

## 1 About Gujarat Vidyapith

Gujarat Vidyapith is a renowned educational institution founded by **Mahatma Gandhi** on **18 October 1920** during the Indian freedom movement. Established with the vision of promoting self-reliance, rural upliftment and value-based education, the university continues to follow Gandhian principles as its core foundation. In 1963, it was recognized as a **Deemed University** under Section 3 of the UGC Act, 1956.

The university offers a wide range of academic programs, from undergraduate and postgraduate courses to M.Phil. and Ph.D. research programs. Its disciplines include humanities, social sciences, education, science, computer science, rural management, journalism, library science and vocational studies. Gujarat Vidyapith emphasizes constructive work, community service and character-building as part of its holistic education system.

With its central campus located in Ahmedabad, along with several regional centers and affiliated institutes, Gujarat Vidyapith provides an environment where academic learning is combined with social responsibility. The university also encourages students to engage in rural camps, extension activities and social awareness programs, aligning with Gandhiji's ideals of Nai Talim and community-oriented learning.

The Department of Computer Science at Gujarat Vidyapith is committed to providing quality education in the field of computing and information technology. The department aims to develop skilled professionals who can apply technical knowledge to real-world challenges while upholding the values and educational philosophy of the institution.

The department offers a range of academic programs, including **BCA**, **MCA**, **MSC(IT)** and **PGDCA** courses designed to equip students with strong theoretical foundations and hands-on practical experience. The curriculum covers key areas such as programming, data structures, databases, operating systems, networking, artificial intelligence, machine learning and web technologies.

The Department of Computer Science maintains a strong relationship with industry and academic partners, helping students gain professional insights and career guidance. Its focus on both technical competency and value-based education prepares students to contribute effectively as IT professionals, researchers and responsible members of society.

## 2 Introduction

With the increasing use of digital technologies in educational institutions, there is a growing need for systems that can provide quick, reliable and accessible information to students and visitors. University enquiry processes often depend on physical visits, office hours and static website content, which can lead to delays, confusion and repeated manual work for administrative staff. To address these challenges, the **Gujarat Vidyapith Chatbot** has been developed as an interactive, intelligent and user-friendly solution.

The chatbot functions as a virtual assistant that provides instant responses to queries related to admissions, courses, departments, fees, hostel facilities and general university information. By using Natural Language Processing (NLP) and a trained machine learning model, the system can understand user queries, classify their intent and generate appropriate responses within seconds. This not only reduces dependency on manual enquiries but also ensures 24/7 accessibility, allowing users to obtain information anytime, from anywhere.

The system is built using **Django** for the backend, **PyTorch** for model inference and **MongoDB** for data storage. The architecture is designed to be scalable, enabling future improvements such as multilingual support, voice-enabled interaction and integration with live human assistance. Through a conversational interface, the chatbot enhances user experience while reducing workload for university staff by handling repetitive questions efficiently.

Overall, the Gujarat Vidyapith Chatbot represents a modern approach to university information management. It bridges the gap between users and institutional resources by offering fast, clear and structured responses, making university information more accessible and improving the overall communication process within the institution.

## 3 Existing System

Most universities still depend on simple, traditional methods to share information with students and parents. These methods work, but they are slow and not convenient.

### 1. Physical Enquiry

Students often have to visit the campus to ask about admissions, fees, courses, or facilities. This takes time and is especially difficult for people coming from other cities or villages.

During a physical visit, the user approaches the enquiry counter and explains the question to a staff member. The staff member then checks printed documents, internal files or departmental information to provide the required details. In some cases, the enquiry is redirected to another department, which increases the waiting time. After receiving the information, the user leaves, but if another doubt appears later, the same process must be repeated again.

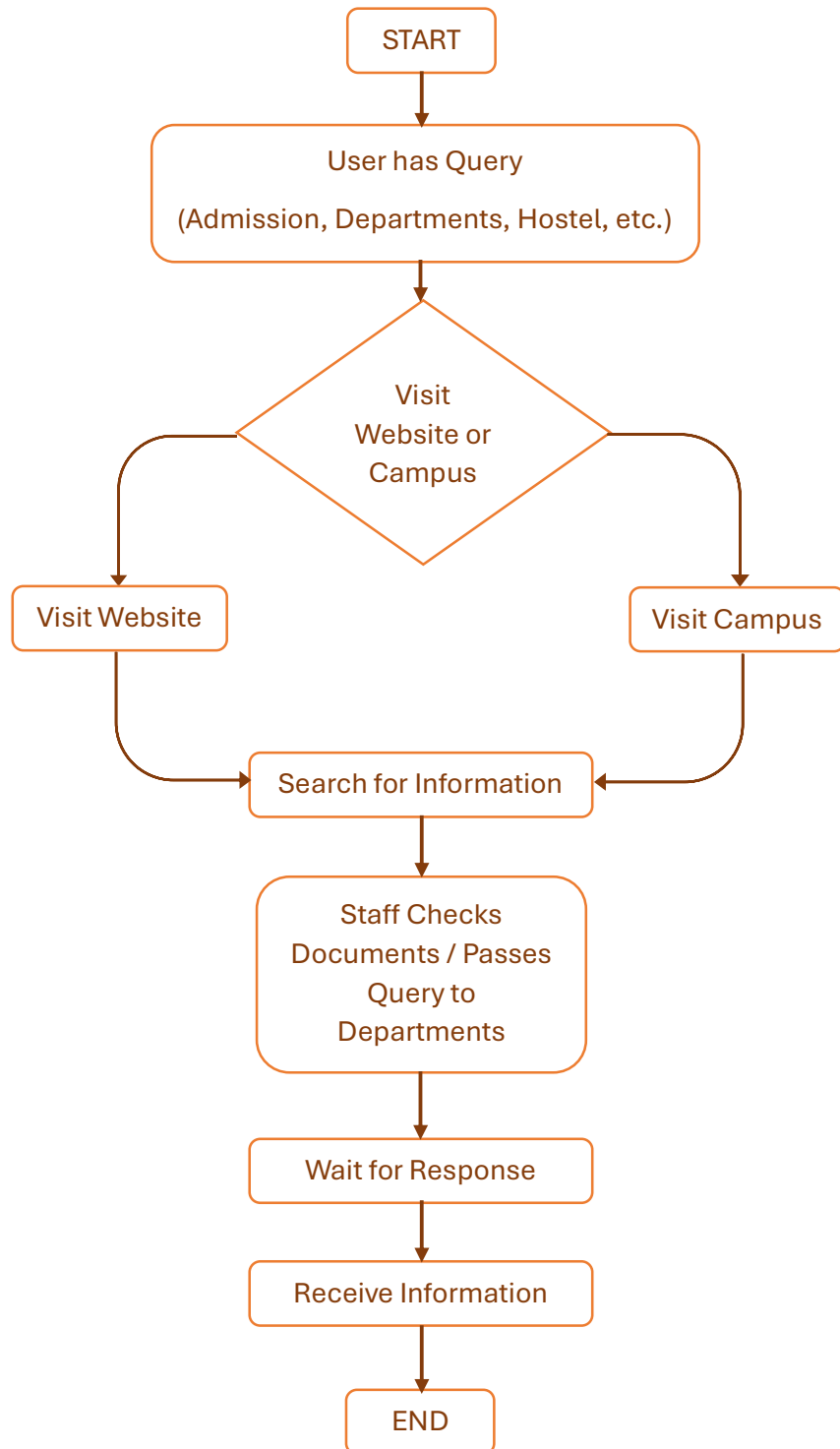
### 2. University Website

Websites provide general information, but the details are scattered across many pages. Users must search manually, and the site cannot answer follow-up questions or give instant clarification.

### Limitations of the Current System

- Physical visits take extra time and effort.
- Websites show only static information.
- No interactive guidance for specific doubts.
- Students depend on staff availability.
- Staff handle repetitive questions during admission season.

## 3.1 Existing System Flow Chart



## 4 Existing System Problem Areas

### **Time-Consuming Information Access**

Students and parents often need to visit the campus or make repeated phone calls just to get basic information. This leads to unnecessary delays, especially during admission periods.

### **Static and Scattered Website Information**

University websites contain information spread across multiple pages and PDF files. Users struggle to find exact details, and the website cannot answer personalized or follow-up questions.

### **No Instant Support**

The existing system depends on office timings. Students cannot get answers during evenings, weekends or holidays, which slows down decision-making.

### **High Workload on University Staff**

Staff members handle the same set of questions repeatedly during admission season. This increases manual effort and reduces the time available for important administrative tasks.

### **Limited Accessibility for Remote Users**

Students from distant locations face difficulty in accessing timely and accurate information. Phone lines may remain busy, and physical visits are not always possible.

### **Lack of Personalized Guidance**

The existing system provides only general information. It cannot adjust answers based on the exact query or provide clarification when the user has additional doubts.

### **Risk of Miscommunication**

Since responses depend on different staff members, there is a chance of inconsistent or incomplete information being shared.

### **No Mechanism for Handling Repetitive Queries Efficiently**

The system has no automation to filter and respond to common questions, leading to repeated manual effort and longer waiting times.

## 5 System Analysis

### Existing System Analysis

#### **Dependence on Traditional Methods**

Students rely mainly on physical enquiry counters and the university website to obtain information, making the process slow and inconvenient.

#### **Lack of Interactivity**

The existing system provides static information. It cannot answer personalized questions or guide users through follow-up queries.

#### **Not Available 24/7**

Information can be accessed only during office hours. Students must wait during weekends, holidays and after closing time to get answers.

#### **Repetitive Manual Work for Staff**

Staff members must respond to the same common questions repeatedly, especially during admission season. This increases workload and reduces efficiency.

#### **Accessibility Challenges**

Students from remote areas or those who cannot visit the campus face difficulty getting timely information. Phone lines may be busy, and emails take time to receive responses.

#### **Higher Chance of Miscommunication**

Since information is communicated manually by different staff members, there is a risk of inconsistent or incomplete details being provided.

### Proposed System Analysis

#### **Introduction of Chatbot as Virtual Assistant**

The proposed system replaces manual enquiry with an intelligent chatbot that answers student queries instantly through a web interface.

#### **Interactive and Conversational Responses**

The chatbot uses Natural Language Processing (NLP) to understand user queries and provide accurate, structured answers in real time.

#### **24/7 Availability**

Users can get information anytime, without waiting for office hours or staff availability. This improves convenience and reduces delays.

#### **Reduced Staff Workload**

By handling repetitive and common questions automatically, the chatbot allows university staff to focus on important administrative tasks.

**Faster Decision-Making for Students**

Instant access to information helps students make quicker decisions about courses, admissions, fees and other academic matters.

**Better Accessibility for Remote Users**

Students from any location can access information through the chatbot, making the system more inclusive and supportive.

**Structured and Consistent Information**

The chatbot uses predefined responses stored in a database or JSON file, ensuring accurate and uniform information for all users.

**Scalability for Future Enhancements**

The system can be extended to support voice queries, multilingual interaction, or integration with human support in future versions.

## 5.1 Objectives to Be Fulfilled

The main objective of the proposed university chatbot system is to simplify the way students and parents access important information. Instead of depending on physical enquiry counters or navigating long website pages, the chatbot provides a fast and interactive method to get accurate answers. The system aims to improve accessibility, reduce staff workload and offer a modern information experience for users from any location.

### **Provide Instant Information**

Deliver quick, reliable answers to common queries related to admissions, courses, fees, hostel facilities, departments and campus life. The goal is to reduce the time students spend searching for information manually.

### **Reduce Physical Enquiries**

Minimize the need for students and parents to visit the campus for basic questions. This saves travel time, reduces crowding during admission season and improves convenience for users from remote areas.

### **Lower Staff Workload**

Automate repeated and routine questions so university staff can focus on more important academic and administrative tasks. This helps improve efficiency within departments.

### **Ensure 24/7 Availability**

Provide information at any time of the day, including weekends and holidays. The chatbot serves as an always-available source of guidance, unlike traditional enquiry counters.

### **Improve Accessibility for All Users**

Support students from rural and urban areas equally by making information accessible with just a message. No special technical knowledge is required to use the system.

### **Offer Consistent and Accurate Responses**

Use predefined structured data and official sources to ensure every user receives the same correct and up-to-date information, avoiding confusion and miscommunication.

### **Support Future Expansion**

Build the system in a scalable way so features like multilingual support, voice interaction or integration with human assistance can be added easily in future versions.



## 5.2 Scope of Proposed System

The proposed system is a **university information chatbot** designed to provide instant and accurate answers to student and parent queries. Instead of relying only on physical visits or static website content, the chatbot will act as a virtual assistant that can be accessed through a web or mobile platform.

The chatbot will handle questions related to admissions, courses, fees, hostel facilities, deadlines, and other university-related information. By using Natural Language Processing (NLP) and a structured database, the system will identify the intent of the query and deliver a suitable response within seconds.

### In Scope

- Providing information about admission procedures, deadlines, and eligibility.
- Answering queries about different courses, duration, and fees.
- Offering details of hostel facilities and other student services.
- Allowing users to interact in a conversational manner through a web-based chatbot interface.
- Ensuring availability of information 24/7.
- Reducing repetitive workload for university staff by automating common queries.

### Out of Scope (Future Enhancements)

- Voice-enabled queries and responses.
- Multilingual support for non-English users.
- Integration with live human agents for complex queries.
- Advanced AI features such as personalized recommendations.

### Overall Scope

The chatbot focuses on delivering accessible, structured and reliable information to users while reducing repetitive workload on university staff. It stays within informational boundaries and always directs users to the correct official documents or department pages for confirmation. Its scope balances helpful guidance with clear limitations, ensuring a smooth and dependable information experience.

