

Web Based Interactive University Chatbot

Dr.V.Shanmugapriya Assistant Professor

Department Of Computer Science shanmugapriyav@skasc.ac.in

Sri Krishna Arts and Science College , Coimbatore.

Rakshit.K.A rakshitka24bcs147@skasc.ac.in

Sri Krishna Arts and Science College , Coimbatore.

ABSTRACT: The project entitled “ Web Based Interactive University Chatbot” is developed using the front-end as PHP and back-end MYSQL tool. This student chatbot project can simulate conversations with the students and staff. The student can ask/query anything with this web application and the chatbot will automatically respond accordingly to the queries/questions. In this project staff module, staff can feed the responses for the possible questions that students may ask/query from the chatbot. The staff panel has many features that are related to the chatbot information. The staff can also manage some details of the chatbot such as the welcome message, no result message for the query, and avatars. Talking about the chatbot responses, the staff will store information in the system and will tag the possible question for that response.

KEYWORD: University chatbot , Web-based chatbot , Interactive chatbot , Virtual assistant for students , AI chatbot for education , Student support chatbot , Academic chatbot .

INTRODUCTION :

This Web-based human-like interactive university chatbot project can simulate conversations with the students and staff. The student can ask anything with this web application and the chatbot will automatically respond accordingly to the questions. chatbot applications are increasingly adopted in various domains such as students and staff and also e commerce and customer services as a direct communication channel between companies and end-users. Multiple frameworks have been developed to ease their definition and deployment. While these frameworks are efficient to design simple chatbot applications, they still

require advanced technical knowledge to define complex interactions and are difficult to evolve along with the company's needs. In this project, a staff module where

staff can feed the responses for the possible questions that students may ask from the chatbot. The staff panel has many features that are related to the chatbot information. The staff can also manage some other details of the chatbot such as the welcome message, and no result message for the query. Talking about the chatbot responses, the staff will store information in the system and will tag the possible question for that response. The system also stores the questions that no answers can be found in the database. In addition, the deployment of a chatbot application usually requires a deep understanding of the targeted platforms, especially back-end connections, costs.

PROBLEM STATEMENT :

Universities often face challenges in providing timely and efficient responses to student queries related to admissions, courses, examinations, schedules, fee payments, placements, and campus facilities. Traditional methods such as manual inquiry desks, emails, and phone calls are time consuming, prone to delays, and require significant human resources. Students today expect instant, 24/7, and personalized assistance, but the existing systems struggle to meet these expectations. As a result, students experience frustration, miscommunication, and delays in accessing critical academic and administrative information. To address these issues, there is a need for a web-based interactive university chatbot that leverages Artificial Intelligence (AI) and Natural Language Processing (NLP) to provide real-time, automated, and user-friendly interaction. Such a system would reduce the burden on university staff, enhance student

experience, and ensure accurate, accessible, and instant support

OBJECTIVES :

1. Provide 24/7 Student Support

Answer common queries instantly (admissions, courses, deadlines, etc.).

2. **Automate Administrative Tasks** Help with application tracking, course registration, and exam info.

3. **Improve Access to Information** Share details on programs, faculty, events, and campus services.

4. Support Admissions & Enrollment

Guide prospective students through application and financial aid processes.

5. **Enhance Campus Engagement** Offer info on clubs, housing, health services, and student life.

6. Enable Multilingual & Inclusive Access

Support diverse users, including international and differently-abled students.

7. **Collect Insights & Feedback** Track user questions to improve services and inform decision-making.

8. **Integrate with University Systems** Connect with LMS, SIS, and other platforms for personalized responses.

9. **Ensure Data Privacy & Security** Follow FERPA/GDPR and secure sensitive student data.

METHODOLOGY :

1. Requirement Analysis

- Define objectives (e.g., assist with student/staff queries).
- Identify target users (students, faculty, applicants).
- Set scope (FAQs, admissions, academics).
- Gather data from university sources.

2. Design Phase

- **Architecture:**
 - Frontend: HTML, CSS, JS, React.
 - Backend: Python (Flask/Django), Node.js.
 - Database: MySQL, MongoDB, Firebase.
 - NLP: Dialogflow, Rasa, OpenAI, Watson.
- Design intents, entities, and conversation flows.

3. Development Phase

- Build responsive frontend with chatbot UI.
- Connect backend with NLP and database.
- Train NLP model with university-specific queries.
- Add fallback/error handling.

4. Integration Phase

- Link to university databases and portals.
- Optional: Add email, calendar, or SMS integrations.

