Smart Campus Assistant: Chatbot Development for DR.TTIT College Website

Ms. Preethi S, Ms.Swetha N, Akash S, Anirudh N, Likith M, Mallikarjuna H K
Department of CSE, Dr.T. Thimmaiah Institute of Technology
KGF, India

Abstract: In educational institutions, addressing frequent student queries and providing timely information is essential for an efficient and responsive campus environment. Traditional methods such as manual inquiry handling and static web pages are often time-consuming and lack real-time interactivity. This project addresses the need for an intelligent and user-friendly communication system by developing a Smart Campus Assistant chatbot for the DR. TTIT College website. The chatbot leverages Python-based natural language processing and integrates with Django templating and JavaScript for a seamless web interface. It is capable of responding to a wide range of queries, including course details, admission procedures, department contacts, and more. The system also includes additional features like joke delivery, math problem solving, and basic web search to enhance engagement. Designed for real-time operation and easy scalability, this chatbot supports the goals of digital transformation in education by reducing staff workload and improving student satisfaction.

Keywords: smart campus, chatbot, DR. TTIT, college website, natural language processing, Django, real-time response, student queries, Python, education technology

I. INTRODUCTION

College life can be overwhelming, especially for new students trying to figure out how things work — from course details to admission procedures, fee structures, department contacts, and more. Traditionally, students would either have to look through multiple web pages, send emails, or physically visit the administration to get answers. This often results in delays, confusion, and increased workload on college staff. With the increasing number of queries, it's clear that a smarter and faster system is needed.Navigating college websites can often be frustrating for students, faculty, and visitors. Critical information—such as exam schedules, faculty contacts, event updates, or admission procedures—is sometimes buried under layers of menus or scattered across multiple pages. Traditional solutions like FAQs or static search bars are limited; they require users to sift through irrelevant content or struggle with vague keyword searches. For a busy student or an overwhelmed newcomer, this inefficiency wastes time and creates unnecessary stress.

This is where modern technology can bridge the gap. With advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP), chatbots have emerged as intuitive tools to deliver instant, conversational assistance.

Our project aims to develop a lightweight, cost-effective chatbot tailored to the college's needs. Built using accessible frameworks like Dialogflow, Rasa, or Python's NLTK, the chatbot will integrate seamlessly with the college website. The chatbot can handle a range of tasks — from providing course and department details to answering general campus-related questions.

II. RELATED WORK

Jordi Cabot et al. [1] A study using a multi model low code platform that enables users to build the chatbots through minimal coding and some no coding at all, followed by T & P evaluations. It empowers non-technical users to create complex chatbots easily, accelerating development and fostering innovation. It may limit customization and scalability for advanced developers due to platform constraints and low code abstractions.

Justin Thomas et al. [2] Tried a mixed method experimental design that will be used, where participants with assessed Big Five personality traits interact with chatbots exhibiting varied personalities. It helps optimize chatbot design by enhancing user trust, engagement and satisfaction through personalized, personality-aligned interactions. Sometimes findings may lack generalizability due to controlled settings and the complexity of accurately modelling human-like personality in chatbots.

Ibrahim Arpachi et al. [3] A hybrid methodology using SEM to validate relationships and ANN to model non-linear impacts of cybersecurity and protection motivation factors on chatbot social sustainability. It offers insights by combining statistical validation with predictive modeling to capture both relationships. it requires large datasets and complex analysis,making it resource intensive and potentially difficult to interpret for non-experts

Maria Virvou and team [4] An experimental study where participants interact with chatbots of varying personalities followed by surveys and statistical analysis to measure the impact of personality congruence on user behavior. it provides insights into improving chatbot-user interactions by tailoring chatbot personalities to match user preferences, enhancing satisfaction and engagement. it may be limited by subjective biases and challenges in accurately simulating diverse personality traits in chatbot behavior

Ranci Ren et al. [5] where participants used SOCIO chatbot for UML with data collected through performance metrics and questionnaires to evaluate. It enables systematic evaluation of chatbot assisted UML modeling, highlighting improvements in learning, usability and modeling efficiency. it may face limitations in capturing the complex modeling requirements and relies on participant variability, affecting result consistency.

Bilal Babayaigit and team [6] Trained on turkish conversational data and evaluates via performance metrics and user interactions feedback. It enables natural and context aware Turkish conversations advancing local language chatbot development with deep learning. it requires large,high quality Turkish datasets and may struggle with handling out-of-domains or complex queries.

Pegah Safari et al. [7] Chinese BERT-based model trained on annotated anti-drug data to perform cognitive intent analysis with evaluation via classification accuracy and user feedback. It provides accurate and context aware intent detection enhancing the chatbot's ability to deliver effective anti-drug education and counseling. it relies on large, high quality annotated datasets and may struggle with interpreting nuanced or ambiguous user intent in real world scenarios.

