

AI-Powered Virtual Assistant for SIBA
FYP Code:29
Software Design Specification

BY:

Muhammad Uzair	CMS. ID: 023-22-0235
Mujeebullah	CMS. ID: 023-22-0271
Afifa Nazeer	CMS. ID: 023-22-0003



Supervised by: Dr Zakira Jamali

Signature of Supervisor

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCE AND INFORMATION TECHNOLOGY
SUKKUR IBA UNIVERSITY

Document History

Version	Name of Person	Date	Description of change
1.0	Muhammad Uzair, Mujeebullah, Afifa Nazeer	30-10-25	Draft1
1.1	Dr Zakira Jamali	31-10-25	Review
1.2	Muhammad Uzair, Mujeebullah, Afifa Nazeer	02-11-25	Revised

1. INTRODUCTION	1
1.1. Purpose of Document	1
1.2. Definitions, acronyms, and abbreviations	1
2. OVERALL SYSTEM DESCRIPTION.....	2
2.1. Project Background.....	2
2.2. Project Objectives	3
3. SYSTEM ARCHITECTURE	4
4. DOMAIN MODEL	6
5. CLASS DIAGRAM	7
6. DATABASE DIAGRAM.....	8
7. ENTITY RELATIONSHIP DIAGRAM (ERD)	9
8. SEQUENCE DIAGRAM	10
9. SYSTEM INTERFACE DESIGN.....	17
10. TEST CASES	23

1. Introduction

1.1 Purpose of Document

The purpose of this document is to define Software Design Specification (SDS) for the Ai Powered Virtual Assistant of SIBA. Its primary purpose is to serve as a comprehensive technical blueprint that guides the development of an intelligent, web-based assistant capable of providing accurate responses to student and faculty queries. This document outlines the system's architecture, design components, data flow, and interaction between modules to ensure a clear understanding of how the chatbot operates and fulfills its objectives. This document serves as a blueprint for developers, testers, and stakeholders, providing detailed design specifications for implementing key functionalities such as data automation, admin-based content management, retrieval-augmented generation (RAG), integration of LLMs, the use of MongoDB and VectorDB, authentication, and role-based access control. Overall, this SDS ensures that the system is designed to deliver real-time, personalized, and up-to data information while maintaining scalability, reliability, and security.

1.1 Definitions, acronyms, and abbreviations

Table 1.1 Abbreviations and Definitions

Acronyms/Abbreviations	Definitions
AI (Artificial Intelligence)	It is capability of a system to simulate human-like reasoning and understanding
RAG (Retrieval-Augmented Generation)	A technique in which relevant information is fetched from knowledge base and sent to LLM to generate response.
LLM (Large Language Model)	An advanced AI model used to process and generate text data.
SST (Speech-to-Text)	A feature which converts voice into text.
TTS (Text-to-Speech)	A feature which converts text into voice

VectorDB	A database is used to store the embeddings of data.
API (Application Interface)	A set rules and protocols that allow communication between different software components.
XML (Extensible Markup Language)	A markup language for structuring like HTML
JSON (JavaScript Object Notation)	A light-weight data-interchange format used for data storage and transfer
DB (Database)	A system which is used for managing and storing data

2 Overall System Description

2.1 Project Background

In Sukkur IBA University, students and faculty members frequently require quick access to a wide range of academic and administrative information, such as course outlines, policies, faculty details, timetables, events, and scholarship opportunities. However, manually searching through university websites can be time-consuming and inefficient. To address these challenges, the AI-Powered Virtual Assistant for SIBA has been conceptualized as an intelligent digital assistant designed to provide instant, accurate, and context-aware responses to user queries.

The system leverages Artificial Intelligence (AI) and Natural Language Preprocessing (NLP) to understand user intent and deliver relevant information in conversational form. It integrates automation mechanisms including email-based data ingestion and information extraction to ensure continuous and reliable updates of university data such as timetables, events, and scholarship information. The chatbot's architecture incorporates MongoDB for storing structured data and a Vector Database for managing embedded text representations used in Retrieval-Augmented Generation (RAG) responses.

Furthermore, the system supports role-based access control (RBAC) to distinguish between guest and authenticated users with SIBA mail, ensuring data privacy and personalized interactions. With additional features such as Speech-to-Text (STT), Text-to-Speech (TTS), chat history management and sharing capabilities, the project aims to enhance the accessibility and efficiency of university communication systems. Ultimately, this chatbot represents a step towards digital transformation within academic institutions by automating information delivery and improving the overall user experience.

2.2 Project Objectives

The primary objectives of the project:

1. **Develop an AI-Chatbot:** Create an intelligent chatbot capable of understanding and responding to natural language queries using large language model (LLM).
2. **Enable Dynamic Data Management:** Implement an email-based ingestion and information extraction mechanism to automatically process and update university-related data such as timetables, events, and scholarship information. In addition, provide an admin-based content management interface that allows authorized administrators to add, update, and maintain dynamic institutional content within the system, ensuring data accuracy and consistency.
3. **Provide Multi-Domain Information:** Allow users to retrieve details related to policies, faculty profiles, course outlines, events and scholarship opportunities, ensure comprehensive information coverage.
4. **Support Role-Based Access Control (RBAC):** Restrict and customize chatbot responses based on user roles (guest, student, faculty) to maintain privacy and personalization. Guests can access general information while students, faculty can access all types of information.
5. **Speech Interaction Capabilities:** Integrate Speech-to-Text (STT) and Text-to-Speech features for a hands-free, voice-enabled interaction experience.
6. **Maintain Chat History and Sharing Options:** Store user chat sessions and allow users to share useful conversions for reference or collaboration.

3 System Architecture

The AI-Powered Virtual Assistant for SIBA follows a layered client–server architecture with a modular backend design. The system operates as a single backend application while keeping functionalities logically separated to ensure clarity, maintainability, and scalability.

1. Presentation Layer

This layer provides user interaction interfaces:

- **Chatbot UI:** Allows students and users to interact with the virtual assistant using text or voice.
- **Admin Panel:** Enables administrators to manage content, users, and system configurations.

Both interfaces communicate with the backend through **secure RESTful APIs**.

2. Application Layer

This layer contains the core business logic of the system, including:

- **Authentication & RBAC (Role-Based Access Control)** to manage user permissions
- **Query Processing Module** to handle user requests
- **Admin Content Management** Module for updating knowledge and system data

All modules are organized separately but deployed together as a single backend application.

3. AI / RAG Layer

This layer enables intelligent responses:

- A Retriever performs semantic search using the Vector Database
- An LLM (Large Language Model) generates accurate and context-aware responses based on retrieved data

4. Data Layer

The system uses two databases:

- MongoDB for structured application data (users, logs, content)
- Vector Database for storing embeddings used in semantic search

5. External API Layer

This layer integrates third-party services, including:

- Speech-to-Text API for voice input
- Text-to-Speech API for voice responses

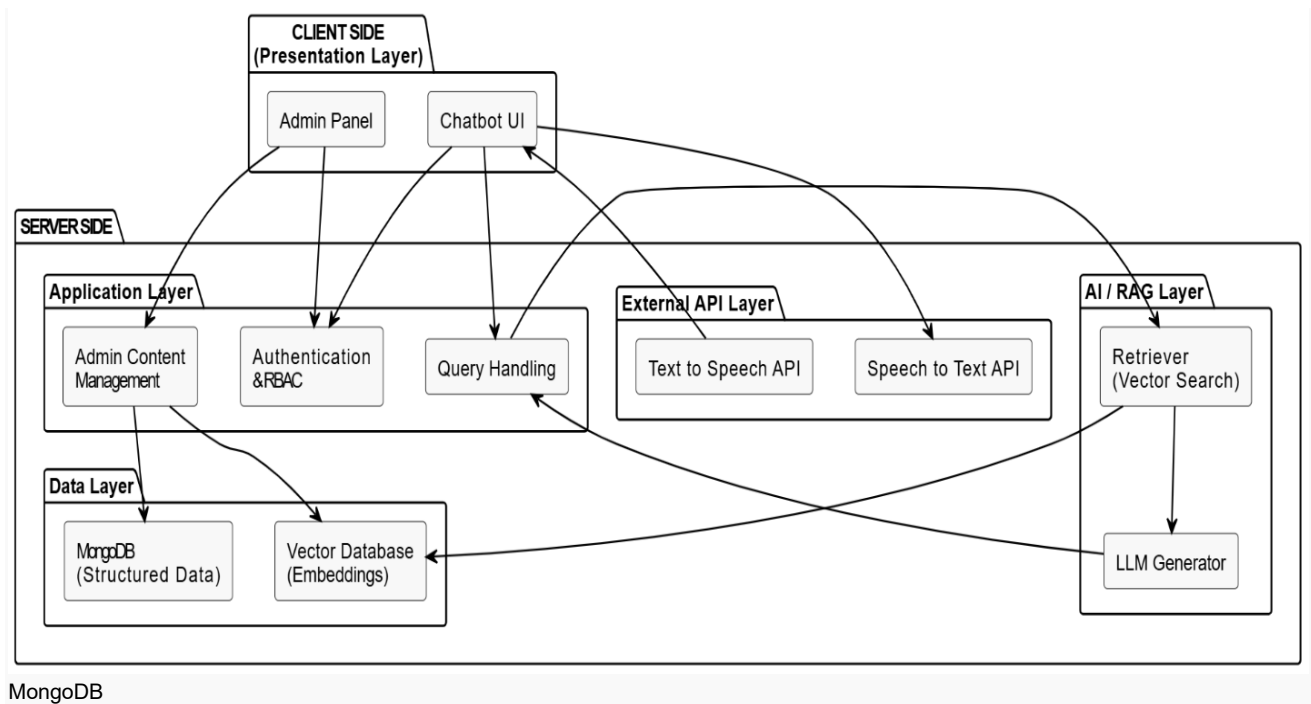


Fig 3.1 System Architecture

4 Domain Modal

The SIBA Virtual Assistant domain model integrates user interaction, conversation management, and knowledge handling. Users, categorized as IBA-Associated and Non-IBA-Associated, initiate chat sessions and submit queries to the system. Administrators manage the centralized knowledge content, which is linked with vector embeddings to support efficient semantic search and intelligent response generation.

