4 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.

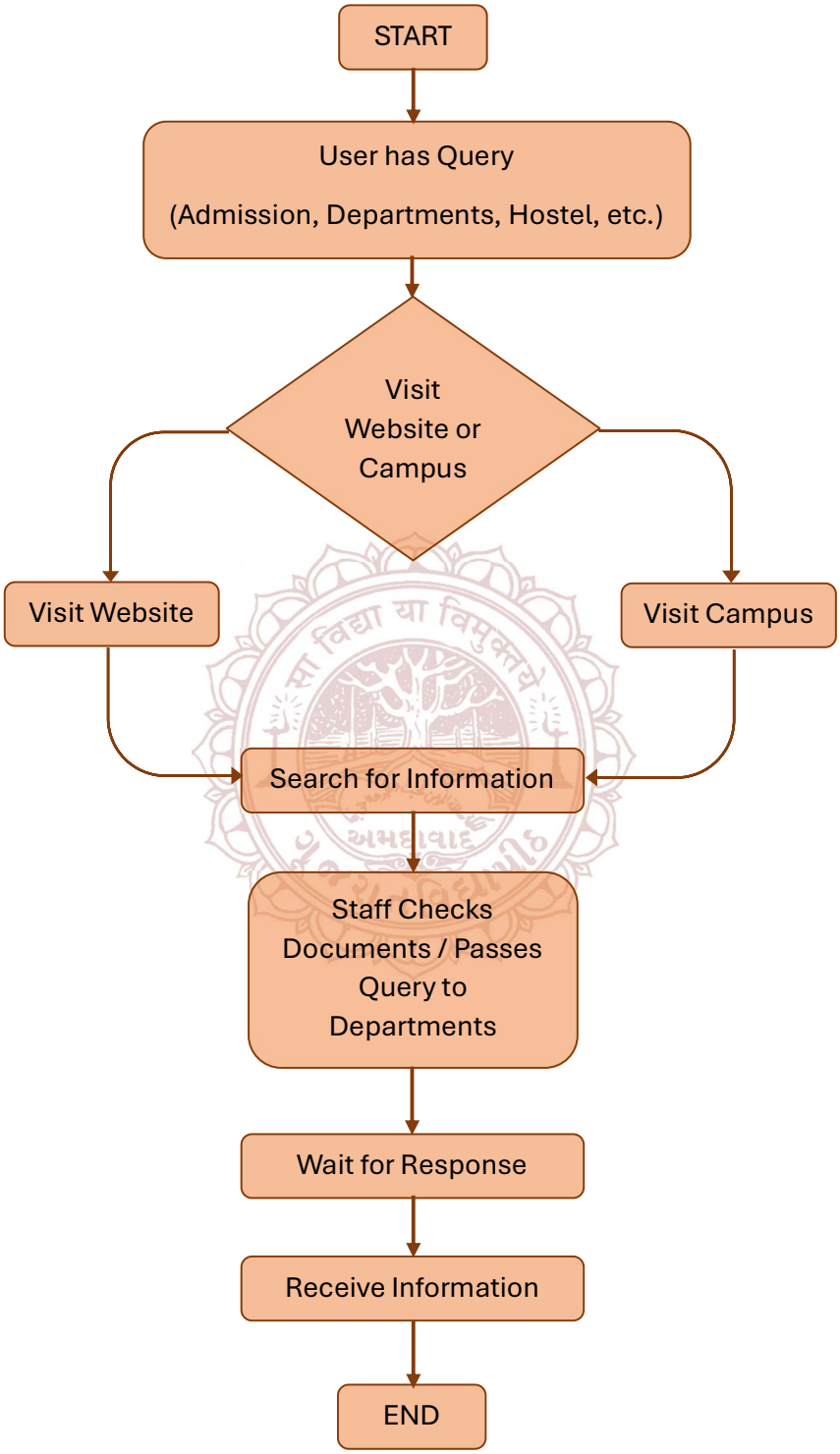
2. University Website

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.