## 5.3 System Features

The proposed university chatbot system is designed to make information access simple, quick and reliable for students, parents and visitors. It uses a conversational interface to answer common queries instantly, reducing the need for manual enquiry counters. The system focuses on providing accurate and structured information through predefined responses, improving the user experience and reducing repetitive workload for staff.

### **Interactive Chat Interface**

The chatbot offers a simple and user-friendly interface where users can type questions naturally. It responds in conversational language and guides users to accurate information without requiring them to navigate multiple web pages or documents.

### **Admission Information**

The system provides clear details about admission procedures, eligibility criteria, application steps, required documents and important deadlines. It also directs users to the official admission prospectus or website for complete and updated information.

### **Course and Program Details**

Users can ask about available UG and PG programs, syllabus outlines, department offerings and academic structure. The chatbot gives short summaries and links to department or syllabus pages where needed.

### **Fee Structure Guidance**

Instead of displaying outdated numbers, the chatbot redirects users to the official fee matrix or prospectus. This ensures accuracy and prevents confusion due to frequently changing fee details.

### **Hostel and Facility Information**

The system supports queries about hostel rules, allocation process, accommodation details and campus facilities. It guides users to official documents for hostel fees and application requirements.

### **Instant Query Resolution**

By processing user queries through predefined intents and responses, the chatbot answers questions within seconds. This reduces waiting time and allows students to get information quickly during peak admission periods.

### **Instant Query Resolution**

By processing user queries through predefined intents and responses, the chatbot answers questions within seconds. This reduces waiting time and allows students to get information quickly during peak admission periods.

## **NLP-Based Accuracy**

Responses are generated from structured data sources, ensuring consistent and reliable answers. The underlying NLP model helps classify user intent and select the best possible response.

## **Reduced Staff Workload**

By automating repetitive and commonly asked questions, the system allows staff to focus on more important administrative tasks. This improves efficiency during admission seasons.

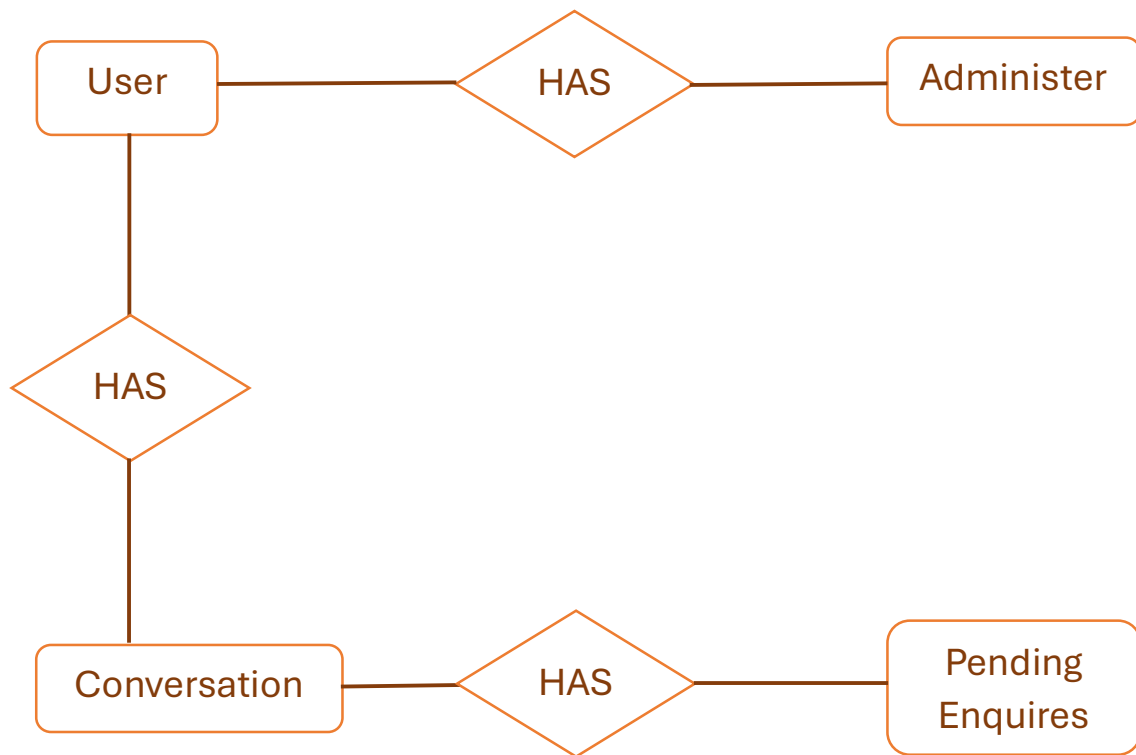
## **Fallback and Clarification Handling**

When the chatbot cannot understand a question, it asks for clarification or suggests related topics instead of giving incorrect information. This maintains accuracy and keeps the conversation meaningful.

## **Scalability and Future Expansion**

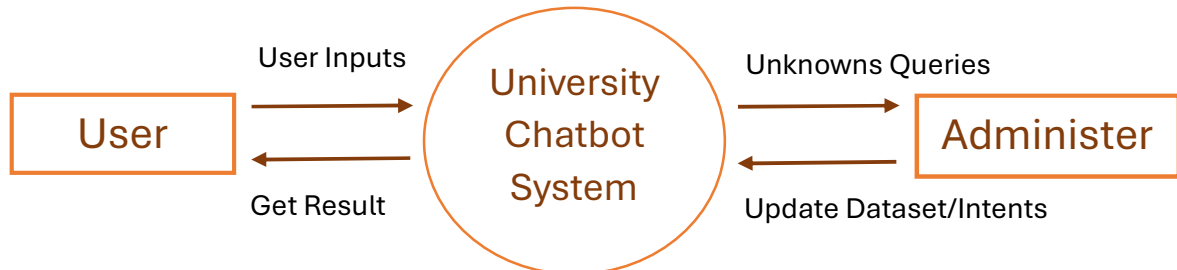
The system is designed to support future upgrades such as multilingual responses, voice interaction, integration with human support, and expanded academic modules. New intents and responses can be added easily as the university's needs grow.

### 5.3.1 E-R Diagram (ERD)

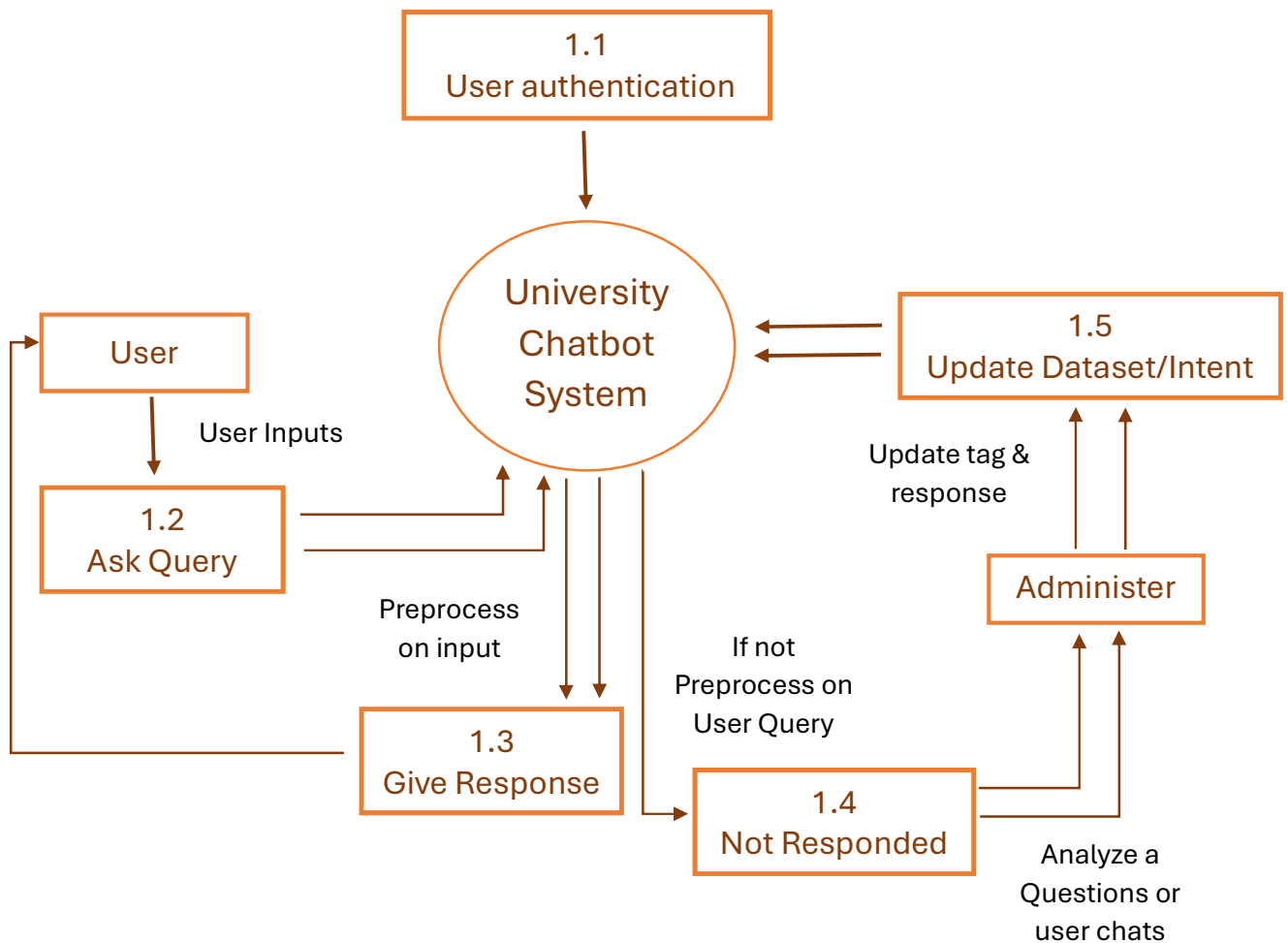


## 5.3.2 Data Flow Diagram (DFD)

LEVEL 0:



LEVEL 1:



## 5.3.3 Hardware & Software Environment

### **Hardware Requirements**

These are enough for development and testing:

- **Processor:** Dual-core (Intel i3 or equivalent)
- **RAM:** 4 GB
- **Storage:** 20 GB free space
- **Display:** Standard 1366×768 resolution
- **Input Devices:** Keyboard, Mouse
- **Network:** Basic internet connection for package installation and model training

### **Software Requirements**

#### **1. Operating System**

The project can run on any modern operating system that supports Python and MongoDB:

- Windows 10 / 11
- Linux (Ubuntu recommended for deployment)
- macOS (for development)

#### **2. Backend Technologies**

These tools form the core backend of the chatbot:

- **Python 3.8+** – Primary programming language.
- **Django Framework** – Used for building the web application, routing and server-side logic.
- **Djongo** – Connector that allows Django to work directly with MongoDB.
- **MongoEngine** – ODM for advanced MongoDB operations.

#### **3. Database**

The chatbot uses a NoSQL database for storing intents, sessions and responses:

- **MongoDB Community Server** (local or cloud)
- MongoDB Compass (optional GUI for administration)

## 4. Machine Learning & NLP Libraries

Libraries used for model training, inference and text processing:

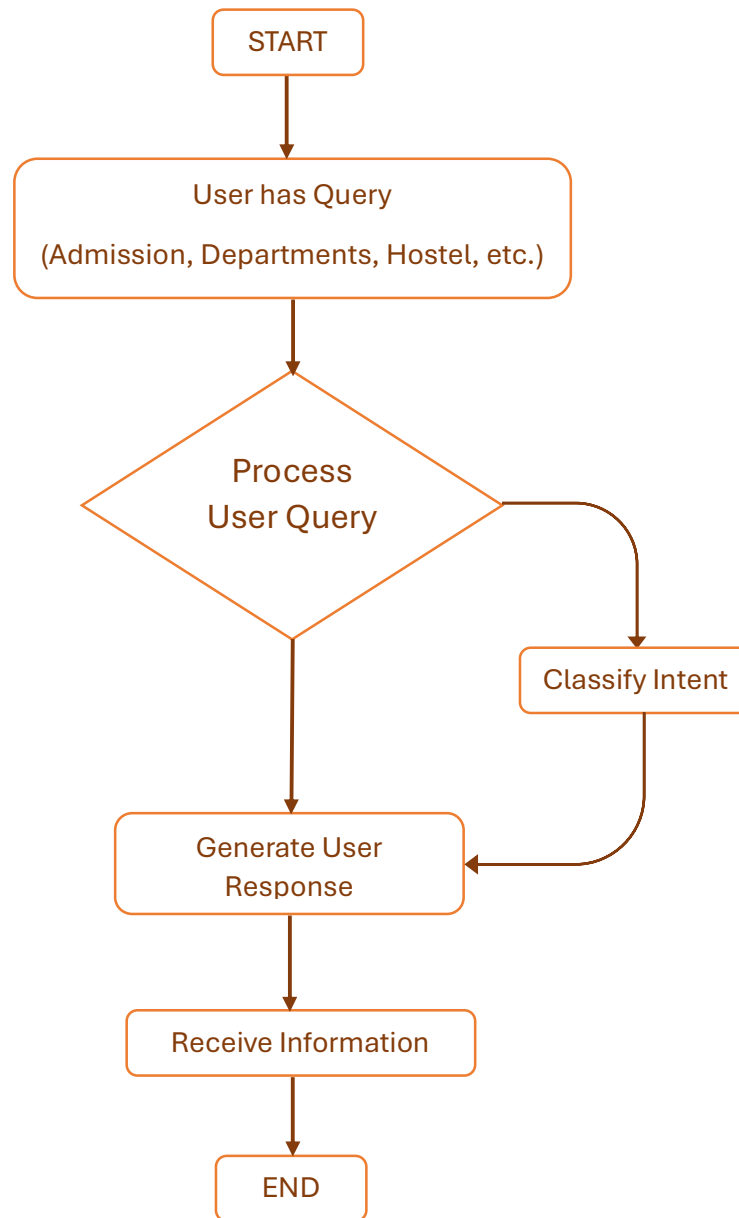
- **PyTorch (torch)** – For building and training the neural network model.
- **NLTK** – For tokenization and basic NLP preprocessing.
- **NumPy** – For vector and matrix operations during training.