The model establishes a direct dependency where system Responses are produced from this verified Knowledge Content. This structure ensures that all answers are grounded in accurate university data (RAG architecture) while clearly distinguishing between internal and external user access.

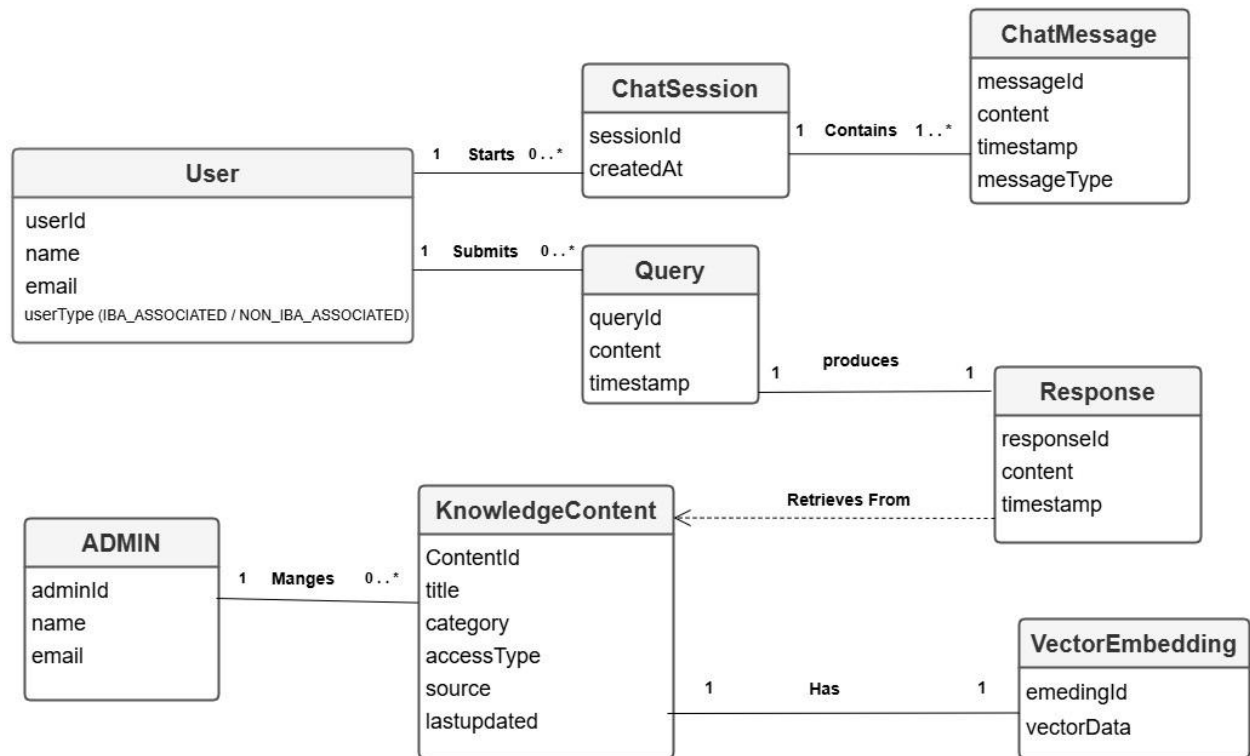


Fig 4.1 Domain Modal of System

5 Class Diagram

This diagram illustrates the high-level object-oriented structure, mapping relationships between actors (User, Admin) and core components like Chatbot System and LLMService. It details the specific classes and methods required to process text and voice queries using Retrieval-Augmented Generation (RAG). Furthermore, it defines the administrative components responsible for managing content and vector embeddings, providing a clear blueprint of the system's static architecture.

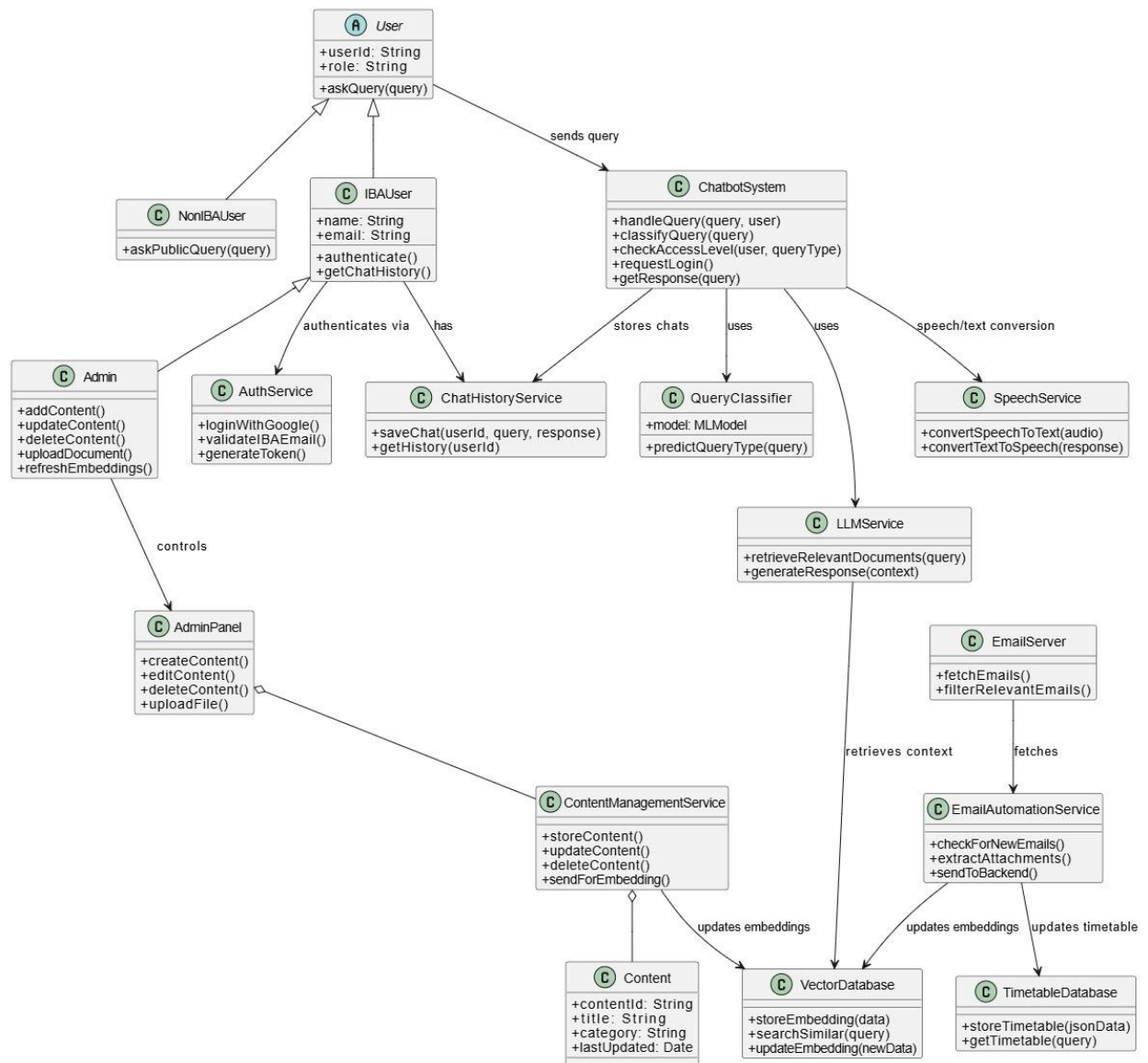


Fig 5.1 Class Diagram of System

6 Database Diagram

The database diagram presents the MongoDB-based data structure used by the AI-Powered Virtual Assistant for SIBA. It defines collections such as Users, Sessions, Chats, Messages, Scraped Data, Embeddings, and Timetables, along with their attributes and relationships. The diagram shows how user interactions, chat histories, extracted email data, and vector embeddings are stored and linked to support authentication, conversation tracking, and semantic search. This structure ensures efficient data management, scalability, and seamless integration with AI-based retrieval mechanisms.