4.1 Existing System Flow Chart



5 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.

6 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.

6.1 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.

6.2 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.

6.3 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

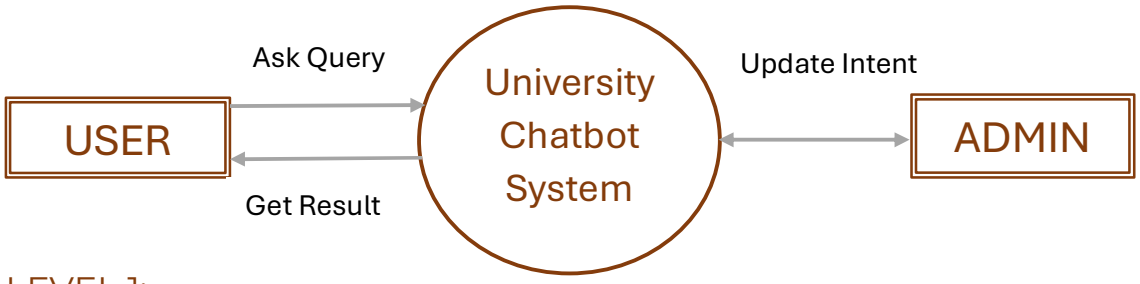
When the chatbot cannot understand a question, it asks for clarification or suggests related topics instead of giving incorrect information. This maintains accuracy and keeps the conversation meaningful.

Scalability and Future Expansion

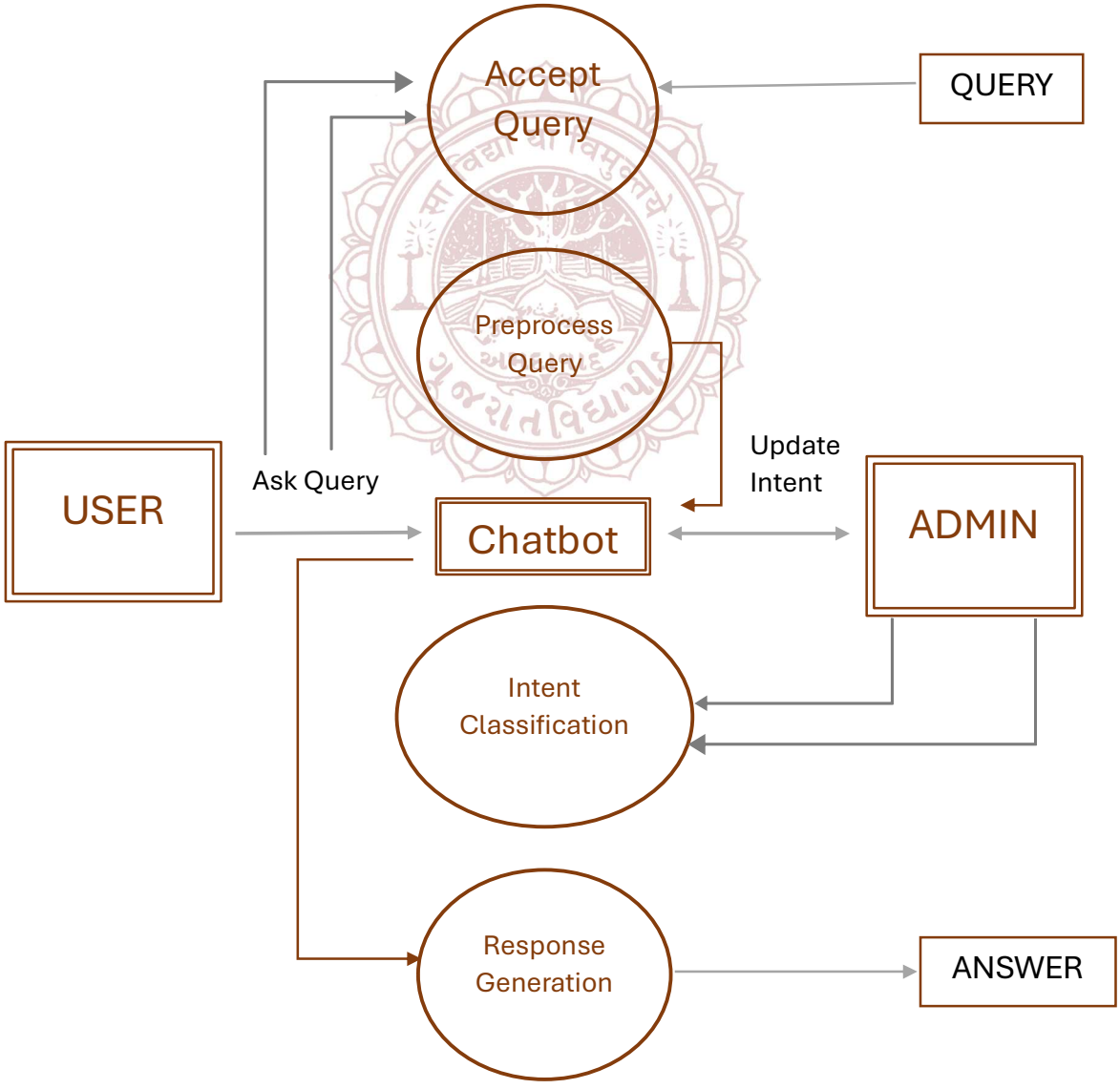
The system is designed to support future upgrades such as multilingual responses, voice interaction, integration with human support, and expanded academic modules. New intents and responses can be added easily as the university's needs grow.