5. Testing Phase

- Functional, usability, performance, and security testing.

6. Deployment

- Deploy on university website/cloud (AWS, Heroku).
- Monitor usage and performance.

7. Maintenance

- Update intents and data regularly.
- Improve based on feedback.
- Add new features (e.g., voice, multilingual).

8. Tools & Technologies

- Languages: Python, JavaScript
- Frameworks: Flask, React
- NLP: Dialogflow, Rasa, OpenAI
- Hosting: Firebase, AWS

9. Evaluation Metrics

- Intent accuracy
- Response time
- User satisfaction
- Usage stats

Key Components in the Design Phase

1. System Architecture Design

- **Frontend:** User interface on the website or portal.
- **Backend:** Server-side logic and APIs.

- **Database:** Stores user queries, FAQs, and chat logs.
- **NLP Engine:** Processes natural language input from users.

2. Conversation Flow Design

- Define **intents** (user goals, e.g., "check admission status").
- Define **entities** (keywords like "course name", "semester").
- Design **dialog paths** (how the chatbot responds step by step).

3. Technology Stack Selection

- **Frontend:** HTML, CSS, JavaScript, React/Vue.
- **Backend:** Python (Flask/Django), Node.js.
- **NLP:** Dialogflow, Rasa, or OpenAI API.
- **Database:** Firebase, MongoDB, MySQL.

ALGORITHM USED :

1. Rule-Based

- Uses: *If-else logic*, pattern matching, decision trees
- Best for: Simple FAQs (e.g., deadlines, fees)

2. Retrieval-Based

- Uses: *Intent classification* (SVM, Naive Bayes), *TF-IDF*, *Cosine similarity*
- Best for: Matching questions to predefined answers (e.g., course info)

3. Generative (AI-Based)

- Uses: *Deep learning*, *Transformers* (e.g., GPT, BERT)
- Best for: Natural conversations, open-ended queries

4. Hybrid Approach

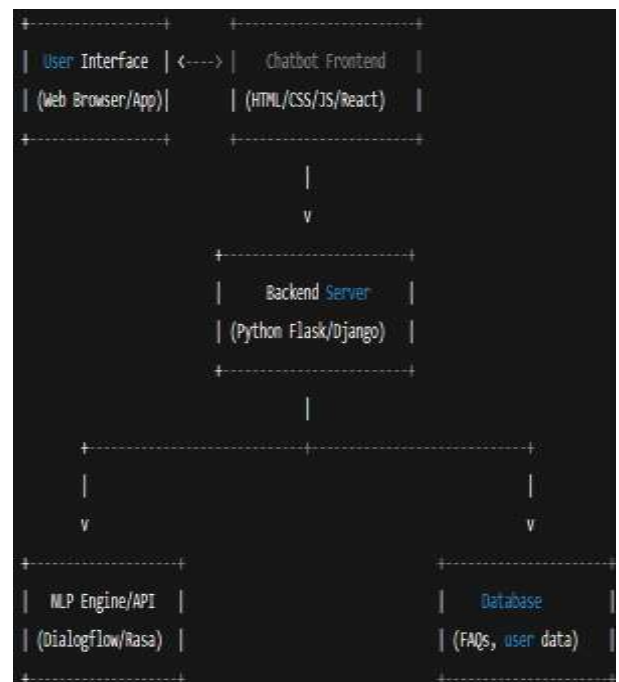
- Combines rule-based + AI
- Best for: Complex chatbots needing both control and flexibility

A **Generative Algorithm** is an algorithm that can **generate new content** (such as text, images, music, etc.) based on patterns it has learned from data. Unlike rule-based systems, generative models can **create**