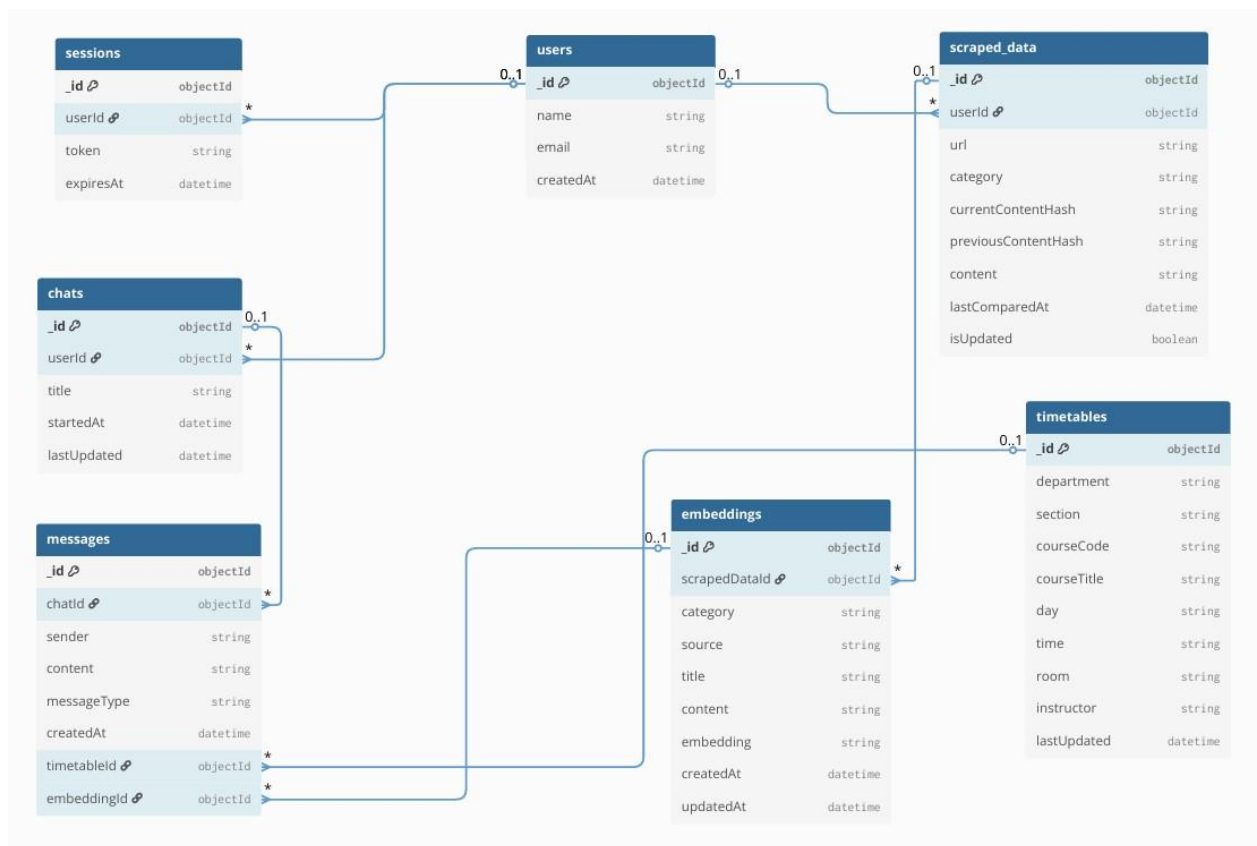


Fig 6.1 Database Diagram

7 Entity Relationship Diagram (ERD)

The Entity Relationship Diagram (ERD) illustrates the conceptual data model of the AI-Powered Virtual Assistant for SIBA by highlighting key entities and their relationships. It captures how Users interact with Chats and Messages, how sessions are tracked for security, and how knowledge sources such as Knowledge Base and Email Sources contribute information to the system. The ERD also shows the role of Admin in managing content and updates, ensuring accurate and up-to-date information. Overall, the ERD provides a clear conceptual understanding of data flow and relationships within the system.

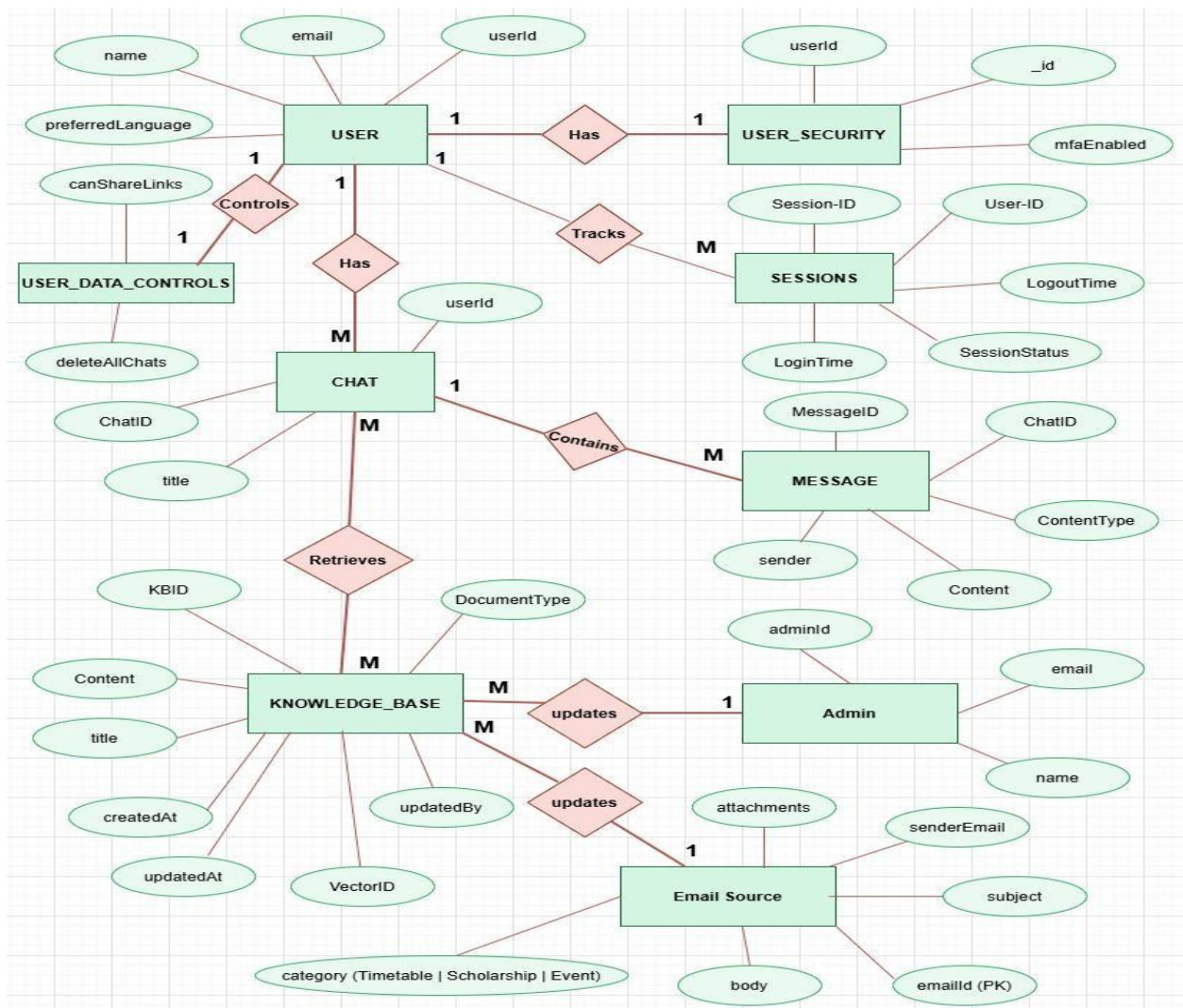


Fig 7.1 Entity Relationship Diagram (ERD)

8 Sequence Diagram

8.1 User Signup / Login via IBA Gmail Authentication

This diagram illustrates the secure login process for the application. The system uses Google OAuth to authenticate users but enforces a strict domain validation rule.

Specifically, the Auth Service checks if the user's email address ends with @iba.edu.pk. If the domain is valid, the system checks the internal database to either log the user in or create a new account. This ensures the platform is restricted exclusively to authorized university personnel and students.

