International Journal of Electrical Engineering and Ethics- Volume 8 Issue 1, January - February - 2025

EDU ASSIST : AN INTELLIGENT ASSISTANT FOR COLLEGE ENQUIRY

ANMOL KUMAR
(Dept. of Computer Science Engineering, Siddhartha Institute of Technology and Sciences)

R. SIRISHA
Assistant Professor
shirisharangu.cse@siddhartha.co.in
(Dept. of CSE, Siddhartha Institute of Technology and Sciences)

G.AJAY KUMAR
(Dept. of Computer Science Engineering, Siddhartha Institute of Technology and Sciences

CHIKKA VARSHA
(Dept. of Computer Science Engineering, Siddhartha Institute of Technology and Sciences)

DAYAVARI TEJA
(Dept. of Computer Science Engineering, Siddhartha Institute of Technology and Sciences)

***Abstract*—The aim of this project is to develop a college enquiry Chabot that answers any queries post by students like college details, course-related questions, location of the college, fee structure etc. This project is built using Python that analyze user's queries and understand the user's message. This system is a web-based application designed to provide responses to user queries. Individuals can interact with the bot by submitting their questions, and the responses are tailored to match the queries. Users can inquire about various college-related activities through the platform. The system eliminates the need for the user to visit the college in person for inquiries. It processes the user's question and provides an appropriate response. Additionally, users can share their feedback through a suggestion box. The system utilizes an interactive Graphical User Interface, creating the impression of a real person engaging with the user.**

Keywords—Chatbot, Python, Chatterbot, Django

INTRODUCTION

In today's digital era, educational institutions work towards improving communication and optimizing processes to meet the dynamic needs of students and staff. A college enquiry chatbot functions as a virtual assistant, providing current and prospective students with easy access to information and helping them resolve their queries in real time. A chatbot is software designed to simulate human-like conversations through text with the primary goal of answering user questions. These bots are engaging tools that can be found across various platforms, from old websites to modern social media. Chatbots are often seen as interesting and dynamic systems with which humans can communicate. The College Enquiry Chatbot is developed to handle academic-related inquiries such as admissions, fee structures, scholarship details, department timetables, and document requirements. The chatbot offers a text-based interface, allowing students to type questions and receive immediate responses. By analyzing the users input the chatbot would draws form its stored data to provide the accurate answers making it an efficient tool for addressing queries in less time this system helps students navigate *academic processes more* easily reducing the need for direct human assistance while offering prompt and reliable information.

LITERATURE SURVEY

Eliza is considered as the first Chatbot, which works on the pattern matching system. It is developed by Joseph Weizenbaum in 1964. ELIZA was one of the first chatterbots and one of the first programs capable of attempting the Turing test. ELIZA's creator, Weizenbaum, regarded the program as a method to show the superficiality of communication between man and machine, but was surprised by the number of individuals who attributed human-like feelings to the computer program, including Weizenbaum's secretary.[1]

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity), also referred to as Alicebot, or simply Alice, is a natural language processing chatterbot —a program that engages in a conversation with a human by applying some heuristical pattern matching rules to the human's input. It was inspired by Weizenbaum's classical ELIZA program. The program is unable to pass the Turing test, as even the casual user will often expose its mechanistic aspects in short conversations. ALICE was implemented by Richard Wallace in 1995.[2]

Cleverbot is a chatterbot web application that uses an artificial intelligence (AI) algorithm to have conversations with humans. It was created by British AI scientist Rollo Carpenter. Unlike some other chatterbot. Cleverbot's responses are not pre programmed. Instead, it learns from human input: Human's type into the box below the Cleverbot logo and the system finds all keywords or an exact phrase matching the input. After searching through its saved conversations, it responds to the input by finding how a human responded to that input when it was asked, in part or in full, by Cleverbot.[3]