## 5. Frontend Technologies

Used to design the chat interface and user experience:

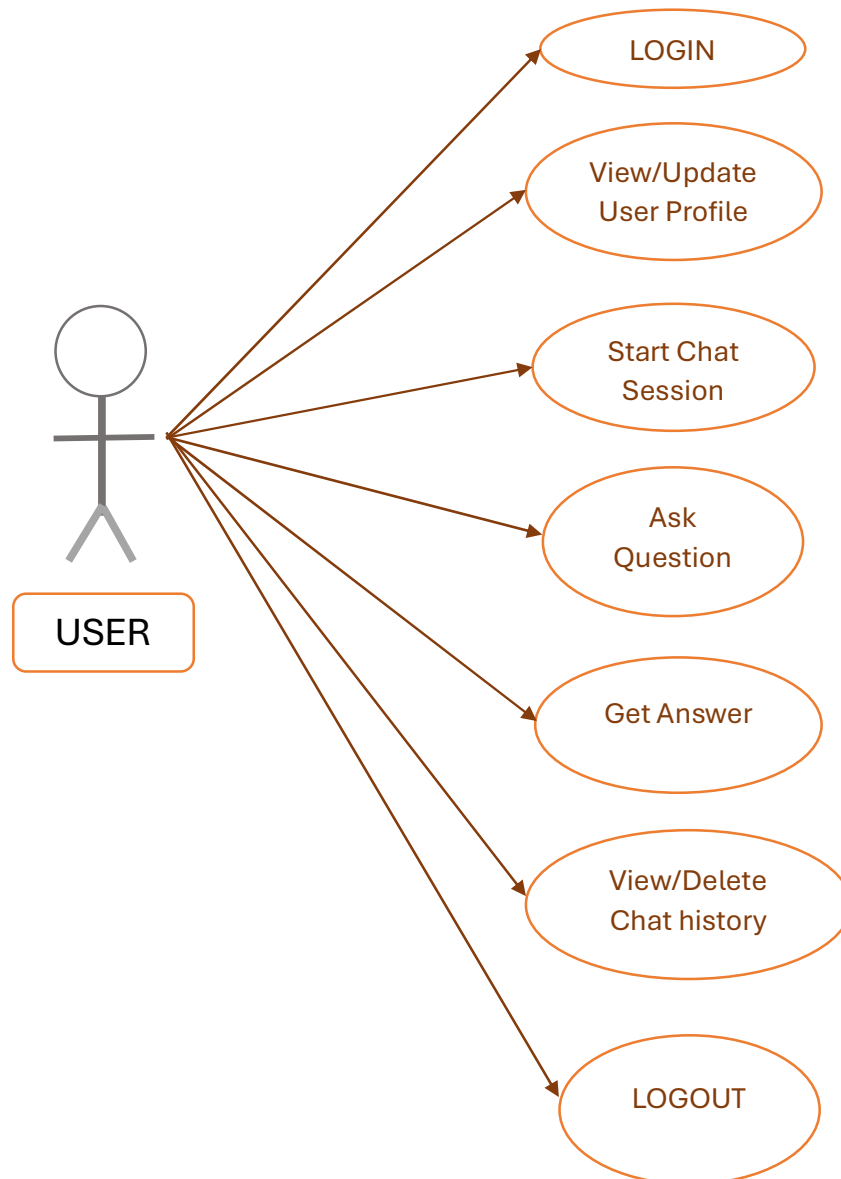
- **HTML5**
- **CSS3**
- **Bootstrap 5** (for layout and styling)
- **JavaScript (AJAX/FETCH)** for sending user queries to Django without reload.

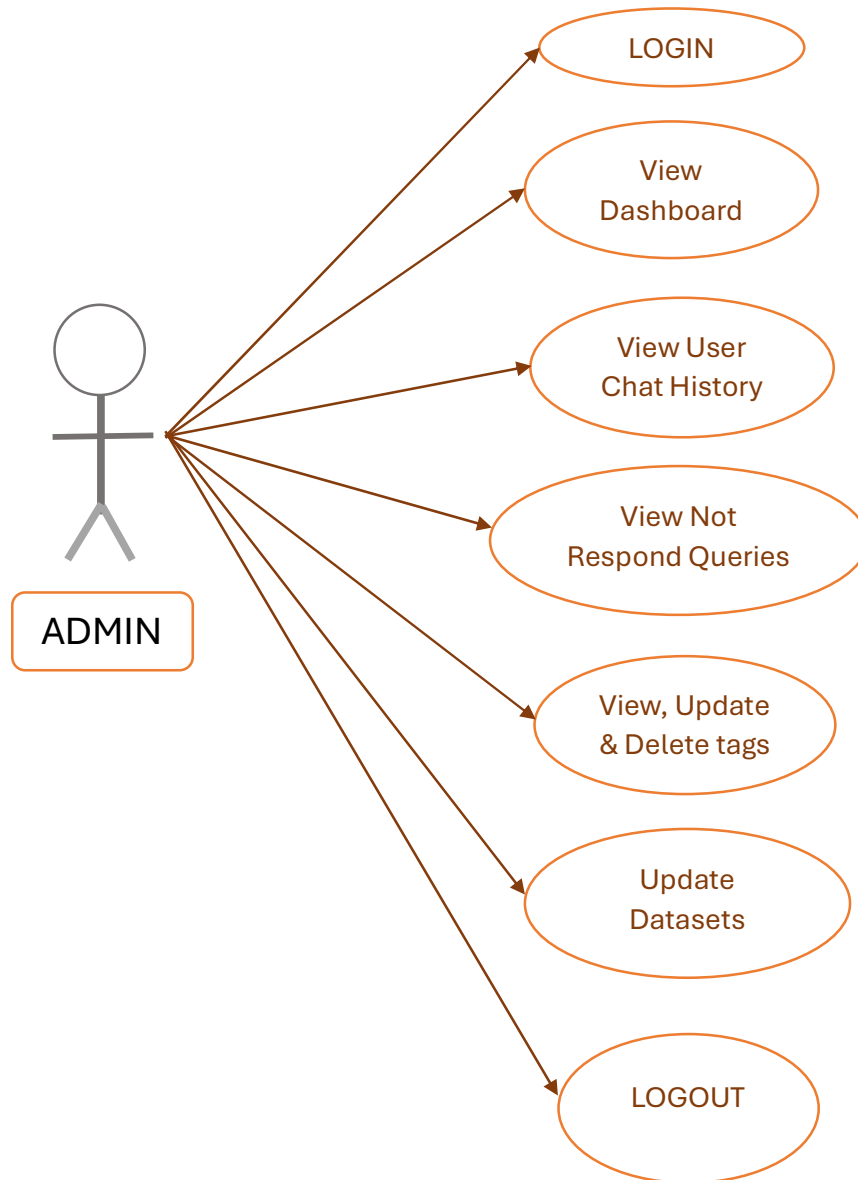
## 6 Chatbot System Flow Chart



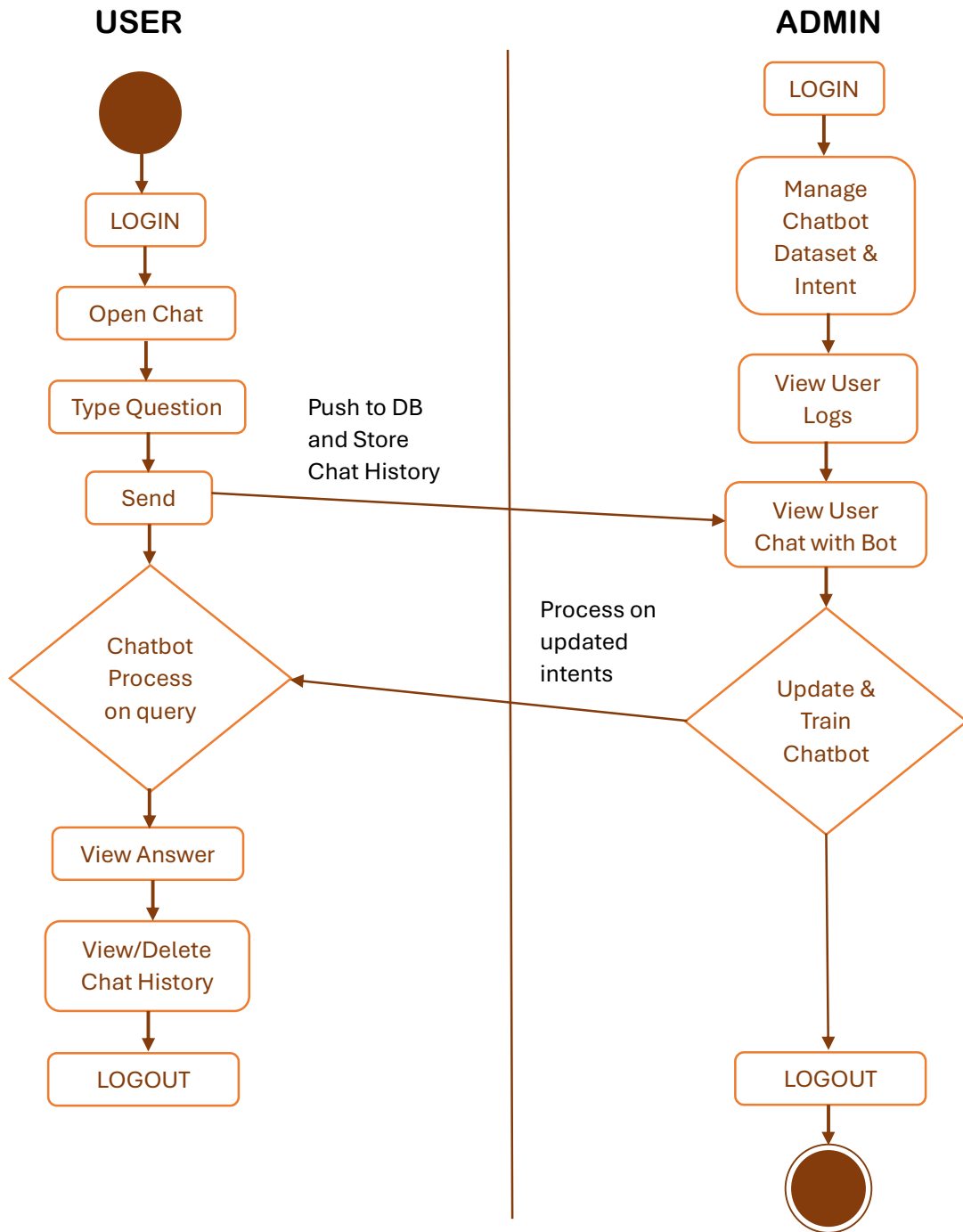


## 6.1 Use Case Diagram





## 6.2 Activity Diagram



## 7 Data Dictionary

### 1. User Table Structure (ChatBot\_user)

Column Name	Data Type	Description
<b>_id</b>	ObjectId	MongoDB internal ID
<b>id</b>	Integer	Auto-increment user ID
<b>full_name</b>	String	User's full name
<b>username</b>	String	Unique login username
<b>email</b>	String	User email address
<b>password</b>	String	Plain password (should be hashed in production)
<b>gender</b>	String	male / female
<b>role</b>	String	student / parent / faculty / staff / other
<b>profile_image</b>	String	Profile picture path
<b>joined_date</b>	DateTime	Registration timestamp
<b>status</b>	String	active / inactive

### 2. Admin User Table (ChatBot\_adminuser)

Column Name	Data Type	Description
<b>_id</b>	ObjectId	MongoDB auto ID
<b>username</b>	String	Admin username
<b>email</b>	String	Admin email
<b>password</b>	String	Admin password

### 3. Pending Questions Table (ChatBot\_pendingquestion)

Field Name	Data Type	Description
<b>_id</b>	ObjectId	MongoDB internal ID
<b>id</b>	Integer	Unique question ID
<b>user_id</b>	Integer	FK → <a href="#">User.id</a>
<b>username</b>	String	Username of the person who asked
<b>question_text</b>	String	User's actual question
<b>model_tag</b>	String	Chatbot's predicted tag
<b>confidence</b>	Float	Prediction confidence (0-1)
<b>created_at</b>	DateTime	When question was asked
<b>is_resolved</b>	Boolean	false = pending, true = handled
<b>admin_note</b>	String	Admin explanation / fix note

## 4. Conversation Table (ChatBot\_conversation)

Column Name	Type	Description
<b>_id</b>	ObjectId	Unique conversation ID
<b>id</b>	Integer	Conversation row ID
<b>user_id</b>	Integer	FK → <a href="#">User.id</a>
<b>message</b>	String	Chat message content
<b>is_bot</b>	Boolean	true = bot message, false = user
<b>created_at</b>	DateTime	Message timestamp

