Web Service on Render - Start command: gunicorn app:app - Environment variables configured properly

8. Future Enhancements

- Voice-based chatbot - Multilingual support - ERP system integration - WhatsApp chatbot integration