Kuki, formerly known as Mitsuku, is a chatbot created from Pandorabots AIML technology by Steve Worswick. It is a five time winner of a Turing Test competition. Kuki claims to be an 18-year-old female chatbot from the Metaverse. It contains all of Alice AIML files, with many additions from

user generated conversations, and is always a work in progress.[4]

PARRY was written in 1972 by psychiatrist Kenneth Colby, then at Stanford University. Parry is natural language program that simulates the thinking of a paranoid individual. This program was the first to pass the "Turing Test".[5]

PROBLEM FORMULATION

To design and develop "College Enquiry Chatbot" using knowledgeable dataset

PROBLEM DEFINITION

To Create and implement an chatbot designed specifically for a college to help prospective students with questions about admissions, course details, campus amenities, and other academic or administrative information. Colleges frequently receive numerous inquiries from students and parents about admissions, courses, tuition fees, scholarships, campus amenities, and other relevant details. Addressing these inquiries manually takes a significant amount of time and can result in delays, inconsistent answers, and inefficient communication.

OBJECTIVE OF PROJECT

The main aim of the project is to create an facility whereby students can have access of enough information concerning their prospective college at any given time.To Establish a mechanism for users to provide feedback on the chatbot's performance, allowing for continuous refinement and optimization based on user input and to keep the users fully updated about the ongoing and upcoming events on the campus. The College Enquiry Chatbot would develop an Natural Language Processing (NLP) capabilities to understand diverse queries and adapt to different conversational styles, ensuring effective communication.

SCOPE OF PROJECT

The scope of this project encompasses the development and the deployment of an chatbot designed to handle a wide range of college-related queries. The chatbot will serve as a virtual assistant, offering an efficient and user-friendly interface for the students, parents, and other stakeholders to access important information. This project aims to enhance the efficiency of communication, improve student engagement, and streamline the information dissemination process for colleges, making it a vital tool for modern educational institutions and to provide an Real-time Query Resolution , 24/7 Accessibility and User-friendly Interface

RELATED WORK

College Websites and Brochures:

Colleges maintained static websites and printed brochures that provided general information. Students had to manually search for information, often navigating through multiple pages of the website or reading through lengthy brochures.

On-Campus Help Desks:

Colleges typically had help desks or reception areas where staff would assist visitors with queries about admissions, programs, fees, and campus facilities.

Email Communication:

Prospective students often reached out via email to get their questions answered. Colleges had designated admissions officers or customer service teams to respond to these emails, which could take time depending on the volume of queries.

Phone Calls:

Many inquiries were handled over the phone, where students or parents would speak with administrative staff to get information. Long wait times and limited availability of staff were common challenges with this system.

Open Days/Information Sessions**:**

Colleges would host physical open days or informational events where students could ask questions in person. While helpful, these events were scheduled and not available year-round.

DRAWBACKS OF EXISTING SYSTEM

1. Time-Consuming and Inefficient: Students had to wait for responses through calls, emails, or in-person visits, leading to long waiting times for information. The manual query resolution process also put a strain on administrative staff, making it inefficient.

2. Limited Accessibility**:** Traditional systems, such as help desks or phone lines, were only available during office hours, restricting students' access to information outside of these times. Students in different time zones or with busy schedules faced difficulties in getting timely responses.

3. Overburdening of Administrative Staff: Administrative staff faced high volumes of repetitive queries, leading to overwork and delays in addressing other important tasks. This increased the workload on staff and impacted their ability to provide quality service.

4. Slow Updates to Static Information: Updates to college websites or brochures were not always immediate. If there were changes in admission deadlines or course offerings, students may not

have been aware of them right away, causing delays in getting the most current information.

PROPOSED SYSTEM

Skills and interests assessment: Users should be prompted to take an assessment that will help them identify their skills and interests. This will help the website provide tailored career recommendations.
Education and training resources: The website should provide resources for users to learn new skills and pursue education and training opportunities. This could include online courses, certifications, and degree programs

SYSTEM DESIGN

The system design includes an architectural overview and the use of Unified Modeling Language (UML) diagrams, such as class diagrams, use case diagrams, sequence diagrams, and activity diagrams. This section outlines the components, modules, interfaces, and data flow within the chatbot system.