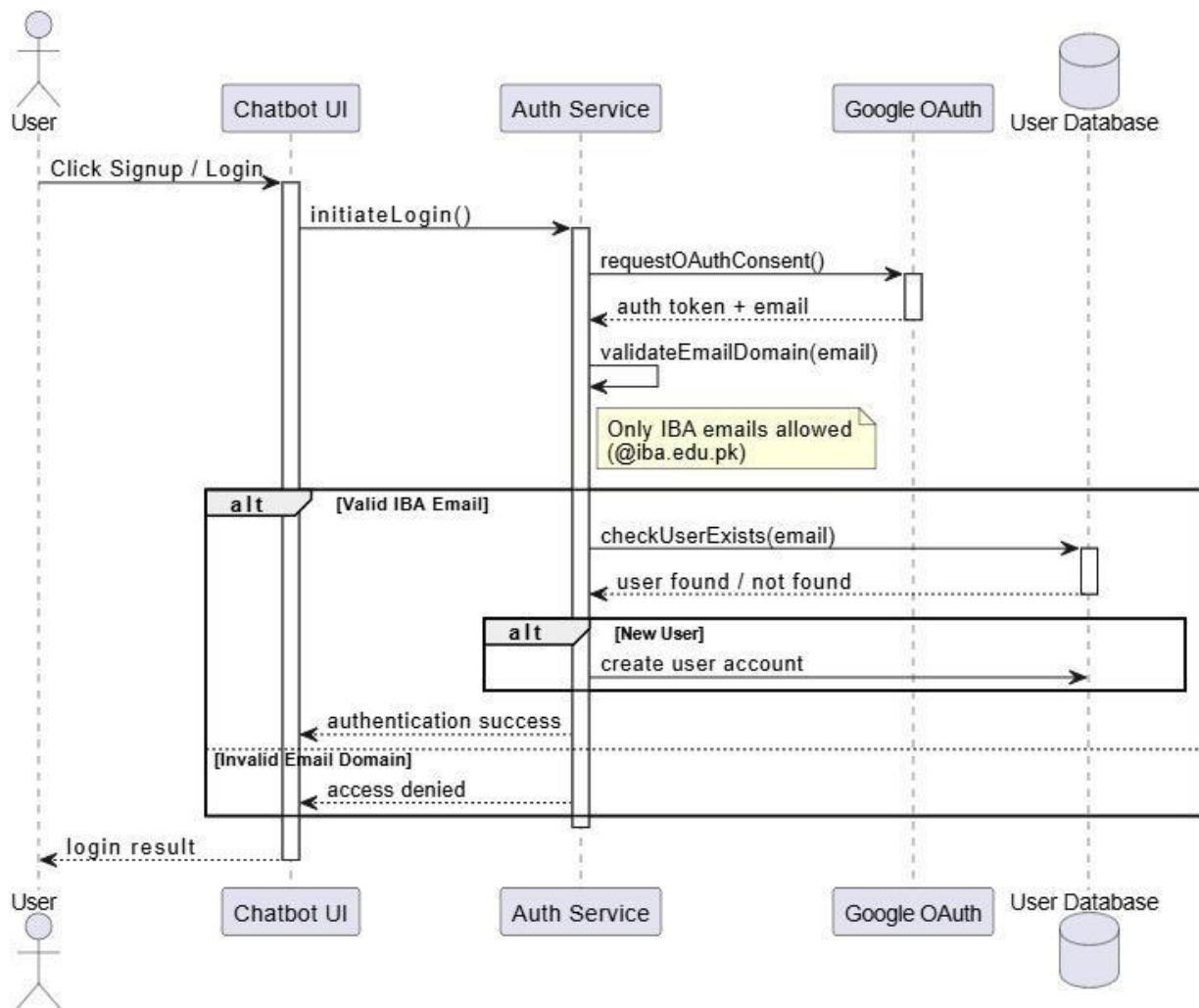


Fig 8.1: User Authentication Flow

8.2 General Query Processing

This diagram outlines the standard flow for answering general user questions. The system utilizes a Retrieval-Augmented Generation (RAG) approach. When a query is received, the system first searches the Vector Database for similar documents to gather relevant context. This context is then passed to the Large Language Model (LLM) Service, which generates a natural language response to be displayed to the user.

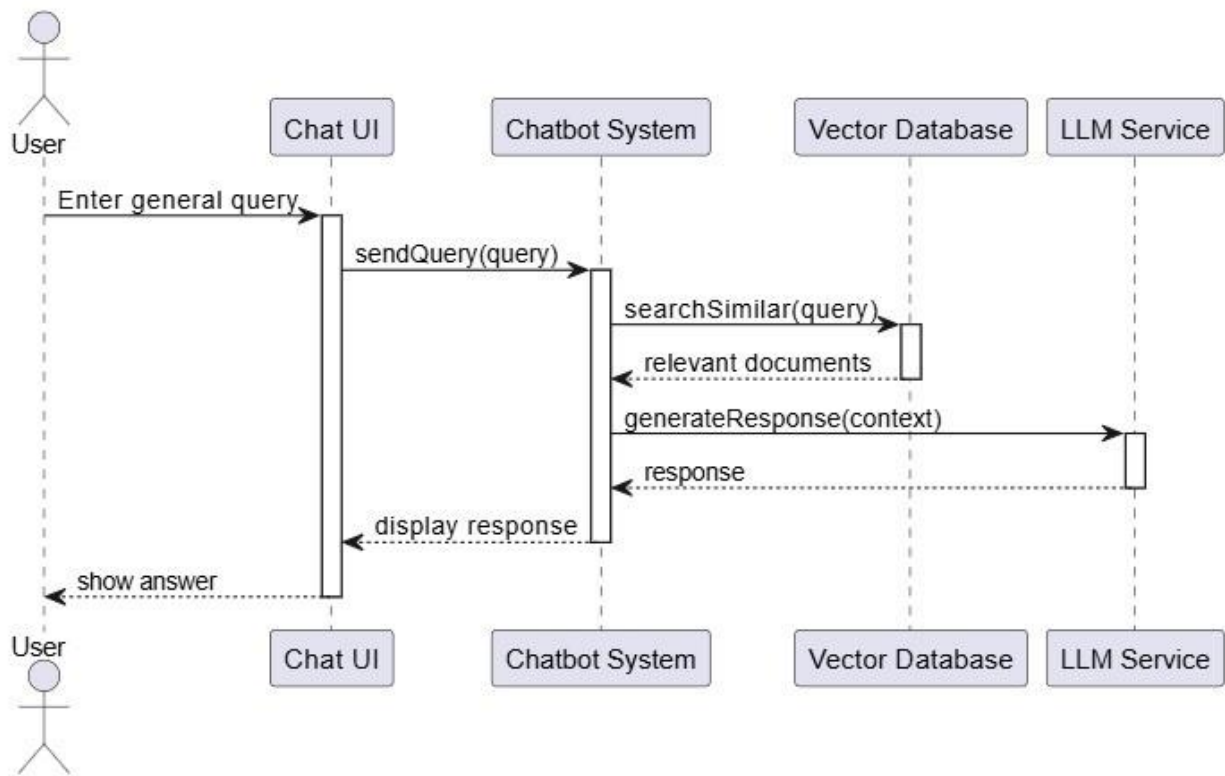


Fig 8.2: General Query Processing Flow

8.3 Create New Chat Session

This sequence demonstrates how a new conversation is established. When a user requests a new chat, the Chatbot System generates a unique session identifier and persists the session details in the Chat Database. This ID is returned to the UI, allowing the system to track conversation history and maintain context for subsequent interactions.

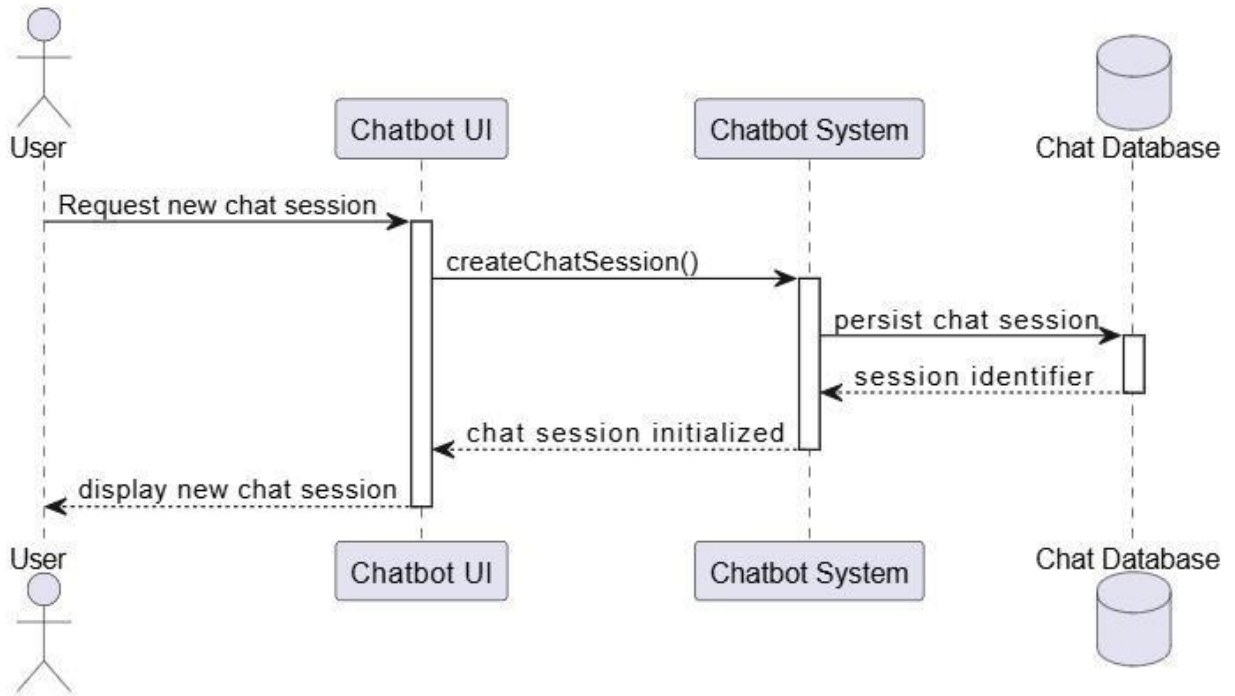


Fig 8.3: Chat Session Creation Flow

8.4 Ask Restricted Query

This flow highlights the security measures applied to sensitive or restricted academic data. Before processing a query, the Chatbot System calls the Auth Service to validate the user's access rights (validate Access). Only if access is granted does the system search the authorized knowledge base and generate a response. The conversation then persisted in the database for audit and history purposes.