6.3.1 Data Flow Diagram

LEVEL 0:



LEVEL 1:



6.3.2 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:

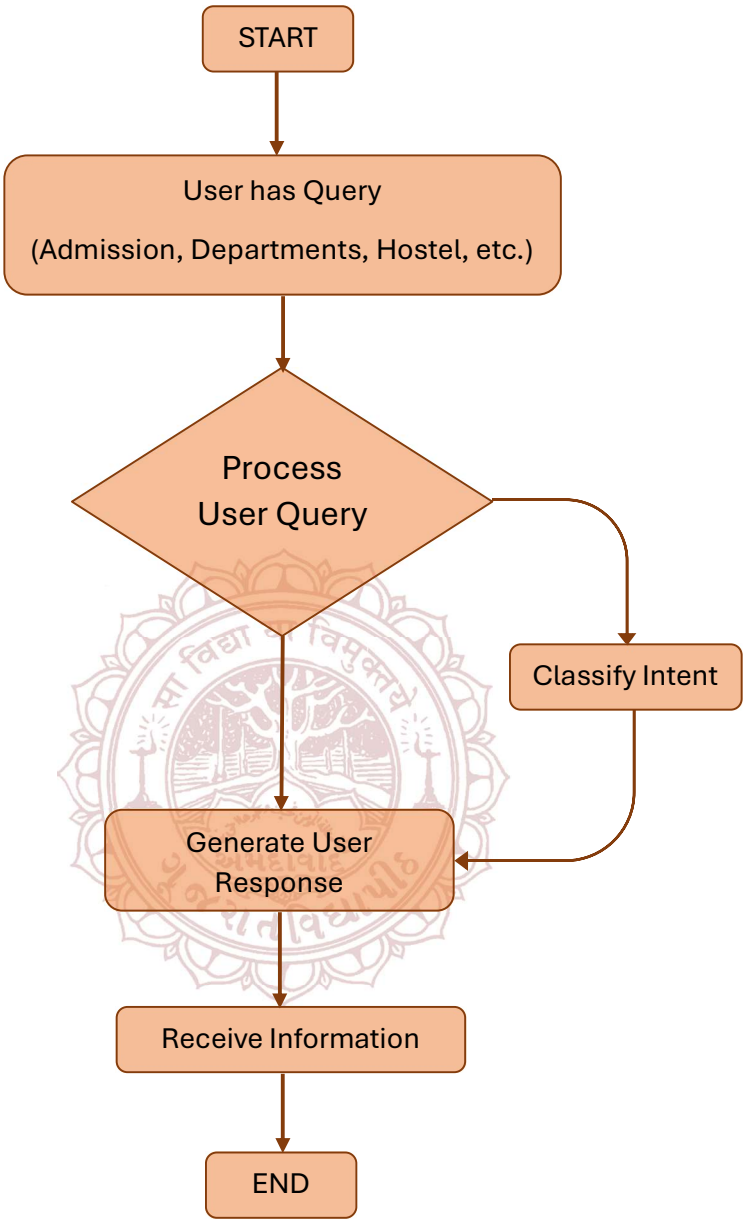
- **PyTorch (torch)** – For building and training the neural network model.
- **NLTK** – For tokenization and basic NLP preprocessing.
- **NumPy** – For vector and matrix operations during training.