The architectural design comprises three main layers: the user interface, the application logic, and the database. The user interface facilitates interaction between the user and the system through a web-based platform. The application logic, developed in Python, handles query processing and response generation using NLP techniques. The database stores all relevant information, including frequently asked questions, course details, and user feedback.

UML diagrams are employed to visualize the system architecture and its components. The use case diagram illustrates the interaction between users and the chatbot. The class diagram defines the structure of the chatbot system, detailing the attributes and methods of each class. Sequence diagrams depict the flow of messages between objects during the execution of a particular function, while activity diagrams represent the flow of control or data within the system.

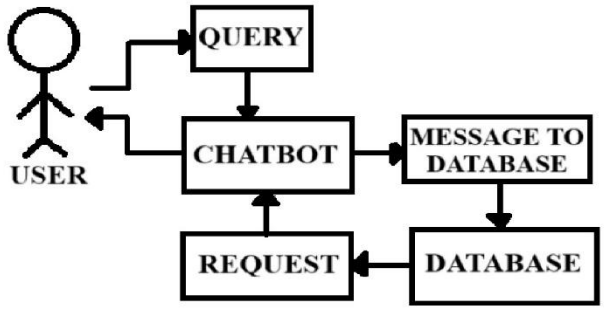

FIG. 1. IT SHOWCASES THE ARCHITECTURE DIAGRAM OF THE PROJECT

METHODOLOGY

The chatbot system employs Natural Language Processing (NLP) techniques to understand and respond to user queries. The backend is developed using Python and Flask, with a MySQL database for storing relevant information. The frontend is designed using HTML, CSS, and JavaScript for an intuitive user experience.

The methodology involves several steps, starting with data collection and preprocessing. A dataset comprising various queries and responses is curated to train the NLP model. Tokenization, stemming, and lemmatization are applied to preprocess the text data, converting it into a format suitable for model training. The chatbot uses a bag-of-words approach to represent text data numerically, followed by the implementation of a machine learning model to classify and generate responses.

The Flask framework is used to develop the web server, handling HTTP requests and routing them to the appropriate functions. The MySQL database stores the preprocessed data and the trained model, enabling the chatbot to retrieve and respond to queries efficiently. The frontend design ensures a seamless user experience, allowing users to interact with the chatbot through a simple and intuitive interface.

UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering.

The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software.

In its current form UML is comprised of two major components: a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process.

The UML uses mostly graphical notations to express the design of software projects.

GOALS The Primary goals in the design of the UML are as follows:

1) Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2) Provide extendibility and specialization mechanisms to extend the core concepts.

3) Be independent of particular programming languages and development process.
4) Provide a formal basis for understanding the modelling language.
5) Encourage the growth of OO tool.

TYPES OF UML DIAGRAM

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modelling tools include:

A. USECASE DIAGRAM

Actor: You (the person whose career roadmap it is).
Use Cases: Major career milestones or achievements you want to reach

Example: "Become proficient in programming languages, Attain a leadership position, Earn a certification," etc.