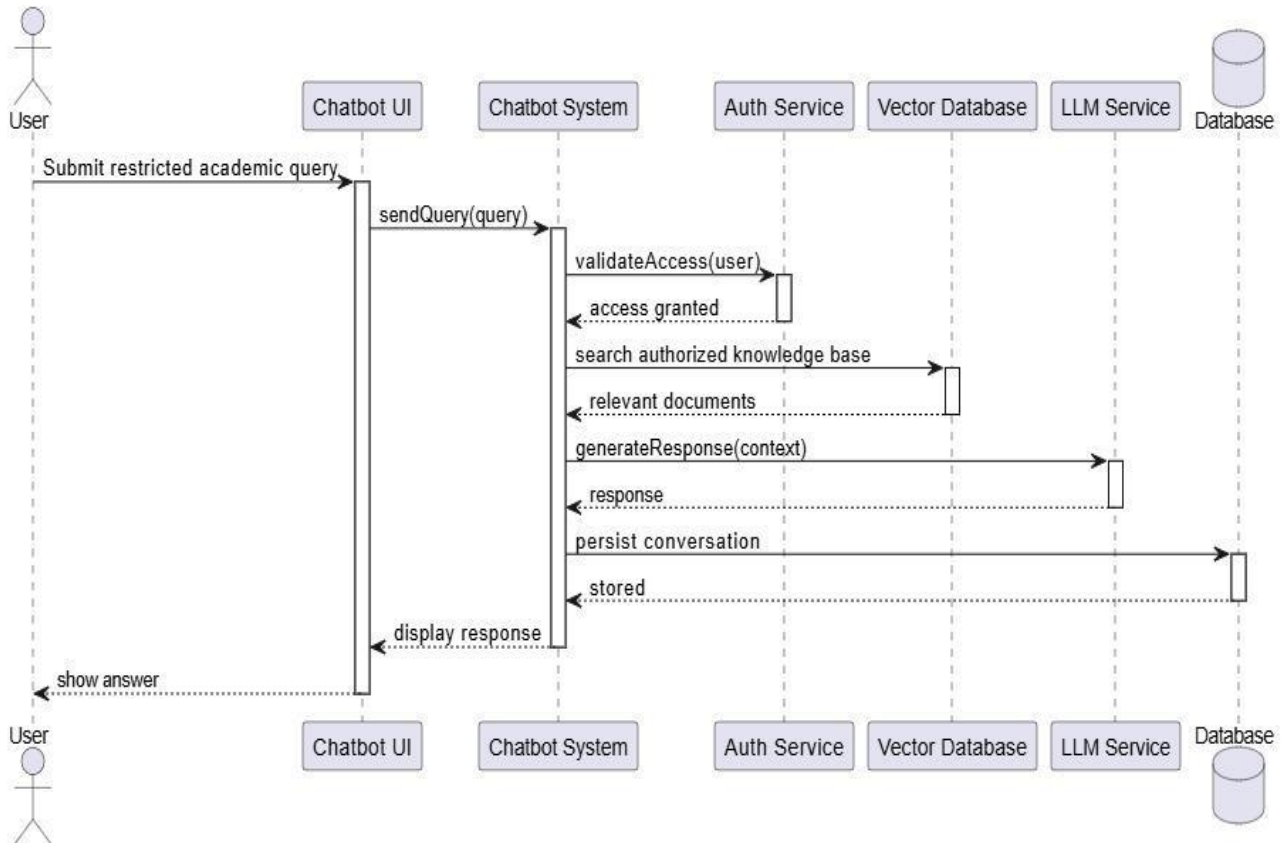


Fig 8.4: Restricted Query Handling Flow

8.5 Voice Mode Interaction

This diagram details the multimodal voice interaction feature. The process begins with the user tapping the microphone, capturing audio which is converted to text via a Speech-to-Text (STT) service. This text is processed similarly to a standard query. Once the LLM generates a text response, the system performs two parallel actions: it displays the text on the screen and simultaneously converts it into audio using a Text-to-Speech (TTS) service for the user to hear.

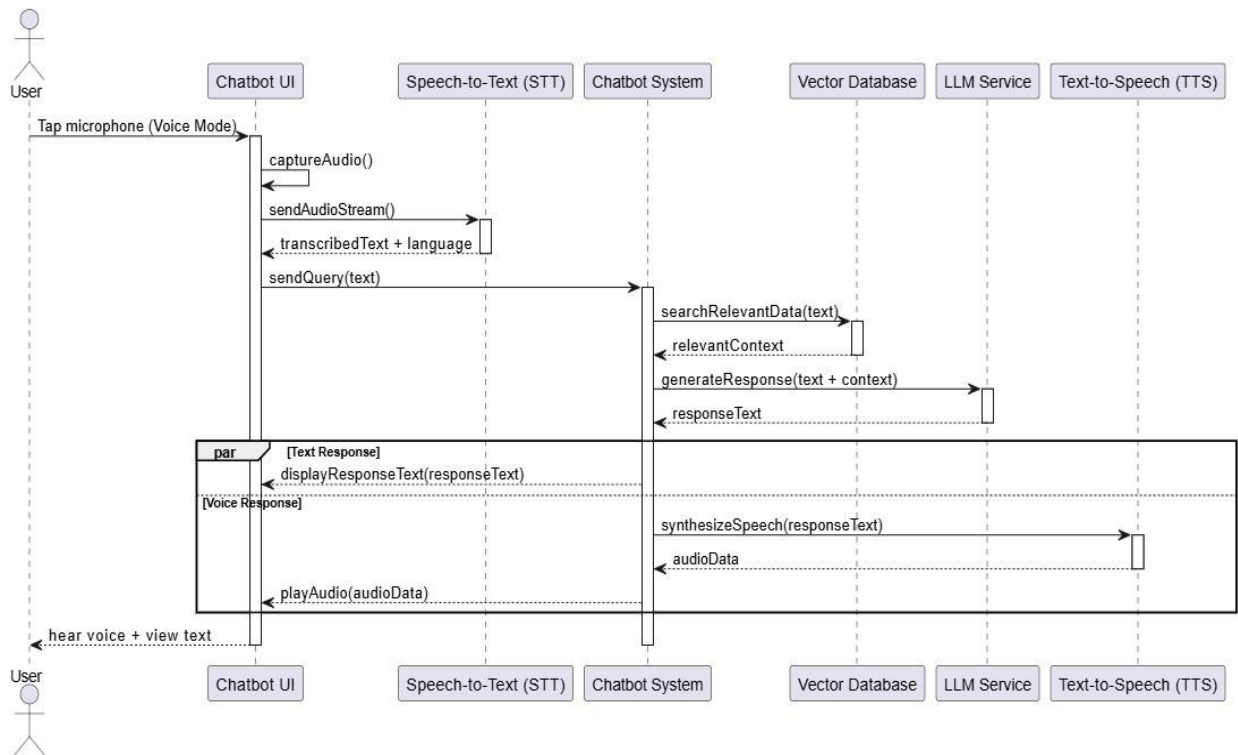


Fig 8.5: Voice Interaction Flow

8.6 Admin Content Management

This diagram shows the administrative workflow for managing the knowledge base. When an admin adds, updates, or deletes content via the Admin Panel, the system updates the primary MongoDB storage. Crucially, the system also automatically regenerates embeddings for the new content and updates the Vector Database. This synchronization ensures consistency between the raw data and the AI's search index.