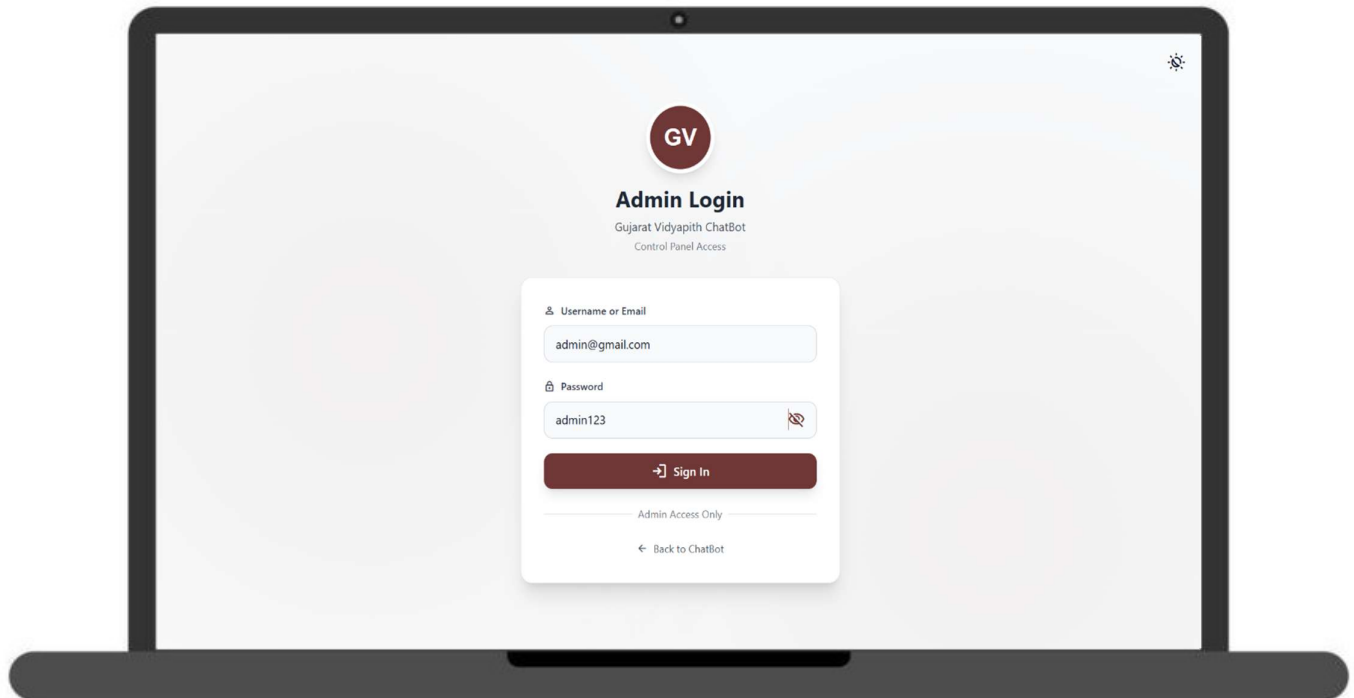
## 5. Chat Dataset (intents.json)

Field	Type	Description
<b>tag</b>	String	Intent identifier
<b>patterns</b>	Array	User input examples
<b>responses</b>	Array	Bot output responses

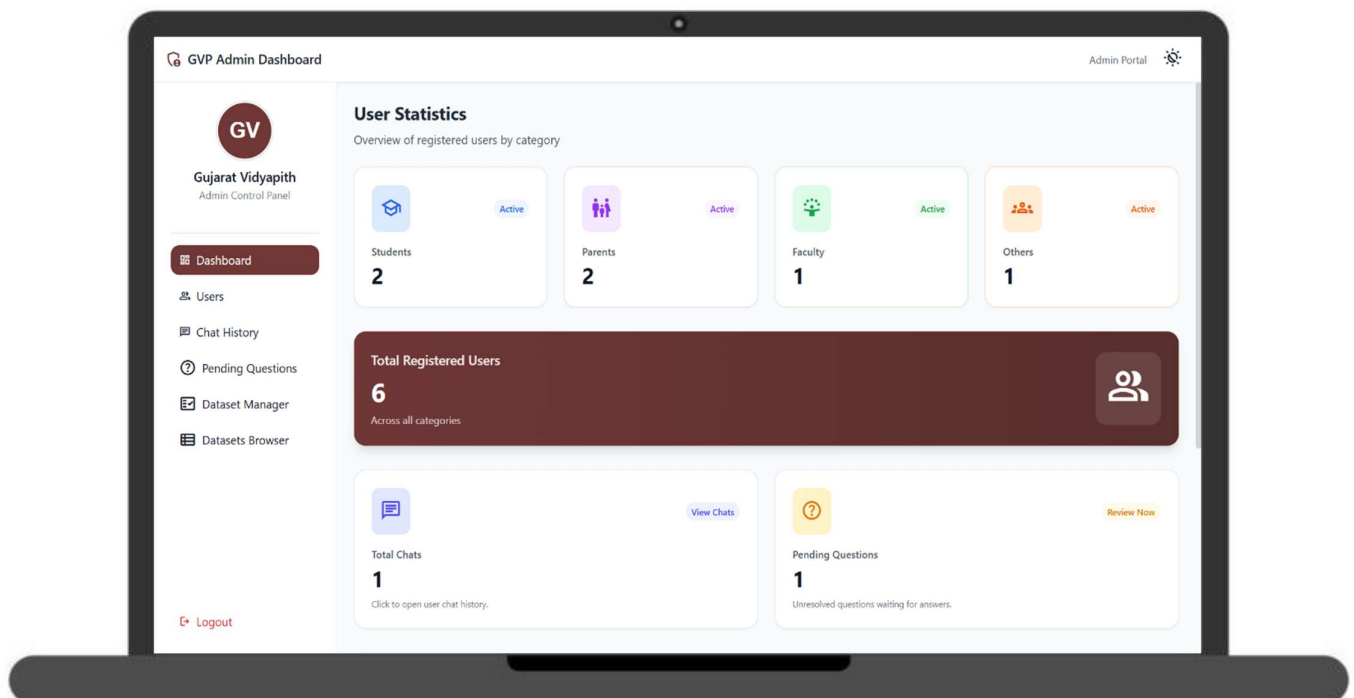
## 7.1 Ui Design

### + Admin Dashboard

#### 1. Admin Login

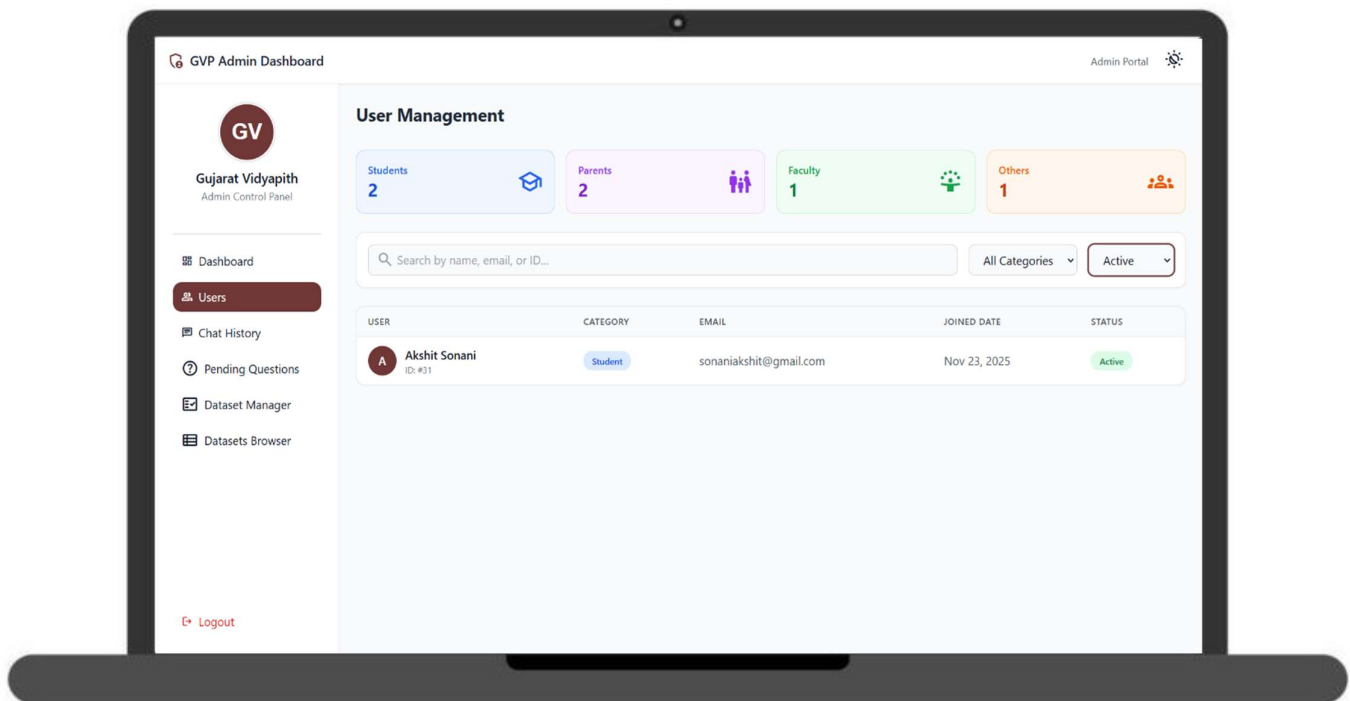
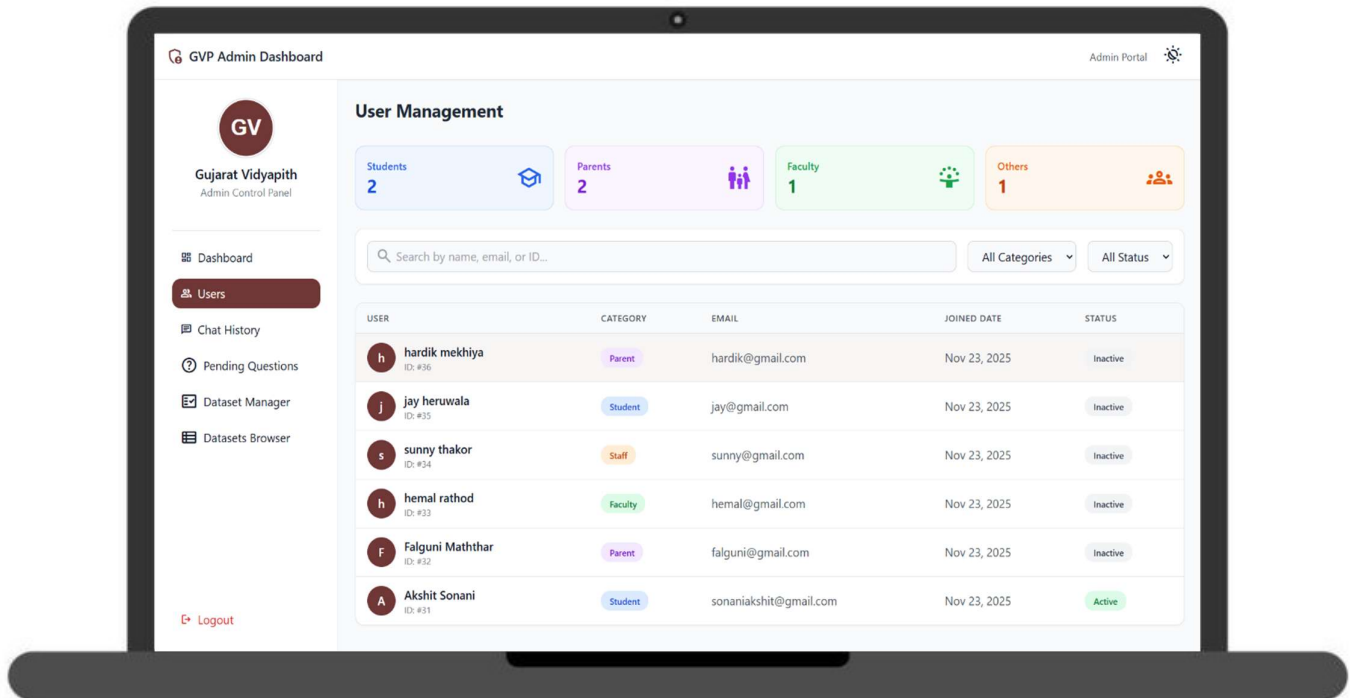