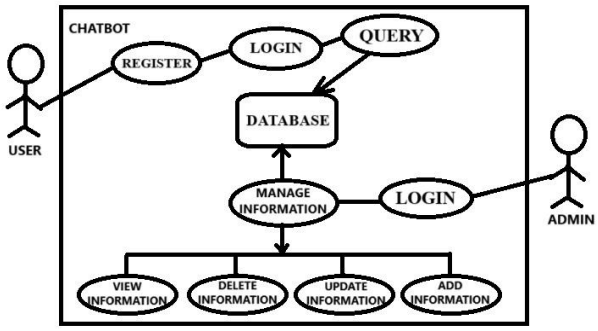

FIG. 2. USE CASE DIAGRAM

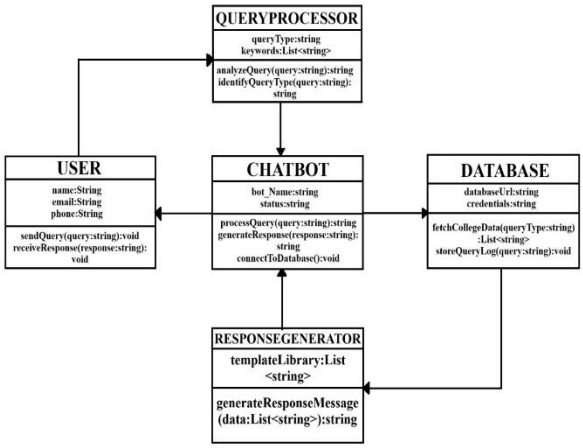

FIG. 3. CLASS DIAGRAM

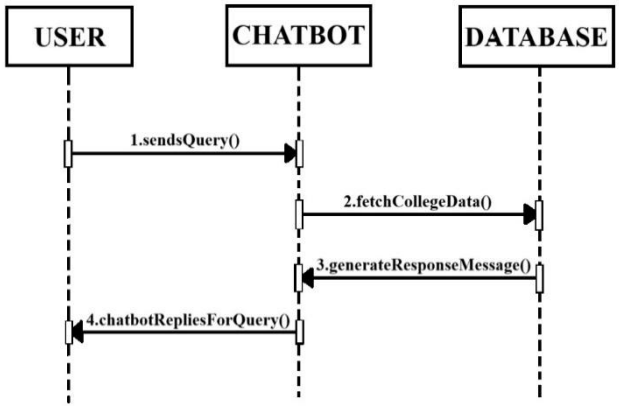

FIG. 4. SEQUENCE DIAGRAM

IMPLEMENTATION

The implementation involves coding the chatbot's functionalities, integrating the Flask framework with the database, and developing the user interface. The system is tested iteratively to ensure accurate responses and robust performance under various query scenarios.

The chatbot is implemented in Python, leveraging libraries such as NLTK for NLP tasks and SQLAlchemy for database interactions. The backend code is structured into modules, each responsible for specific tasks such as query handling, response generation, and user session management. The frontend is developed using HTML and CSS, with JavaScript enhancing interactivity through AJAX calls, ensuring real-time communication between the client and server.

During implementation, emphasis is placed on optimizing the system for performance and scalability. The code is regularly refactored to improve efficiency, and various testing techniques are employed to identify and fix bugs. Unit tests ensure individual components function correctly, while integration tests verify the overall system functionality.

RESULT

The chatbot's performance is evaluated based on its response accuracy, user satisfaction, and system reliability. Metrics include the speed of response, error rates, and feedback from students interacting with the chatbot.

User testing is conducted to gather feedback on the chatbot's usability and effectiveness. The system's accuracy is measured by comparing the chatbot's responses with expected answers for a set of predefined queries. The evaluation also includes stress testing to assess the system's performance under heavy loads, ensuring it can handle multiple simultaneous queries without degradation in response time or accuracy.