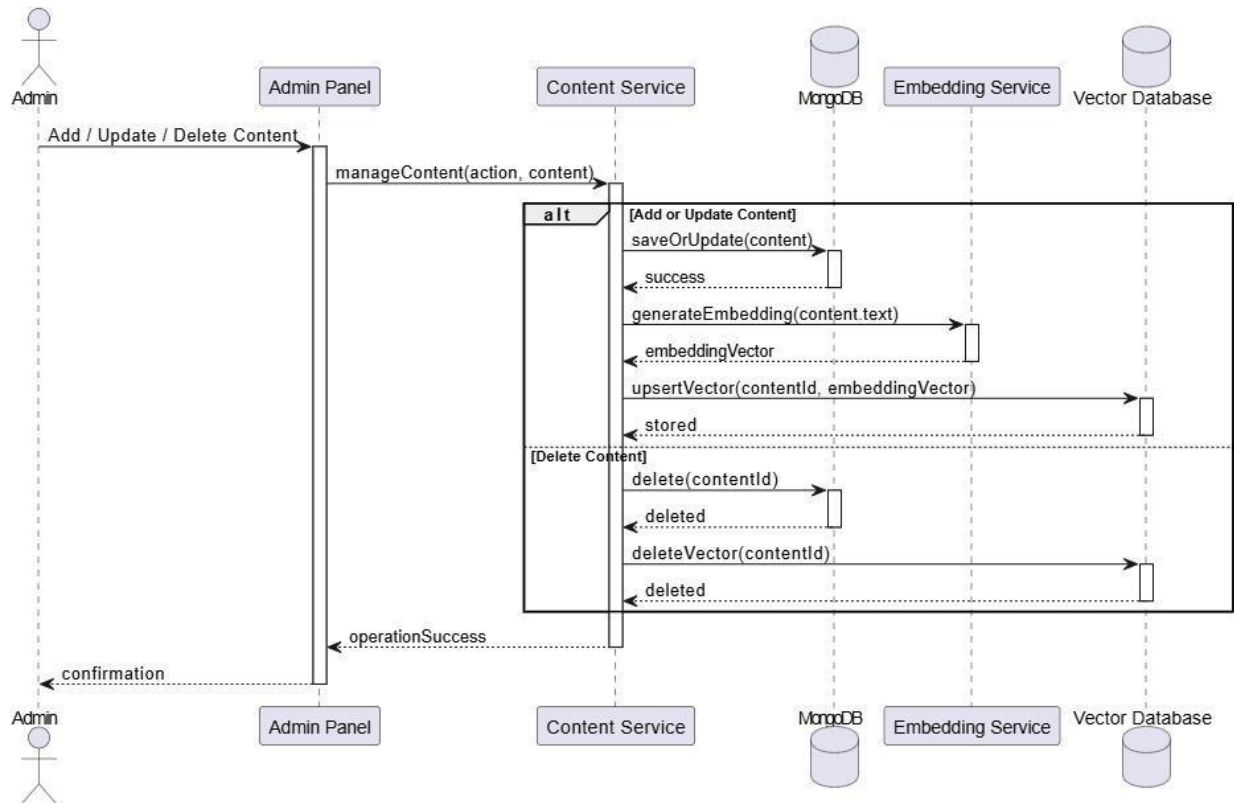


Fig 8.6: Admin Content Management Flow

8.7 Automated Email Ingestion and Information Extraction

This diagram depicts the background automation for keeping the system updated via email. When the Email Server receives a message, the Automation Service triggers a process to extract and classify the content (e.g., timetables, events, scholarships) using an LLM. The structured data is stored in MongoDB, and embeddings are generated and unsorted into the Vector Database, ensuring the chatbot always has the latest institutional information.

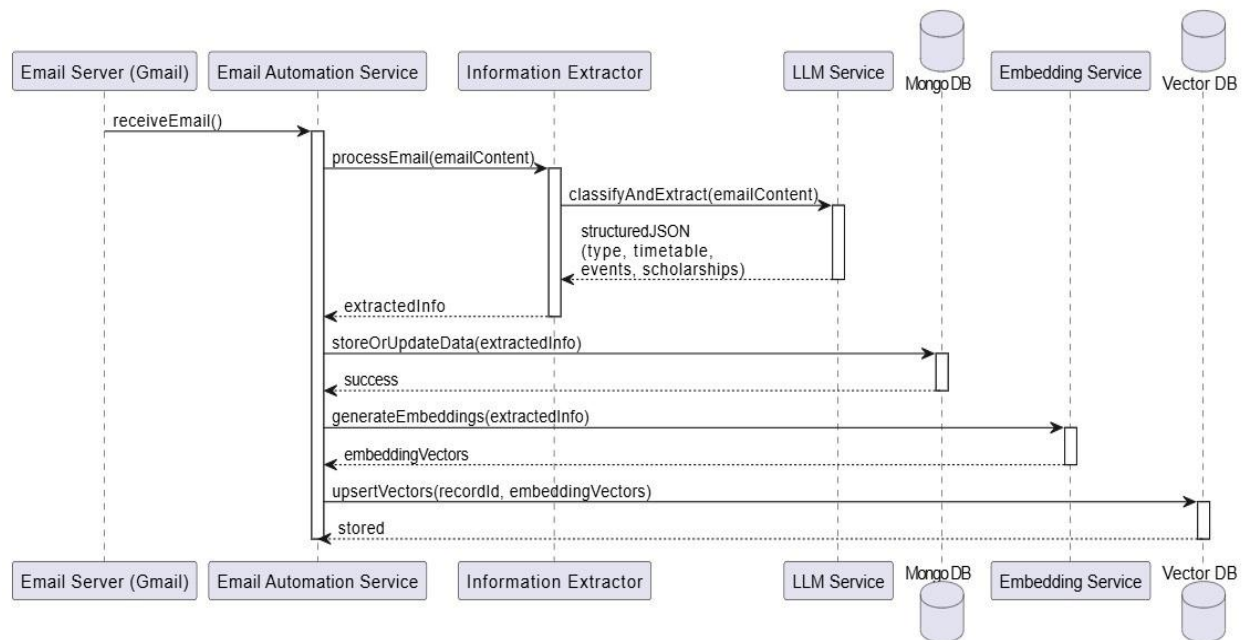


Fig 8.7: Automated Email Processing Flow

9 System Interface Design

System Interface Design describes how different users interact with the AI-Powered Virtual Assistant for SIBA through clear and role-based interfaces. It includes a Guest interface, where users can access basic information and view general content, and an authenticated user interface, where logged-in users can submit text or voice queries and receive intelligent responses from the chatbot. The Admin Panel interface is a separate, secure interface used only for content management, such as adding, updating, or deleting knowledge base content. Overall, this design ensures smooth interaction between users, administrators, and the backend AI system, while maintaining security, usability, and a consistent user experience.