#### 2. Admin Dashboard & User Statistics

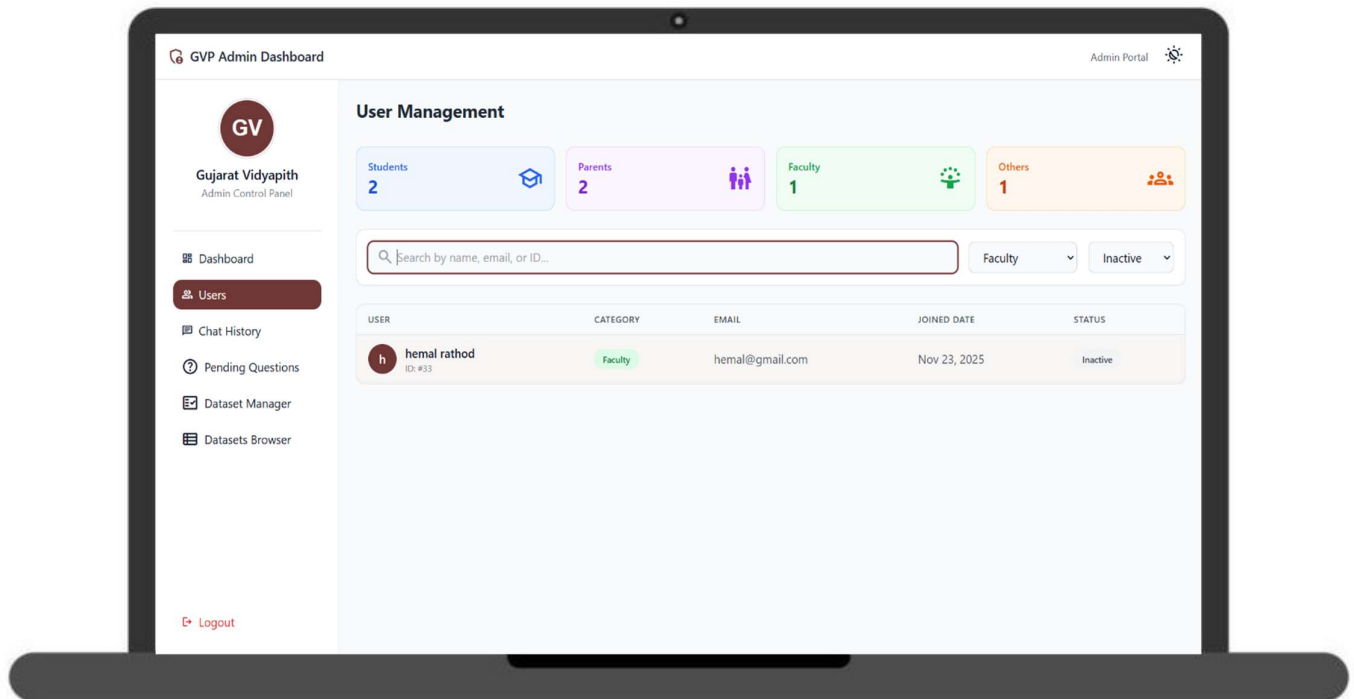


# Gujarat Vidyapith Chatbot

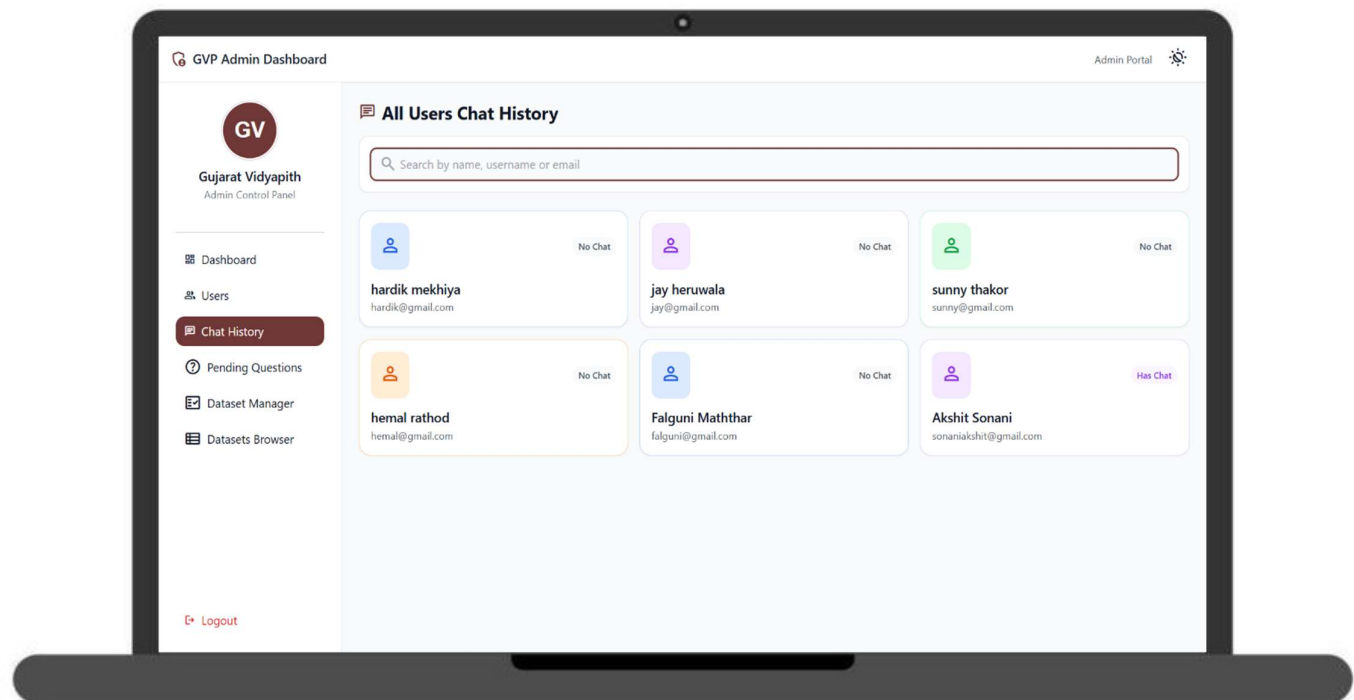
## 3. User Management



# Gujarat Vidyapith Chatbot

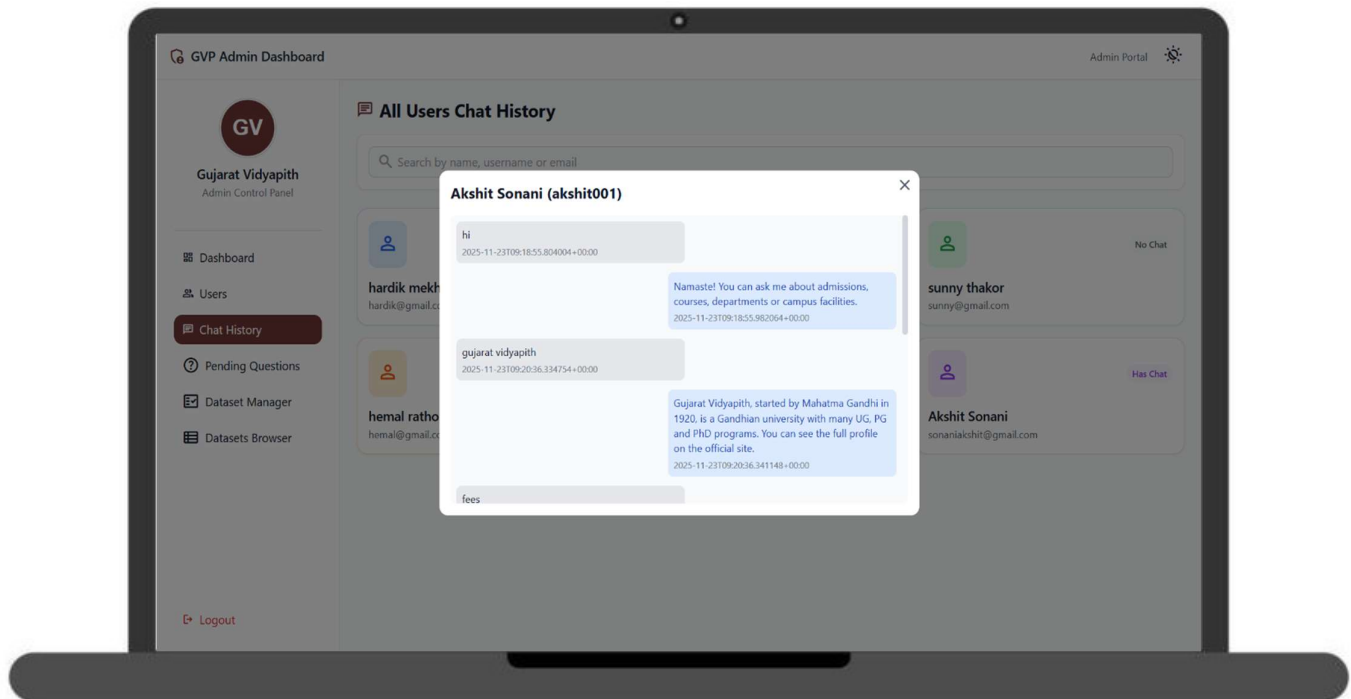


## 4. All Users Chat History

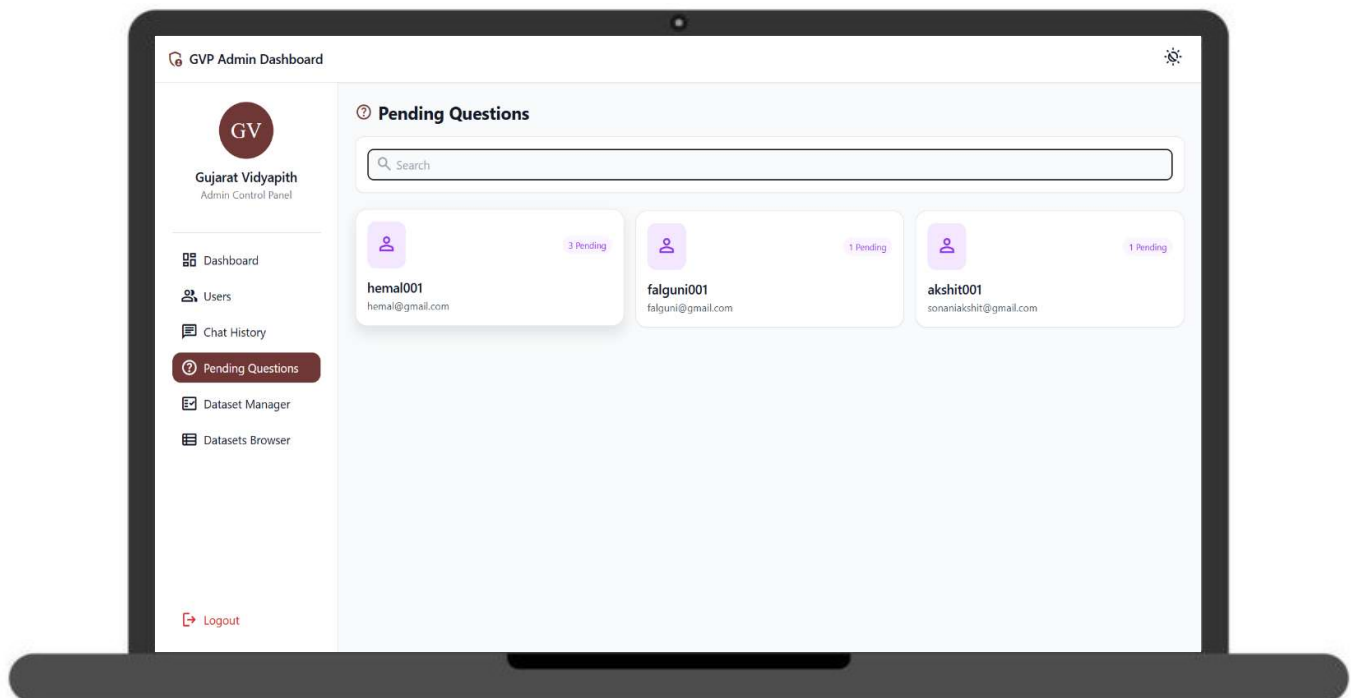




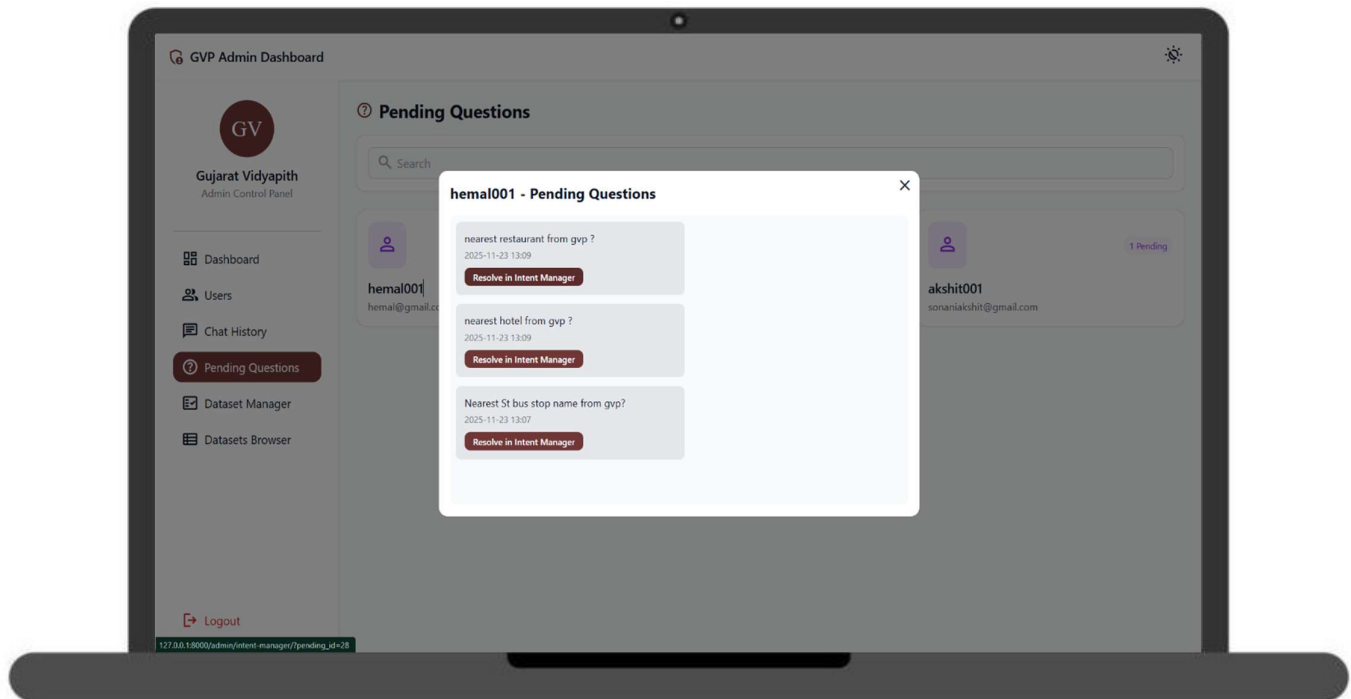