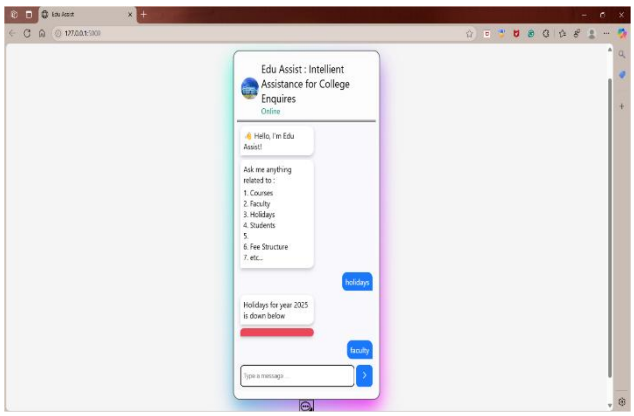

FIG. 5. DEFAULT HOMEPAGE

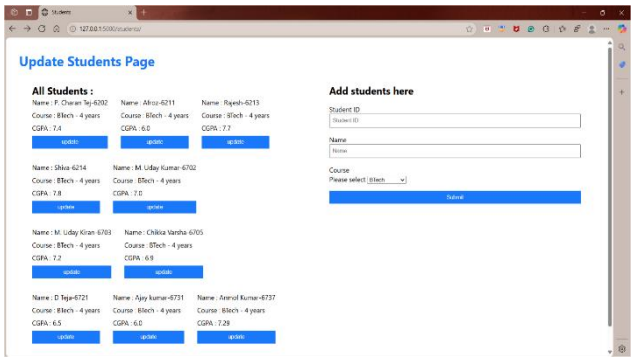

FIG. 6. STUDENTS ENQUIRY PORTAL

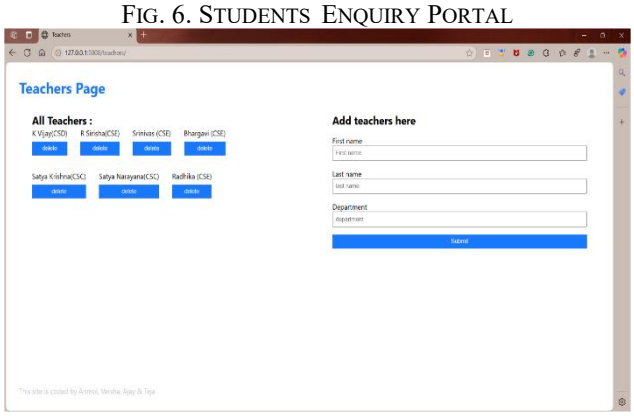

FIG. 7. TEACHER'S PORTAL

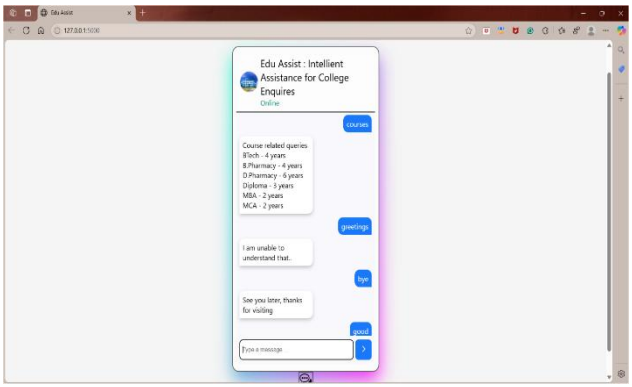

FIG. 8. COLLEGE ENQUIRY

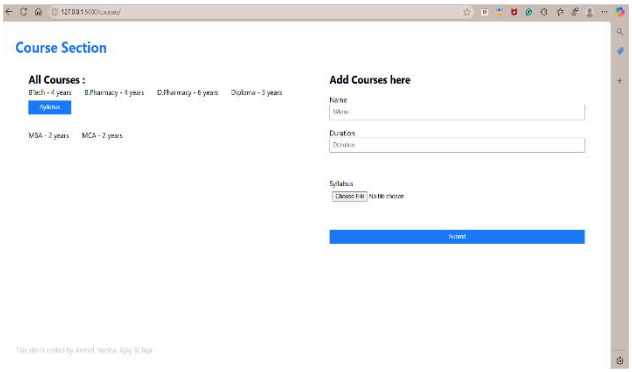

FIG. 9. COURSES PORTAL

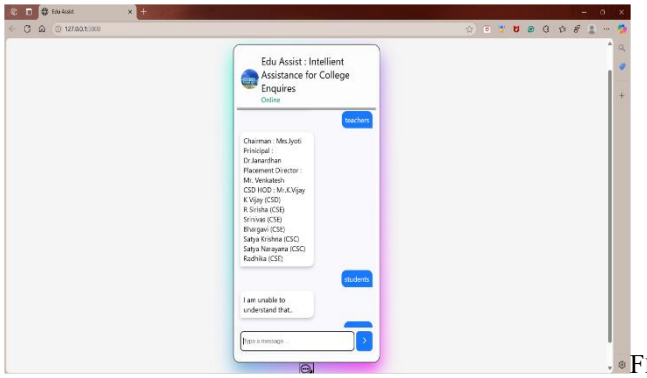

FIG. 10. ENQUIRIES

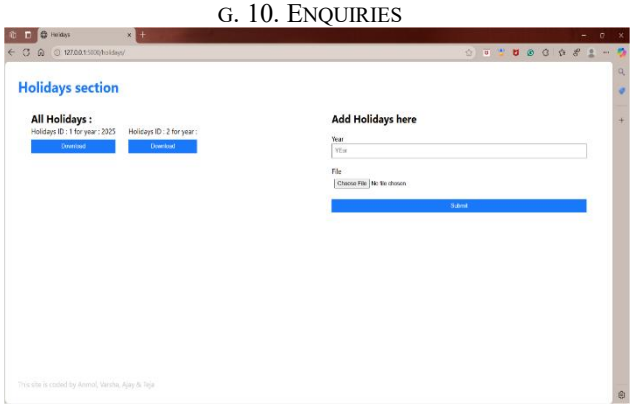

FIG. 11.HOILDAYS' PORTAL

FUTURE SCOPE

Future developments could include enhanced personalization, multilingual support, integration with AR/VR for virtual campus tours, and advanced analytics for predictive insights. These enhancements aim to further improve the chatbot's effectiveness and user engagement.

Enhanced personalization could involve integrating the chatbot with student databases to provide tailored responses based on individual profiles. Multilingual support would cater to a diverse student body, enabling interactions in various languages. The integration of AR/VR technologies could offer immersive virtual tours, enhancing the user experience for prospective students. Advanced analytics could provide insights into common student concerns, enabling institutions to address these proactively and improve their services.

REFERENCES

1.Rasa Framework-Based Chatbots:
Research by Windiatmoko et al. highlights the use of the Rasa framework to create a chatbot tailored for university inquiries. This system uses deep learning to handle queries about admissions, programs, and facilities, enhancing accessibility and interaction through platforms like Facebook Messenger. The Rasa framework's modular architecture makes it efficient for handling diverse university-related queries.