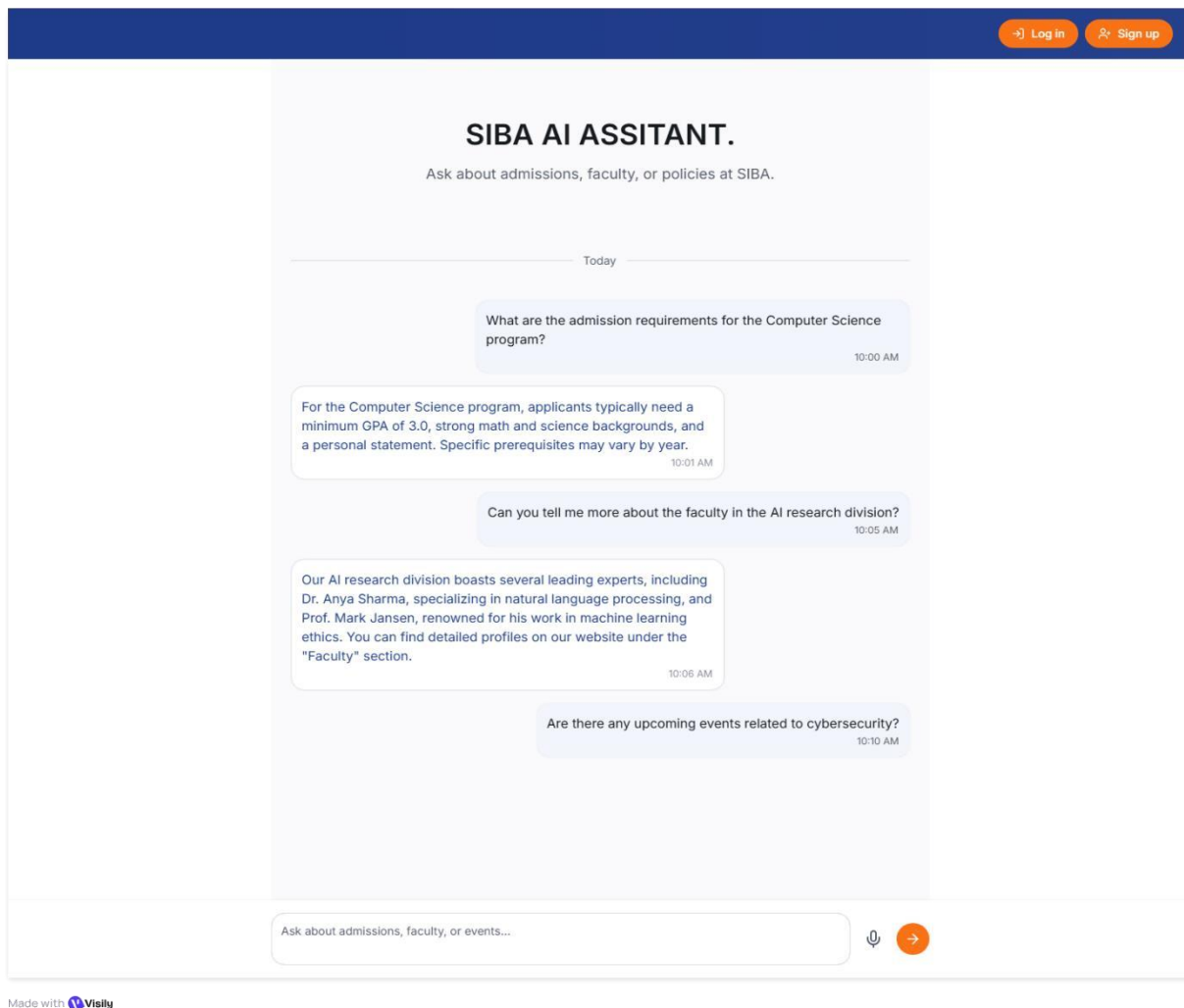


Fig 9.1 Guest Mode Interface

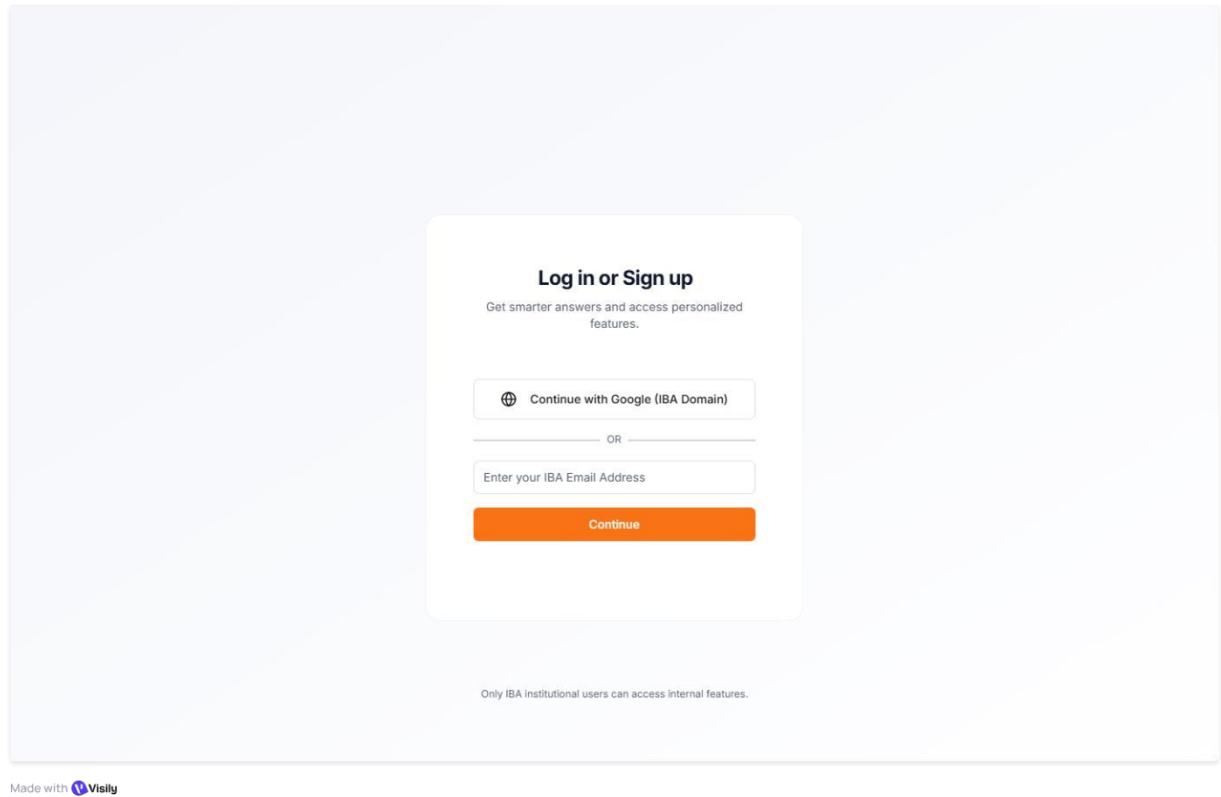


Fig 9.2 Login Interface

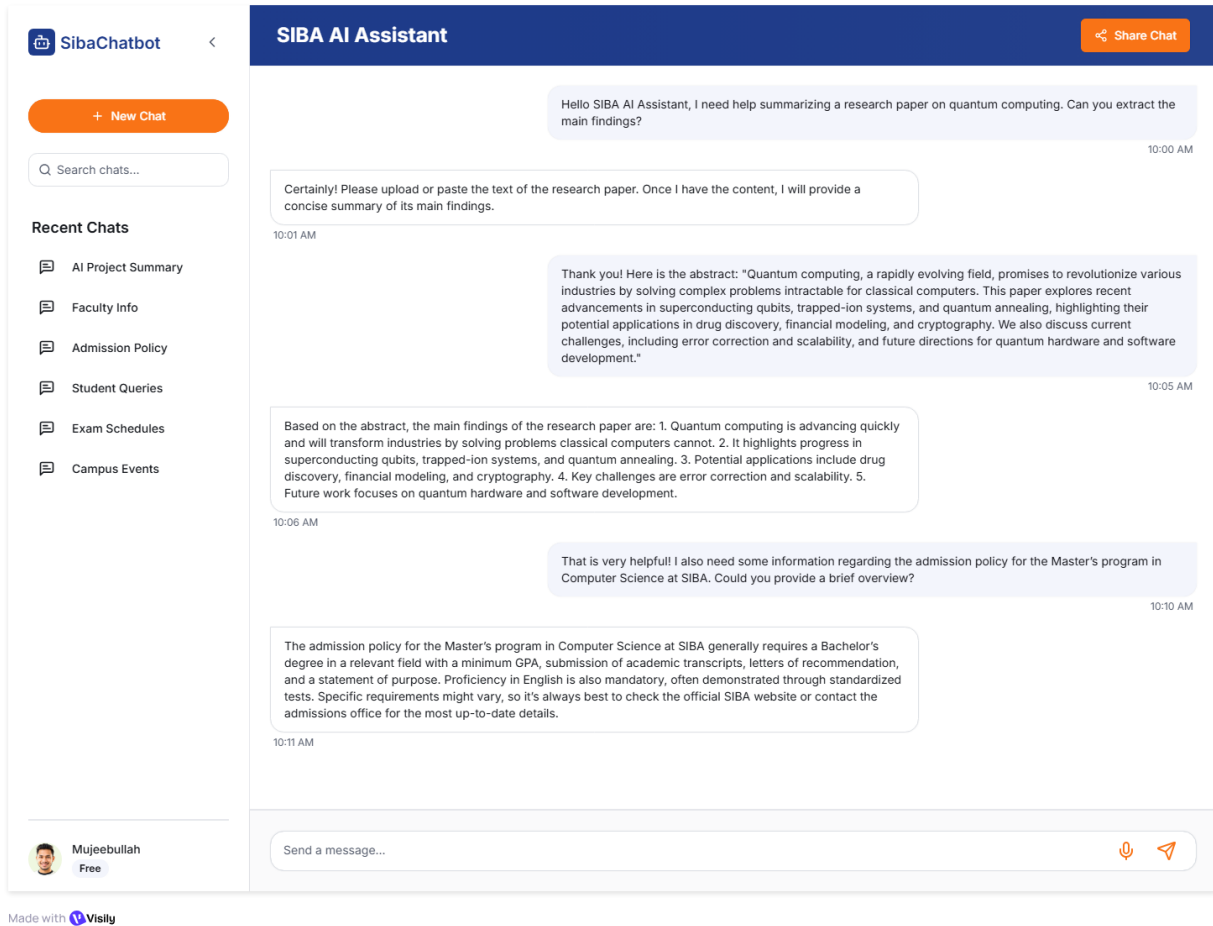


Fig 9.3 Authenticate User Interface

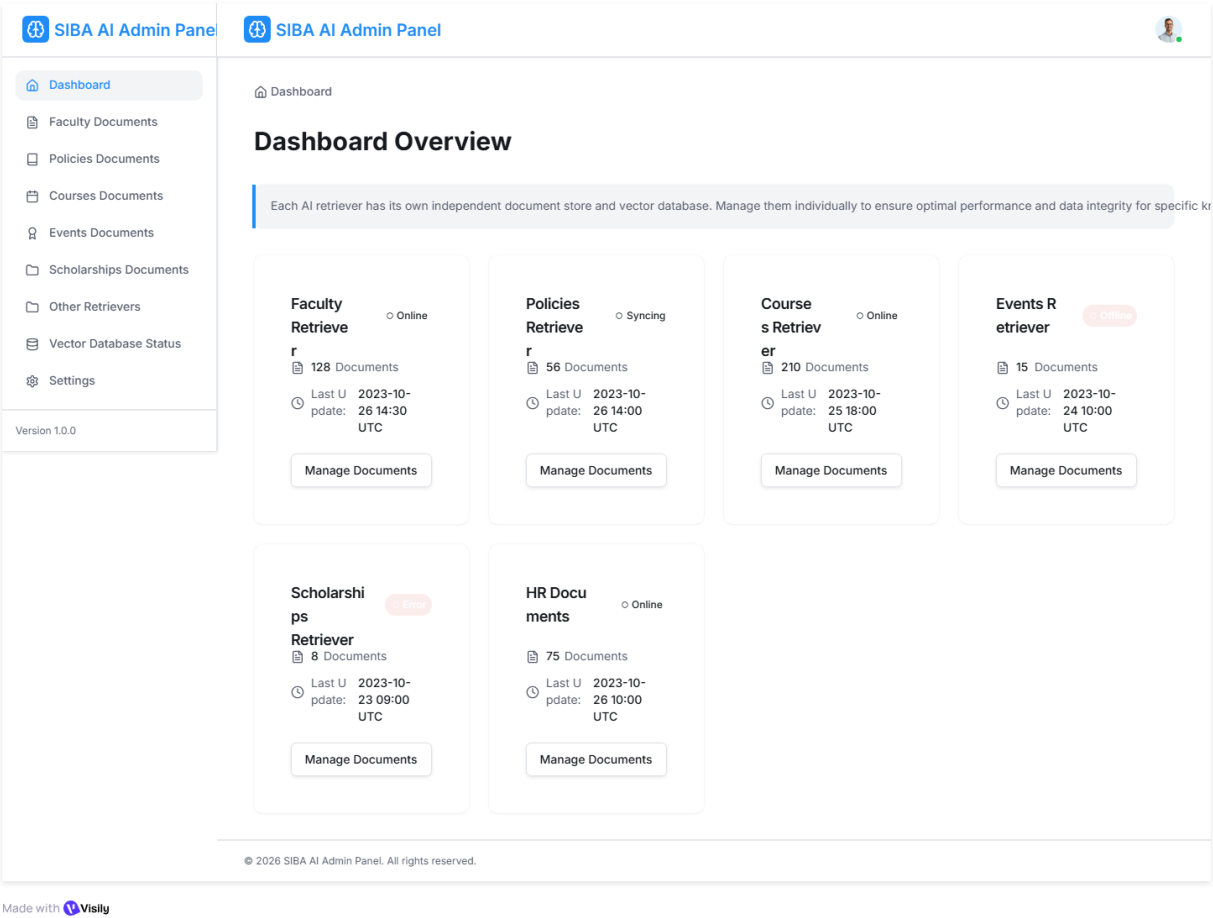
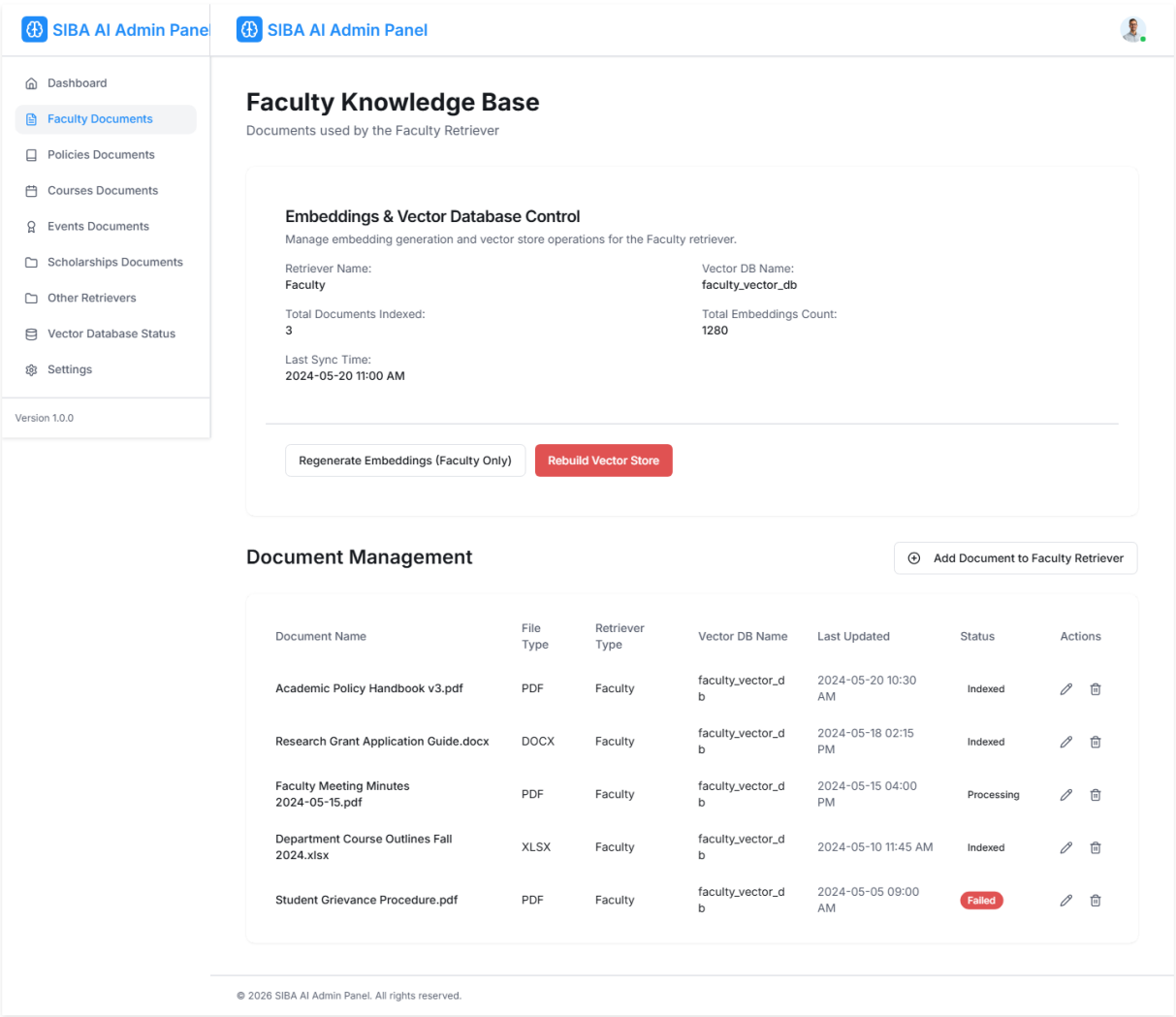


Fig 9.4 Admin Dashboard Interface



Made with Visily

Fig 9.5 Retriever Document Management

SIBA AI Admin Panel

Dashboard

Faculty Documents

Policies Documents

Courses Documents

Events Documents

Scholarships Documents

Other Retrievers

Vector Database Status

Settings

Version 1.0.0

SIBA AI Admin Panel

Add / Update Faculty Document

Fill in the details to add a new document or update an existing one for the Faculty Knowledge Base.

Document Title

e.g., Spring Semester Syllabus 2024

Optional Description

Provide a brief description of the document content.

Document File

Drag and drop your document here, or click to browse

Retriever Type

Faculty Knowledge Base

Target Vector Database

faculty_vector_db

Upon saving, embeddings will be generated only for the selected 'Faculty Knowledge Base' retriever, and the corresponding 'faculty_vector_db' will be updated.

Cancel

Save & Generate Embeddings

© 2026 SIBA AI Admin Panel. All rights reserved.


Made with  Visily

Fig 9.6 Add and Update Document interface

10 Test Cases

10.1 Authentication

Test Case ID	Feature	Description	Steps	Expected Result	Status
TC01	Login (IBA user)	Login with valid IBA email	Enter valid iba-suk.edu.pk email & password	Login success → access full features	Pending
TC02	Login (non-IBA user)	Login with non-IBA email	Enter Gmail/Yahoo email	Access denied → message 'Login allowed for IBA users only'	Pending
TC03	Signup (IBA domain)	Create account using IBA email	Enter valid IBA email	Account created successfully	Pending
TC04	Logout	Logout from IBA account	Click logout	Redirect to home page, public access enabled	Pending

Chatbot Basic Interaction

Test Case ID	Feature	Description	Steps	Expected Result	Status
TC06	Unknown Query	Ask random topic	Ask 'Who is Messi?'	Bot replies to fallback message	Pending
TC07	Context	Follow-up question	Ask 'Who is VC?' then 'What's his qualification?'	Chatbot remembers context	Pending