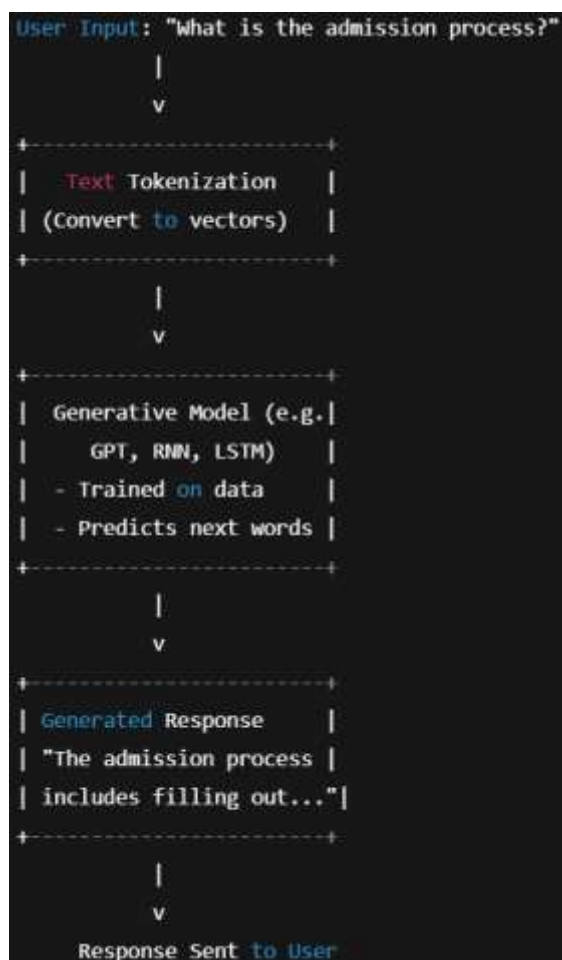
human- like responses or outputs, making them ideal for **chatbots, language models, and creative AI**.

Key Characteristics of Generative Algorithms

- **Learns from large datasets** (text, conversations, etc.)
- **Generates new responses** word- by-word or token-by-token.
- **Context-aware:** Maintains conversation flow.



- Used in models like GPT, LLaMA, etc.



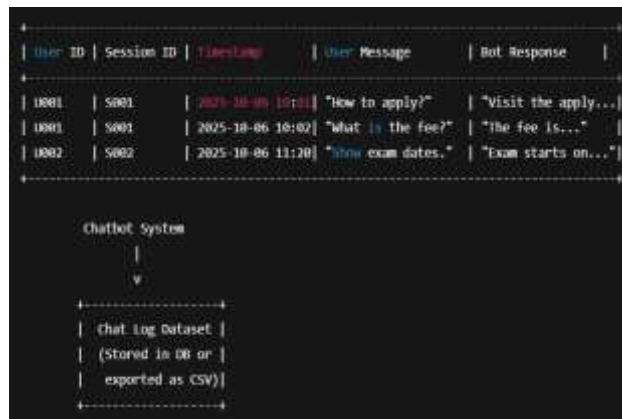
DATA SET :

1. **FAQ Dataset** – Common student questions and answers (e.g., admissions, fees).
2. **Intent Dataset** – User queries labeled with intents (e.g., "ask deadline", "apply").
3. **Entity Dataset** – Labeled data for extracting names, dates, courses, etc.
4. **Chat Logs** – Real or simulated conversations for training dialogue flow.
5. **External NLP Datasets** – Optional (e.g., SNIPS, DailyDialog for general NLP tasks).
6. A **Chat Log Dataset** is a collection of recorded conversations between users and a chatbot. It includes **user inputs (queries)** and the **bot's responses**, often with metadata like timestamps, user IDs, and intent classification.

This dataset is essential for:

- **Training NLP models**
- **Improving chatbot accuracy**

- **Analyzing user behavior**
- **Debugging and feedback collection**



CONCLUSION :

The development and deployment of a web-based interactive university chatbot mark a significant advancement in enhancing digital communication and service delivery within academic institutions. This chatbot serves as a 24/7 virtual assistant, capable of addressing student inquiries, providing course details, answering administrative questions, and guiding users through various academic and non-academic processes. By leveraging natural language processing and real-time interaction, the chatbot improves user satisfaction through immediate, accurate, and personalized responses. It reduces the burden on university support staff, enhances operational efficiency, and ensures that students receive consistent and timely information regardless of time or location. Moreover, as the system evolves through continuous learning and feedback, it can become even more intelligent, adaptive, and integrated with other campus systems such as learning management platforms, enrollment systems, and student portals. In conclusion, the web-based university chatbot is a vital step toward creating a smarter, more connected, and student-centered academic environment.

REFERENCES :

- Jesus Castagnetto, Sascha Schumann, "Professional Php Programming", Addison wisely Publication, Fifth Edition.
- Jay Greenspan, Brad Bulgar, "Mysql/Php Database Applications", Tata McGraw- Hill

Publishing Company, Third Edition.

- William Stallings,
“Cryptography And Network Security”, Tata McGraw-Hill Publishing Company, Third Edition.
- Bruce Schneier, “Applied Cryptography”,
Pearson Education, Second Edition.
- Rogers Pressman, “Software Engineering and Applications”, Galgotie Publication, Sixth Edition.