**MBM Admissions Support Portal Chatbot**

**Problem Statement:**

MBM Engineering College faces recurring challenges during admissions:

* **For students**: Repetitive queries about eligibility, procedures, deadlines, and required documents create long wait times and confusion.
* **For admissions officials**: Managing, responding to inquiries, and tracking procedural updates manually is time-consuming, error-prone, and unsustainable during peak periods.

**Goal:**

Build an AI-powered chatbot that:

1. **Helps students** get instant, reliable answers to admissions-related questions based on historical FAQs and current guidelines.
2. **Assists admissions officials** by providing internal tools for process guidance, automation of standard replies, and access to an internal command interface to track and manage applicant interactions.

**Value Proposition:**

**For students:**

* Quick, 24/7 responses to queries
* Easy navigation of rules, dates, and forms
* Transparent communication throughout the admission cycle

**For officials:**

* Fewer repeated inquiries
* Faster applicant tracking and interaction summaries
* Improved operational efficiency with smart command-based insights (e.g., number of applicants missing documents, application statistics)

**Scope:**

* Student-facing chatbot integrated with admissions portal
* Retrieval-augmented Q&A from past and current admission data
* Internal dashboard for officials (basic release)
* Feedback logging from both students and staff

**Further enchantments options:**

* Deep CRM integration or document upload from within chatbot
* Voice assistant features
* Admin user management or authentication layers beyond basic role separation

**Key Metrics:**

* Average time to resolve queries
* Query response accuracy
* Student satisfaction score (via feedback)
* Admissions official efficiency improvements (e.g., # of tasks automated)

**Sprint Plan with User Stories:**

**Sprint Duration:** 1 week **Team:** AI/LLM/NLP Engineer, UI Developer, Testing Engineer, Data Engineer

**Sprint 1: Data Collection & Use Case Mapping**

**Goal**:

Identify key data sources and define workflows for students and officials.

**User Stories**:

* As a data engineer, collect past admission FAQs, chat logs, and notices so that we can train the chatbot accurately.
* Consult an admissions officer, to list common queries and internal tasks so that we can support both user groups.
* The team collectively should define MVP(Minimum Viable Product) use cases for students and staff separately to plan development priorities.

**Deliverables**:

* Curated and labeled datasets
* Student and staff use case matrix
* MVP scope document (split into public and admin features)

**Sprint 2: Build RAG System & Admin Interface (Basic)**

**Goal**:

Build backend with RAG and set up a lightweight admin panel for staff.

**User Stories**:

* A Data Engineer should embed admissions content into a vector DB so that a semantic search can be performed.
* An official should be able to provide simple command-based inputs (e.g., “list applicants missing 12th marksheet”) so accurate results and summaries can be retrieved quickly.
* *AI developer*, should create a basic staff dashboard or dashboard for query logging, RAG response display, quick reference links display

**Deliverables**:

* RAG pipeline for public + internal queries
* Simple staff-facing query interface (internal chatbot or dashboard)
* Test dataset for internal command examples

**Sprint 3: Chatbot UI & Portal Integration**

**Goal**:

Build chat interface for students and connect staff dashboard to backend.

**User Stories**:

* A student, be able to interact with a friendly chat interface to ask any doubts anytime.
* UI developer should integrate the chatbot widget into the MBM admissions portal.
* The query logs and frequently asked questions should be anonymous and must be accessed only by admin.

**Deliverables**:

* Live chatbot interface on the portal
* Admin log view for admissions officials
* UI enhancements based on feedback

**Sprint 4: Testing, Feedback & Improvements**

**Goal**:

Final testing and feedback loop from both users and staff.

**User Stories**:

* A Test Engineer, should validate both student and official flows across at least 50 questions/tasks.
* gather structured feedback from real students and officials to guide further improvements.
* *As a developer*, I want to refine system responses and improve model coverage based on test results.

**Deliverables**:

* Updated response accuracy
* Staff/stakeholder feedback summary
* Final MVP deployment with usage documentation

**Continuous Improvement Plan:**

* Weekly retrospectives with officials for insights
* Feedback form embedded into chatbot UI
* Future backlog: CRM integration, multilingual support, real-time status tracking, live document validation