2. Kumar, R. D., Prudhviraj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 Through Intensive Investigation with Supervised Machine Learning Algorithm. In Handbook of Artificial Intelligence and Wearables (pp. 145-158). CRC Press.

3 Ache, T., Kumar, R. D., Senthilkumar, P., & Priyanka, V. (2024, June). Security: Cloud Data with Dynamic Protection Via Network Coding. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.
IBM Watson and NLP

4 Techniques:Studies such as those by Bhavika Ranoliya et al. have explored chatbots for university FAQs using IBM Watson. These systems leverage Natural Language Processing (NLP) for effective interaction, allowing users to resolve common queries related to courses, facilities, and procedures quickly.

5. Resource Booking Chatbots: Chatbots like the one proposed by CB Ram Mohan et al. streamline university resource management by allowing students and staff to reserve facilities or equipment through conversational interfaces, reducing administrative overhead

6. Proposed Systems and User-Friendly Design: Other studies have focused on reducing the workload for college staff and simplifying user interactions by integrating comprehensive databases with chatbot interfaces that mimic human interaction. These systems often include fallback mechanisms, such as notifying users when a representative needs to follow up manually.

7. Eaglebot with BERT: Eaglebot integrates the BERT model to create a multi-tier system for answering complex questions by retrieving data from heterogeneous sources. This approach showcases the application of advanced NLP in educational chatbots to address nuanced queries