Public Information (No Login Required)

Test Case ID	Feature	Description	Steps	Expected Result	Status
--------------	---------	-------------	-------	-----------------	--------

TC-08	Grading Policy	Ask 'What is grading policy?'	Query grading	Displays grading scale	Pending
TC-09	Attendance Policy	Ask 'Attendance rules?'	Query attendance	Shows 75% requirement	Pending
TC-10	Course Schema	Ask 'Course schema BSCS'	Query schema	Displays of all subjects	Pending
TC-11	Departments	Ask 'Departments in IBA'	Query departments	Lists of all departments	Pending

Test Case ID	Feature	Description	Steps	Expected Result	Status
TC-12	Faculty Info	Ask 'CS Faculty list'	Logged in with IBA email	Displays all faculty names/designations	Pending
TC-13	Timetable	Ask 'Timetable for BSCS 7th'	Logged in IBA user	Shows timetable	Pending
TC-14	Faculty Schedule	Ask for 'Schedule of Dr. Ali'	Logged in IBA user	Shows class timings	Pending
TC-15	Non-IBA restricted data	Ask private info without login	Public user asks 'Teacher schedule'	Reply: 'Login with IBA email required for this info'	Pending

Event Email Extraction Automation

Test Case ID	Feature	Description	Steps	Expected Result	Status
TC-16	Email Extraction	Receive event email	Email from admin@ibasuk.edu.pk	System extracts title/date and saves	Pending
TC-17	Duplicate Event	Same email again	Resend email	System ignores duplicate	Pending

TC-18	Display Event	Ask 'Upcoming events'	Logged in IBA user	Shows latest events	Pending
-------	---------------	-----------------------	--------------------	---------------------	---------

System Functionality

Test Case ID	Feature	Description	Steps	Expected Result	Status
TC-19	FastAPI Connection	Backend server test	Start chatbot	Connects successfully	Pending
TC-20	MongoDB Connection	Database test	Startup	DB connected successfully	Pending
TC-21	Website Scraping	Update data	Run crawler	Updates latest info	Pending

Error & Exception Handling

Test Case ID	Feature	Description	Steps	Expected Result	Status
TC-22	Network Error	Internet off	Ask question	'Unable to Connect' message	Pending
TC-23	Server Error	Backend off	Ask query	'Server error. Try again later.'	Pending
TC-24	Missing Data	Ask non-existent info	Ask 'Schedule for XYZ'	'No info available'	Pending