5. Frontend Technologies

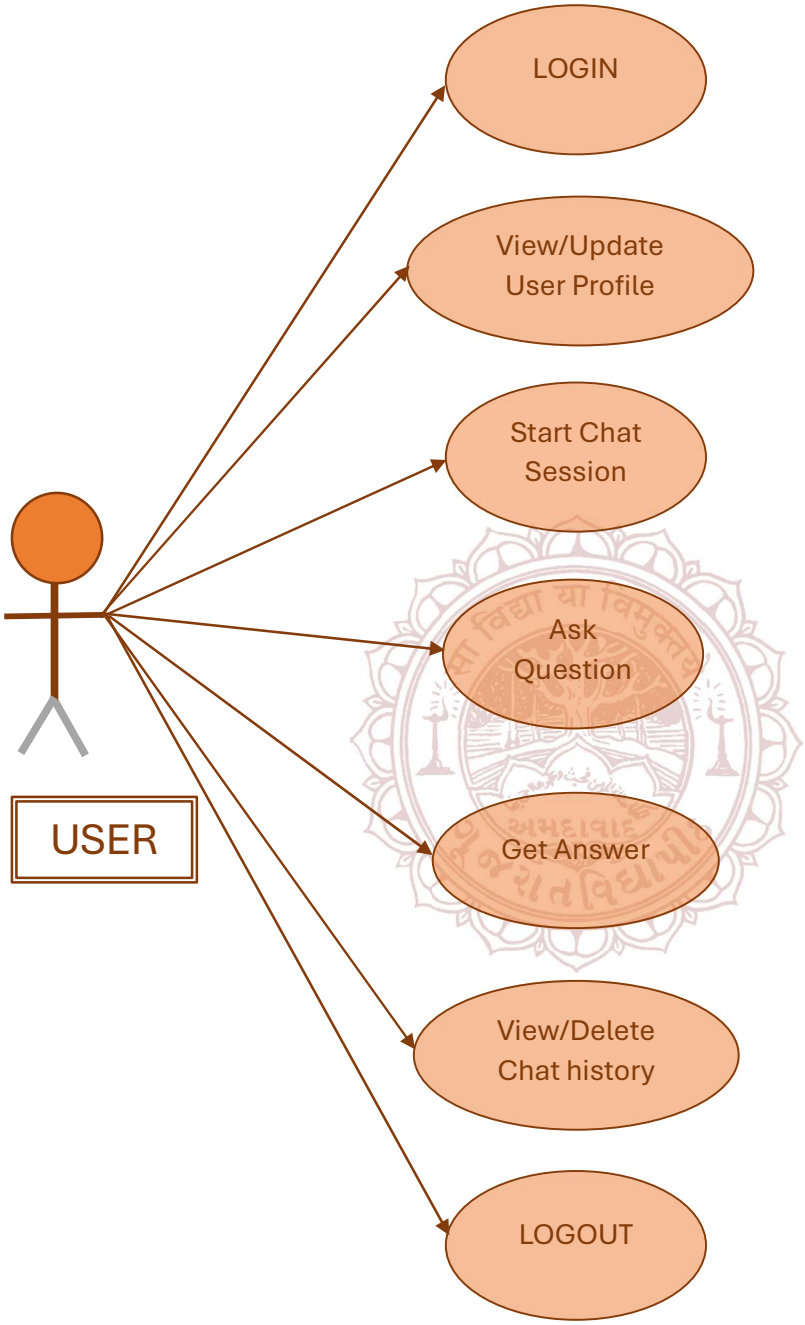
Used to design the chat interface and user experience:

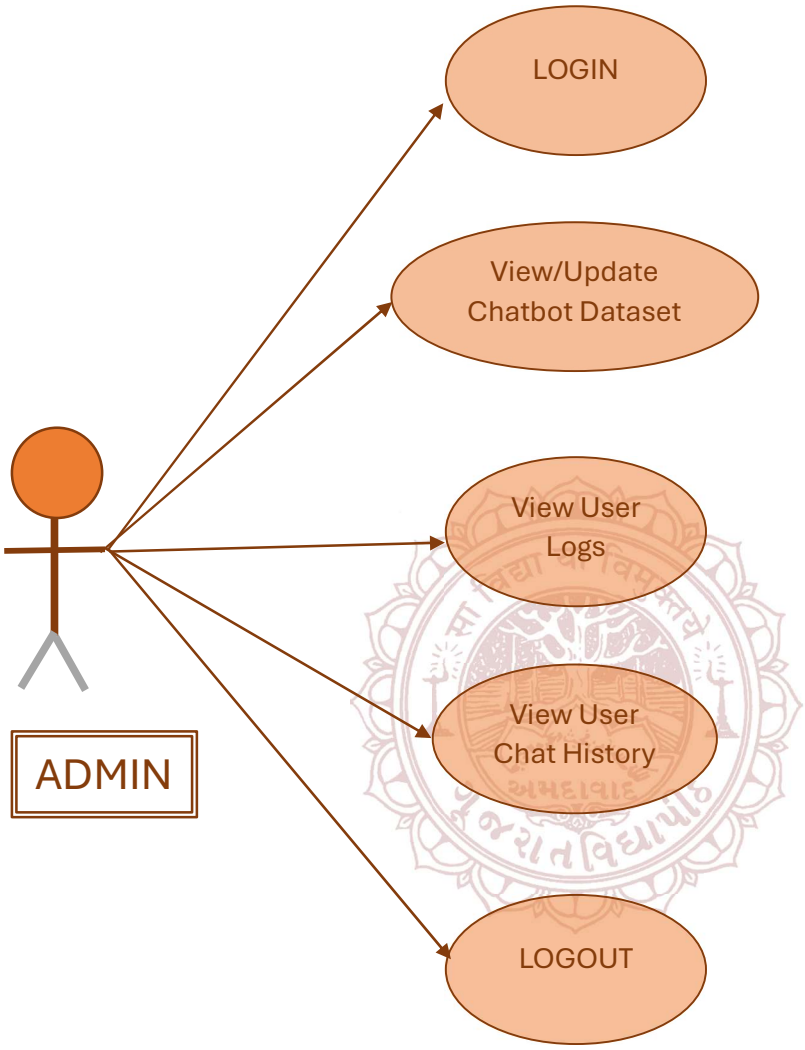
- **HTML5**
- **CSS3**
- **Bootstrap 5** (for layout and styling)
- **JavaScript (AJAX/FETCH)** for sending user queries to Django without reload.