# Gujarat Vidyapith Chatbot



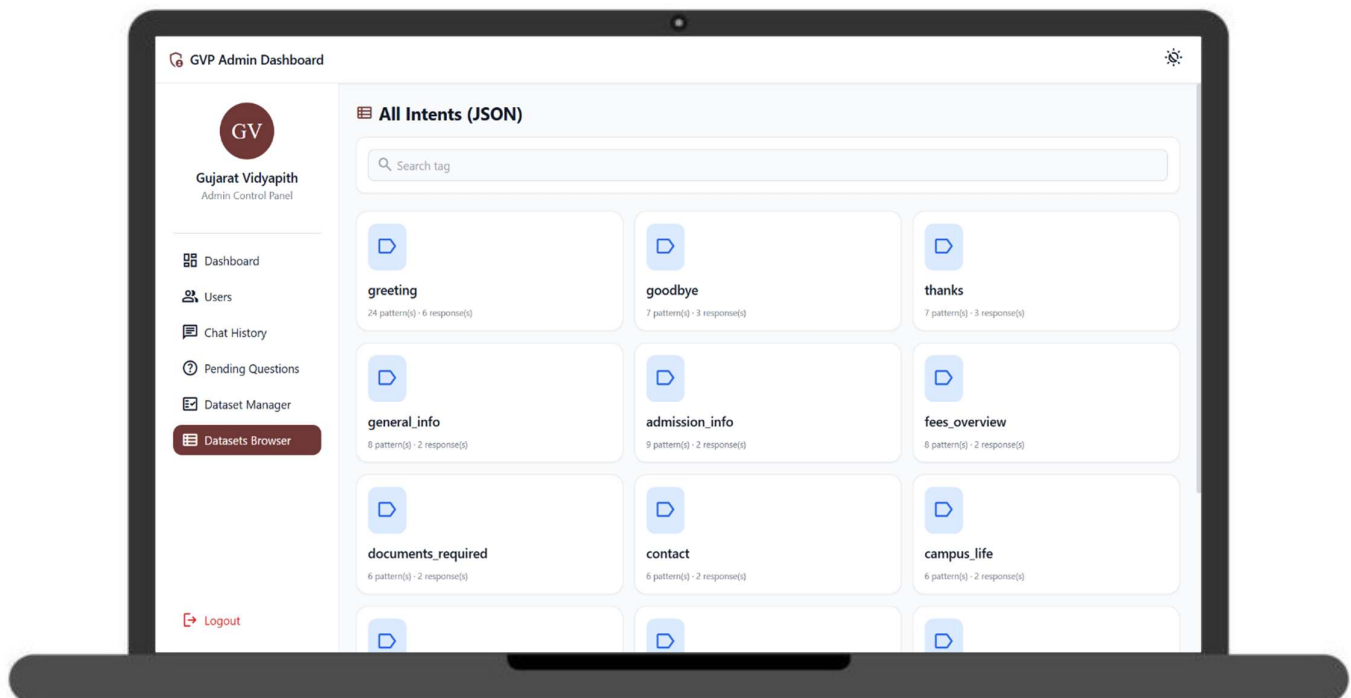
## 5. Pending Questions



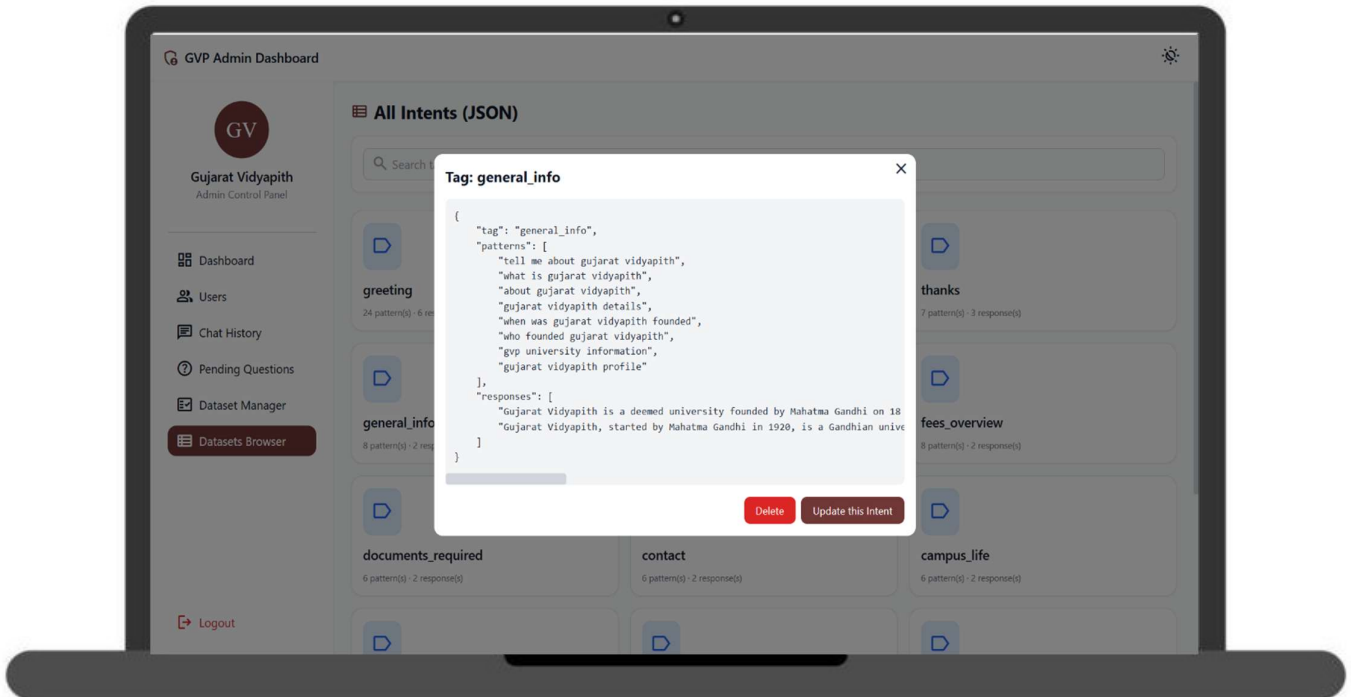
# Gujarat Vidyapith Chatbot



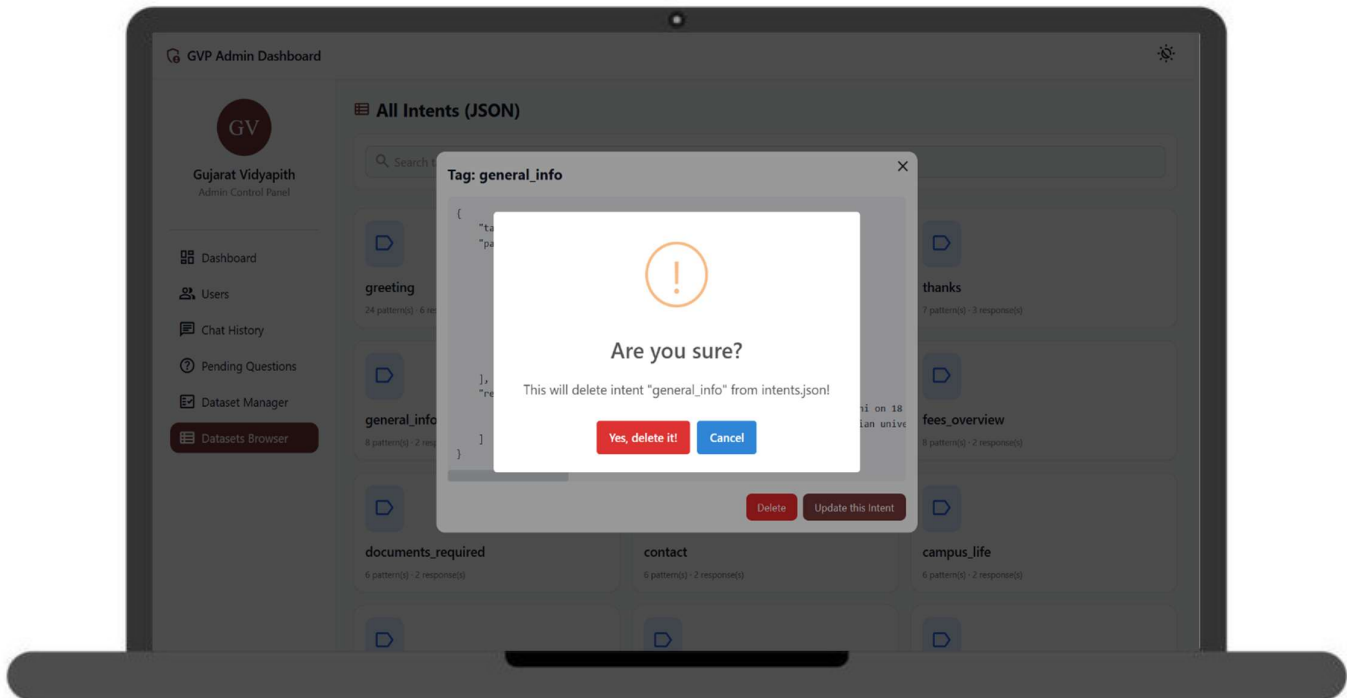
## 6. Datasets Browser



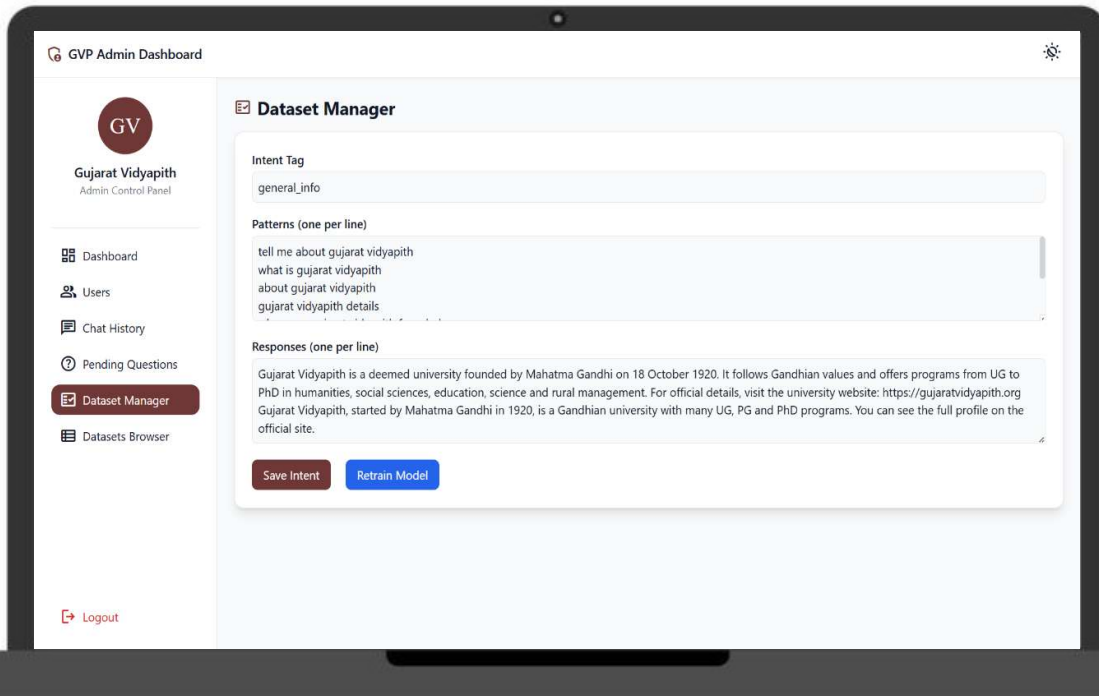
# Gujarat Vidyapith Chatbot



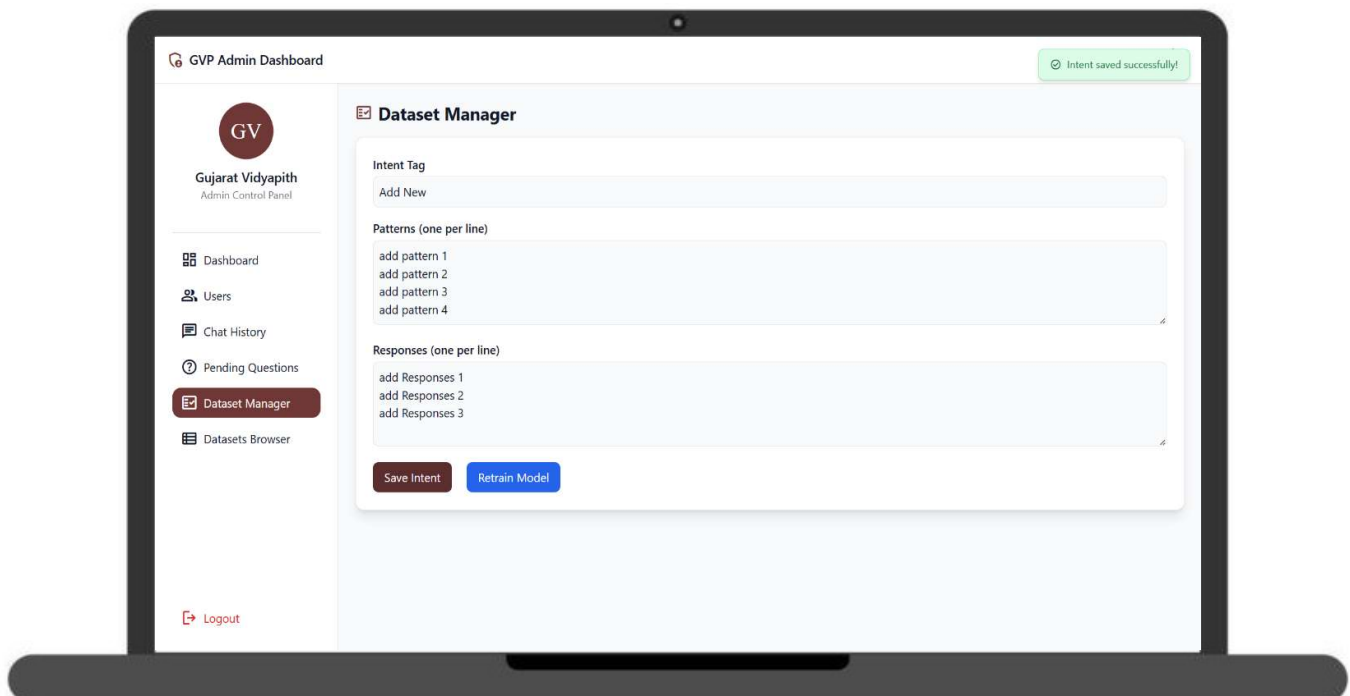
## 7. Delete above(general\_info) intent from JSON file