7 Proposed Chatbot System Flow Chart

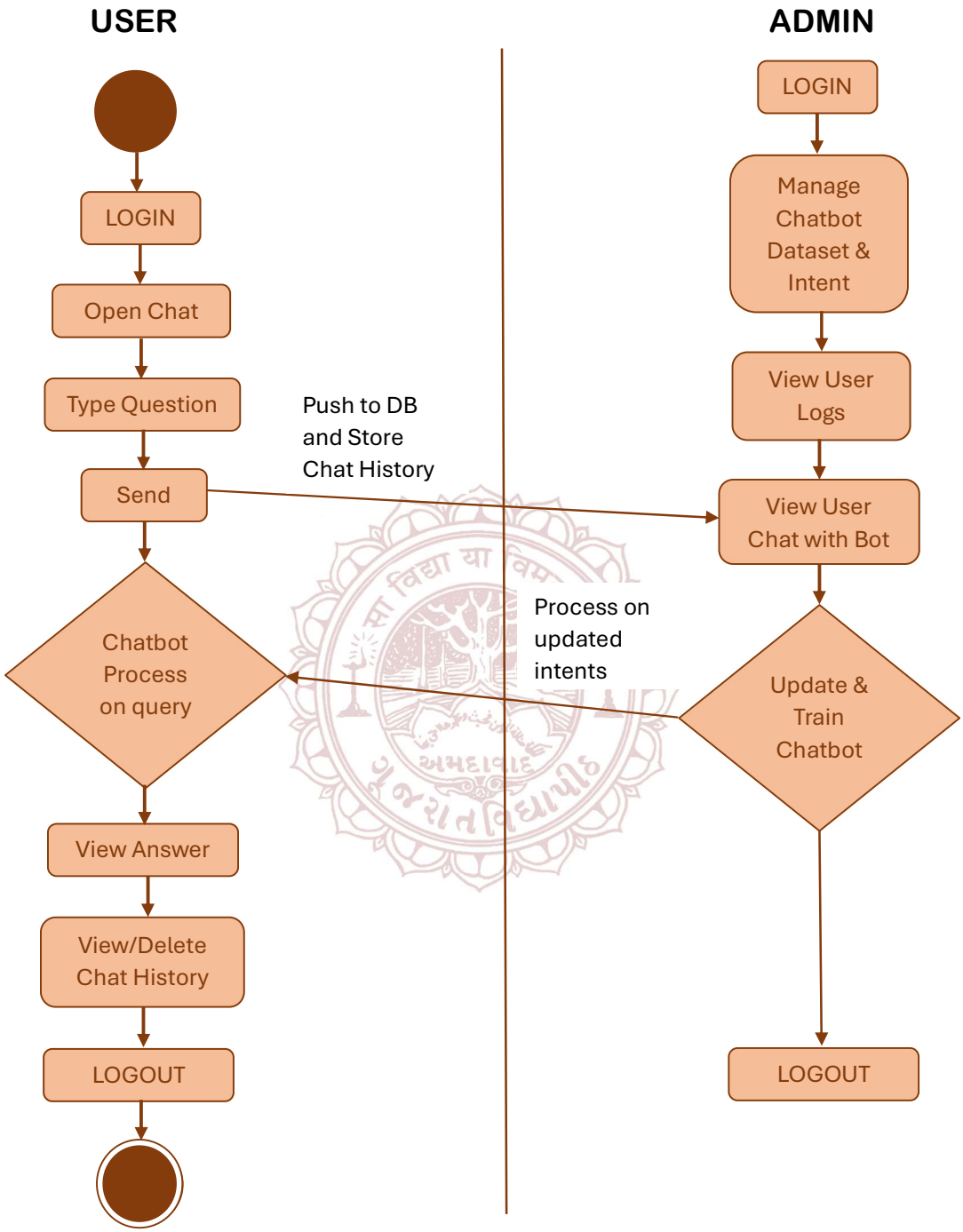


7.1 Use Case Diagram




LOGIN
View/Update
Chatbot Dataset
View User
Logs
View User
Chat History
LOGOUT
ADMIN

7.2 Activity Diagram



7.3 Data Dictionary

Table: ChatBot_chatmessage

Field Name	Data Type	Description	Example