## 8. Update above(general\_info) Intent

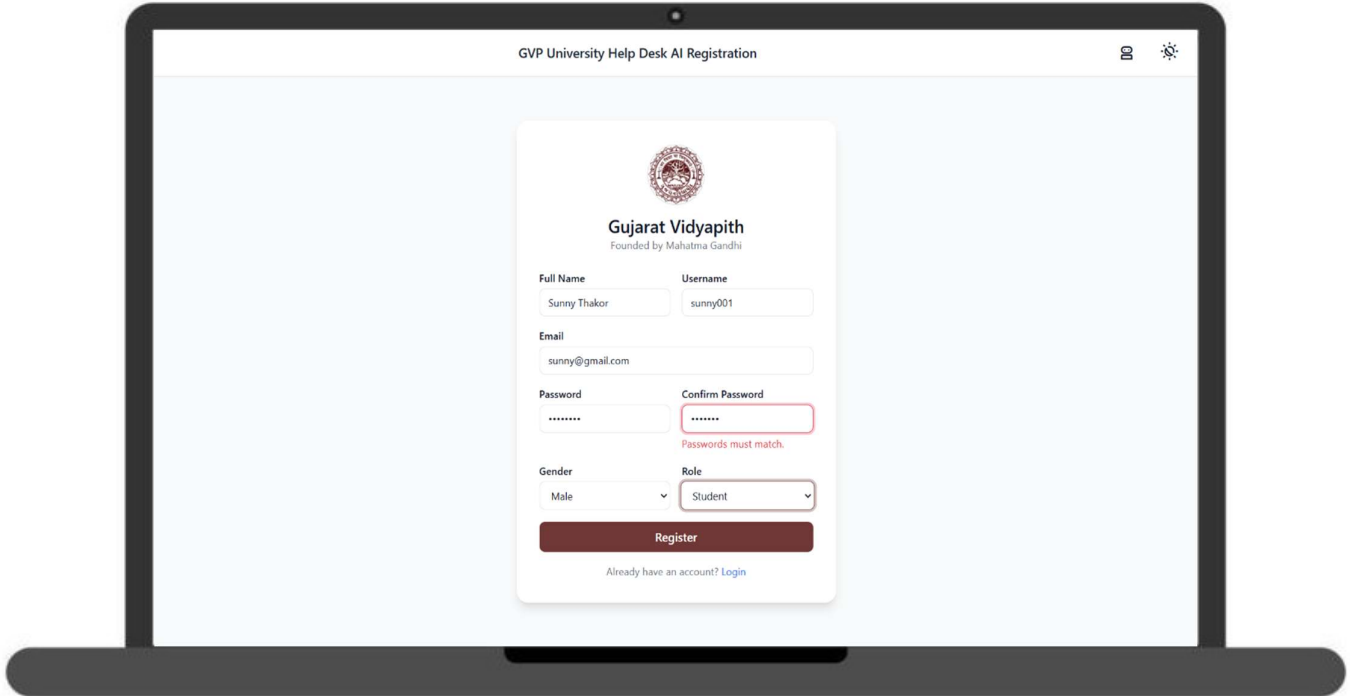


## 9. Add new Intent




## User Side Pages

### 1. User Registration



GVP University Help Desk AI Registration

  
**Gujarat Vidyapith**  
Founded by Mahatma Gandhi

Full Name: Sunny Thakor Username: sunny001

Email: sunny@gmail.com

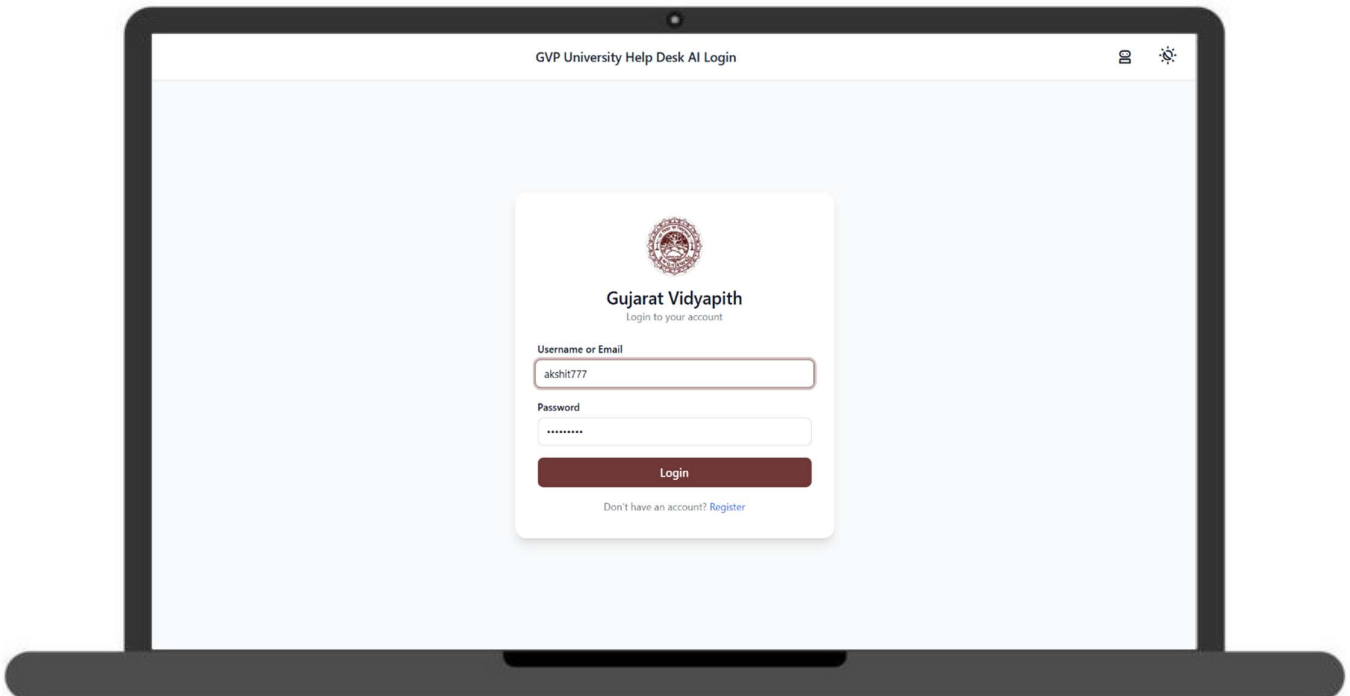
Password: ..... Confirm Password: .....  
Passwords must match.

Gender: Male Role: Student


**Register**

Already have an account? [Login](#)

### 2. User Login



GVP University Help Desk AI Login

  
**Gujarat Vidyapith**  
Login to your account

Username or Email: akshiti777

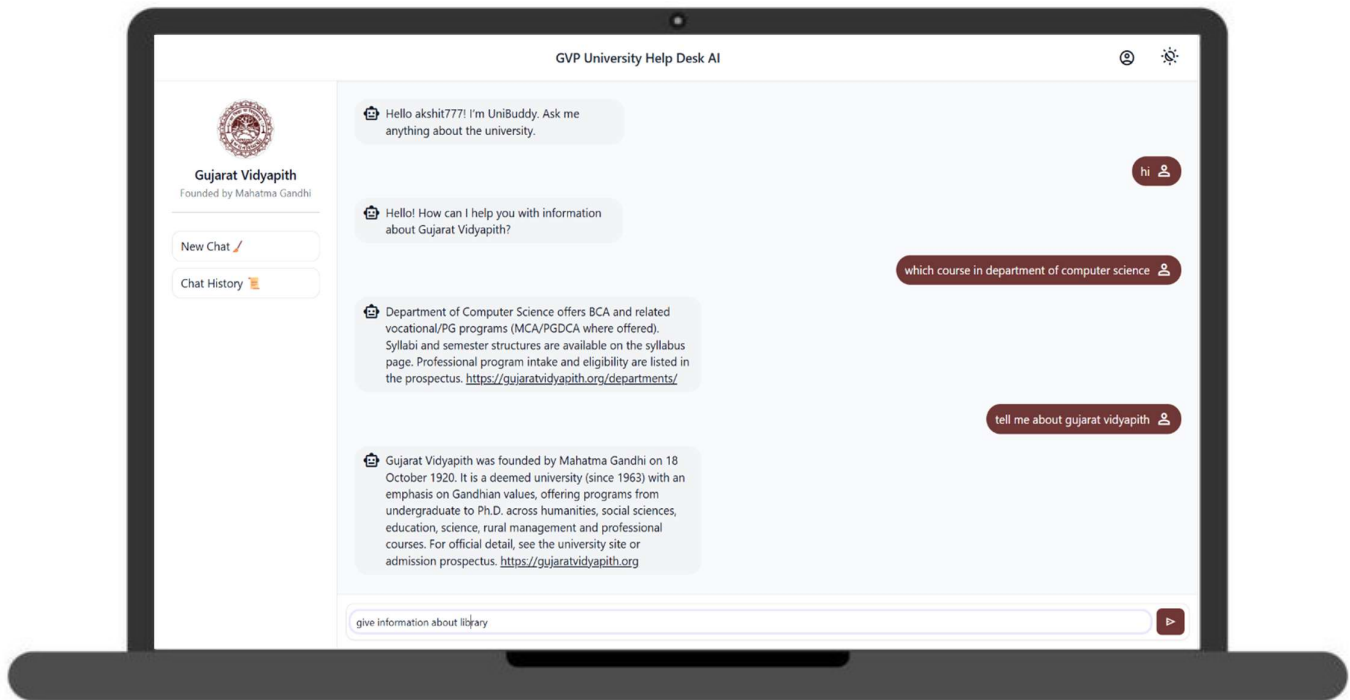
Password: .....

**Login**

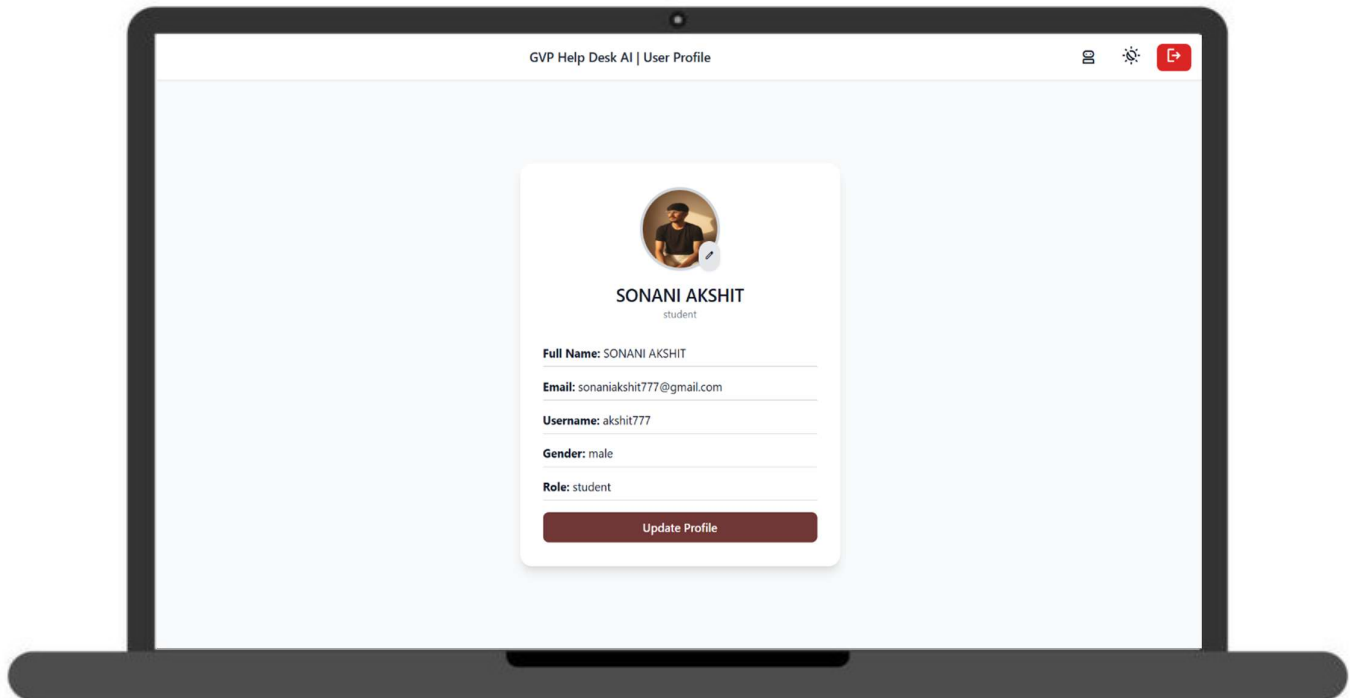
Don't have an account? [Register](#)

# Gujarat Vidyapith Chatbot

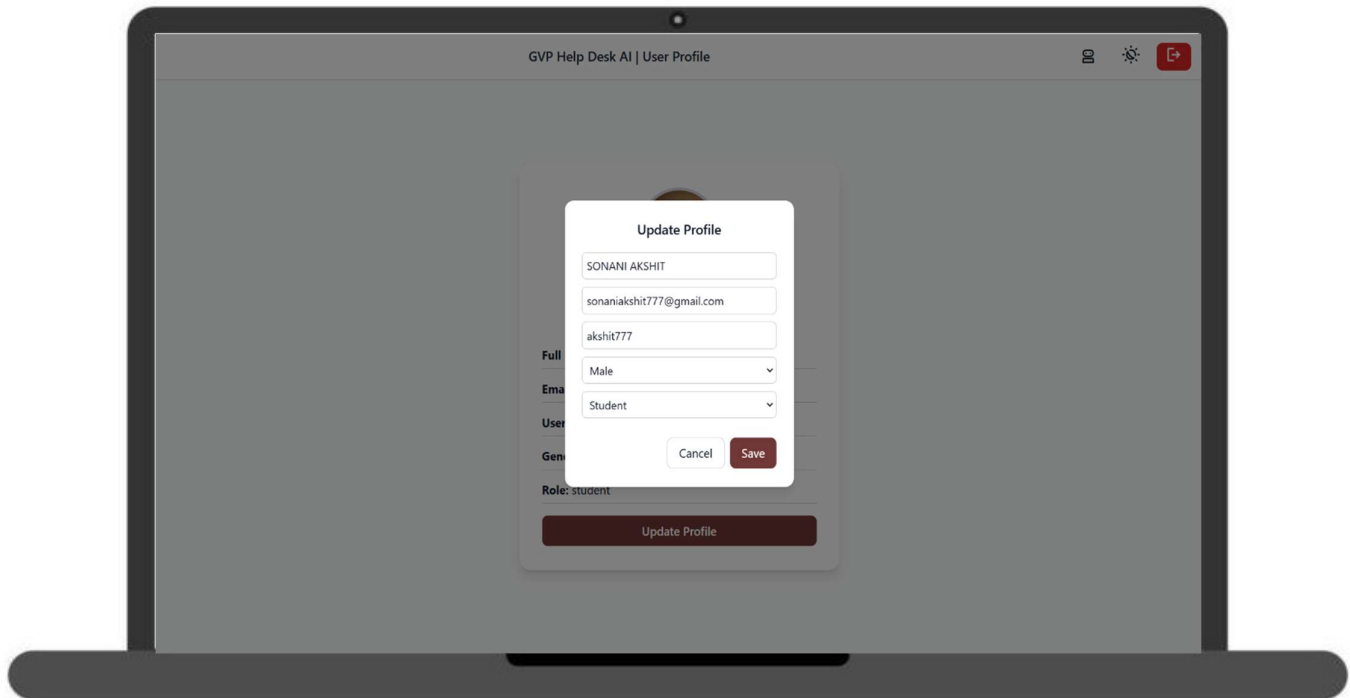
## 3. GVP Chatbot



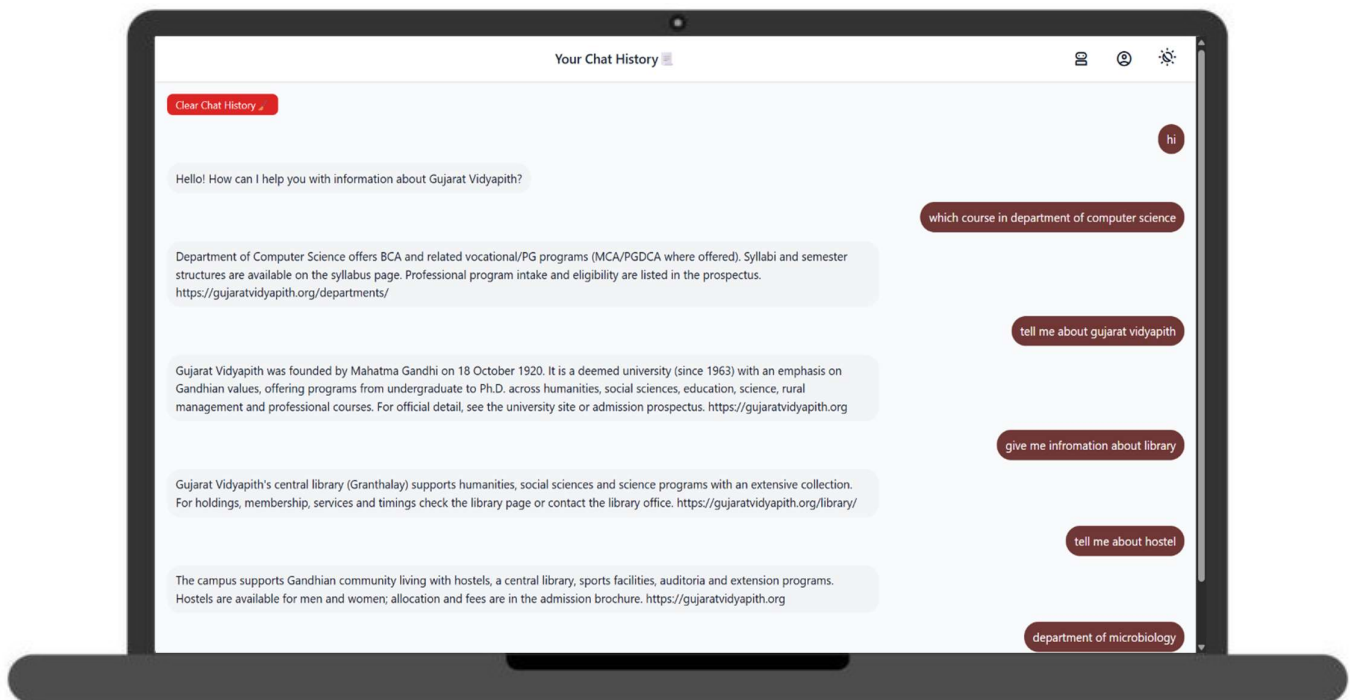
## 4. User Profile



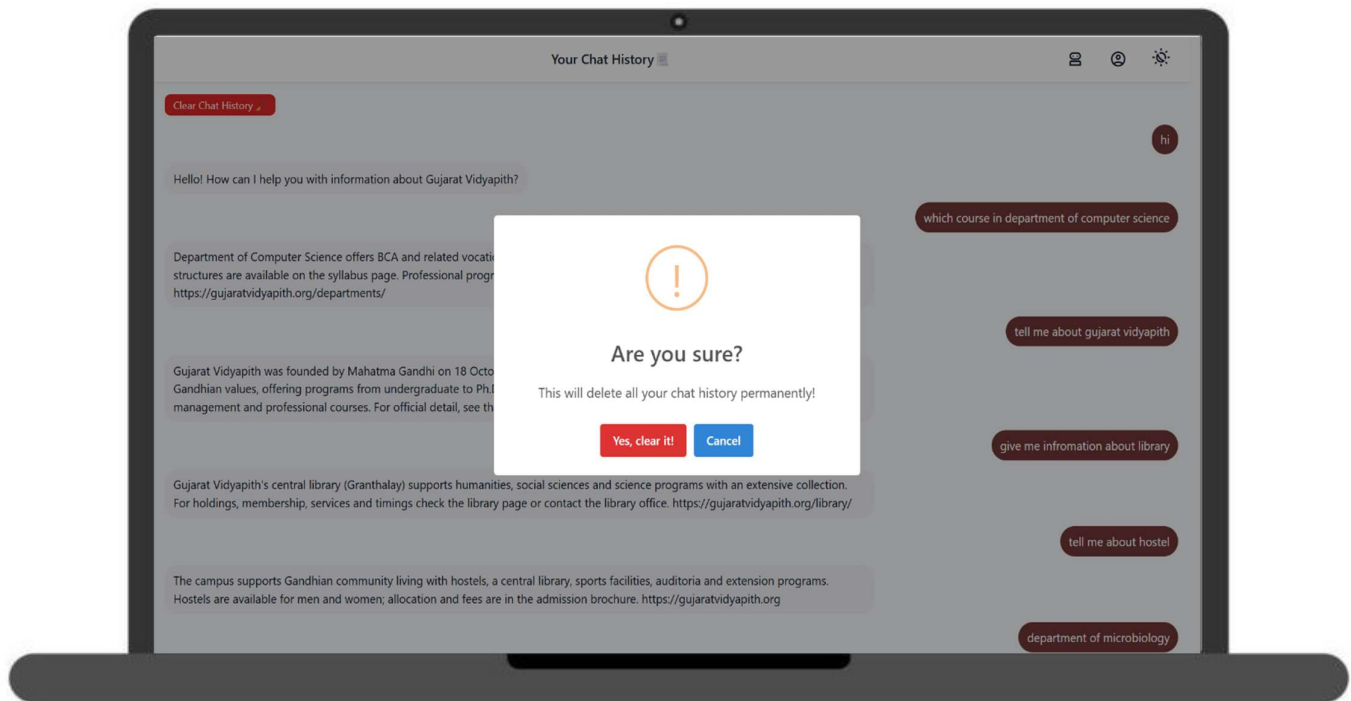
## 5. Update User Profile



## 6. User Chat History



## 7. Clear Chat History





## 8. Limitations of the System

### **Not a Real-Time or Fully Accurate Source**

The chatbot provides high-level guidance based on predefined data. It cannot give real-time updates such as seat availability, changing fee structures, or last-minute admission rules unless the dataset is manually updated.

### **No Official Decision-Making Ability**

The system cannot make admission decisions, verify documents, allocate hostels, or process applications. It only provides general information and links to official sources.

### **Limited Handling of Complex or Ambiguous Queries**

If the user asks a vague, mixed, or highly detailed question, the chatbot may fail to interpret it correctly and revert to fallback messages or clarification prompts.

### **Dependence on Predefined Intents and Data**

The chatbot responds only to the intents and patterns it was trained on. Any new or uncommon question outside the dataset may lead to incorrect or no response.

### **No Automatic Content Updates**

The system does not crawl the university website or prospectus. When university information changes, the dataset and responses must be updated manually.

### **Language and Input Limitations**

The chatbot performs best with clean English queries. Highly informal messages, spelling errors, mixed languages (like Gujarati + English), or slang can reduce accuracy unless additional training data is provided.

### **Does Not Handle Transactions or Secure Portals**

The system cannot manage payments, logins, application submissions, fee payments, or student record access. It only provides informational support.

### **Limited Personalization**

Every user receives the same generic response. The chatbot does not tailor answers based on student history, enrolled course, or personal preferences.

### **Accuracy Depends on Model Quality**

The neural network is trained on a relatively small dataset, so intent classification accuracy may decline when users ask questions in unexpected ways.

### **Requires Manual Maintenance for Improvements**

Updating intents, adding paraphrases, handling new departments, and retraining the model all require manual effort, which may limit scalability.

## 8.1 Proposed Enhancements

Although the current version of the Gujarat Vidyapith Chatbot provides accurate responses using a predefined dataset and a manually trained model, there are several opportunities to enhance the system in future versions. These improvements would make the chatbot more intelligent, scalable and user-friendly.

### 1. Automatic Model Retraining

At present, the chatbot is trained manually using a fixed set of intents and responses. In the future, the system can be upgraded to automatically retrain the model based on real chat logs. This would include:

- Collecting user queries and storing them as training samples
- Automatically identifying new patterns and updating intents
- Running scheduled training jobs (daily/weekly)
- Deploying improved models without manual effort this enhancement will allow the chatbot to evolve continuously and become more accurate over time.

### 2. Multilingual Support

Currently, the chatbot primarily supports English. A major enhancement will be adding support for multiple languages such as Gujarati, Hindi and Hinglish. Future features may include:

- Automatic language detection
- Separate datasets for additional languages
- Translation-based response generation this will make the chatbot more accessible to local users and improve usability for non-English speakers.

### 3. Voice Interaction

In future versions, voice input and output can be integrated using speech-to-text and text-to-speech APIs.

This will allow users to communicate with the chatbot verbally, making the system easier to use for visually impaired students or users who prefer voice-based interaction.

## 9 Bibliography

### Book Referred

Natural Language Processing with Python

### Website Referred

#### **MongoDB Documentation**

MongoDB Inc. *MongoDB Manual and Developer Guide*.

<https://www.mongodb.com/docs>

#### **PyTorch Documentation**

Meta AI Research. *PyTorch Machine Learning Framework Documentation*.

<https://pytorch.org/docs>

#### **NLTK Documentation**

Bird, S., Klein, E., & Loper, E. *Natural Language Processing with Python*.

<https://www.nltk.org>

#### **Djongo Documentation**

*Djongo: MongoDB Connector for Django ORM*.

<https://www.djongomapper.com>

#### **University Website (for informational training data)**

Gujarat Vidyapith. *Official University Website*:

<https://gujaratvidyapith.org>

#### **GitHub Repositories (Open Source Chatbot Examples)**

Various developers. *Chatbot implementations using Python, NLP and Django* (referenced for understanding patterns and structure).

<https://github.com>