III. METHODOLOGY

In this project, we focused on building a smart and simple chatbot system to assist users on the DR.TTIT college website. The goal was to make college information easily accessible without needing to browse through multiple web pages.

We used Django as the core backend framework to manage routing, serve the chatbot interface, and process user queries. On the frontend, we designed a sleek and responsive UI using HTML, CSS, and JavaScript, embedding the chatbot directly into the official college website.

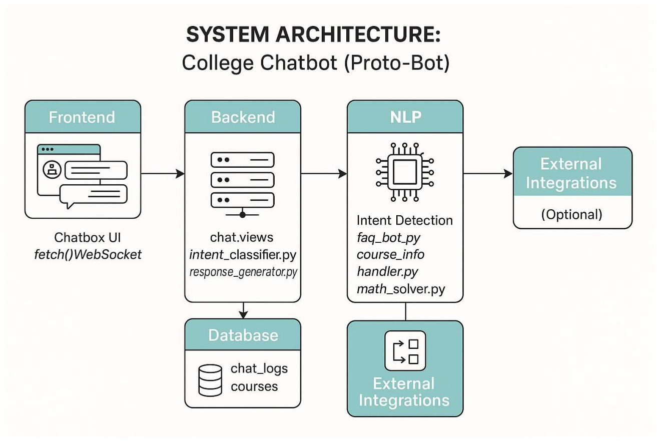


Fig.1: System Architecture

The logic behind the chatbot was initially rule-based — meaning we matched keywords and specific phrases to pre-written responses. This worked well for frequently asked questions like "How to apply?" or "What courses are offered?" The chatbot could also help with navigating the campus, finding department contact information. Everything was designed to keep the experience light and quick, so users could get help without delays.

A. Requirement Analysis

The first step involved identifying the most common queries received by students and visitors. These included questions about admissions, available courses, departments, faculty contacts, campus facilities, fee structures, and placement opportunities. A questionnaire and informal interviews were conducted with students and staff to gather relevant information and use cases.

We have gathered the required amount of data from students, faculty, and administrative staff and from various sources to identify common queries. The collected data is further classified into different categories. choosing a platform between a rule-based or AI driven based on college needs.

B. Design and Planning

Based on the identified requirements, a modular architecture was designed to ensure the system's scalability, maintainability, and ease of integration. The frontend interface of the chatbot was developed using HTML, CSS, and JavaScript to provide an interactive and user-friendly experience. This interface was embedded directly into the college website allowing users to access the chatbot seamlessly. On the backend, Python was chosen as the core programming language for handling user inputs, processing queries, and generating appropriate responses. To support the development of a robust web application, the Django framework was utilized.

Django effectively managed backend operations, including routing, template rendering, and integrating the chatbot with the website's existing infrastructure. This combination of technologies ensured that the chatbot could operate in real-time while maintaining responsiveness and modularity.

C. Data Collection & Preprocessing

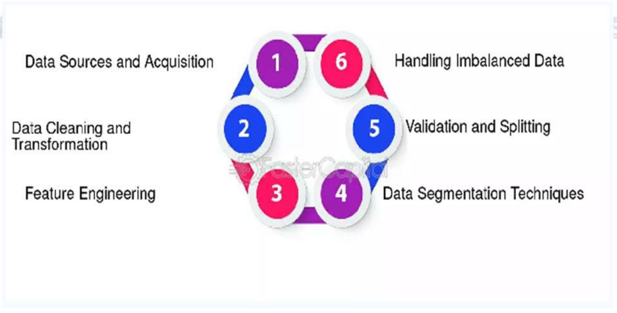

1 Data Sources and Acquisition
2 Data Cleaning and Transformation
3 Feature Engineering
4 Data Segmentation Techniques
5 Validation and Splitting
6 Handling Imbalanced Data

Fig.2: required data to be processed

The foundation of an effective chatbot lies in robust data collection and preprocessing. To build the Smart Campus Assistant, We first compiled a comprehensive dataset from existing FAQs, college handbooks, and official website content, structuring it into a searchable knowledge base. This ensures the chatbot can address common academic, administrative, and event-related queries. Additionally, we analyzed historical user query logs (where available) from the college website to identify frequently

ISSN [Online]: 2583-2654

asked questions and prioritize them during training. To prepare the data for NLP processing, we performed text cleaning to remove irrelevant characters, correct typos, and standardize formats

D. Natural Language Processing

To process user inputs effectively, a rule-based Natural Language Processing (NLP) model was implemented in the initial phase of the project. This model relied on identifying specific keywords and phrase patterns within the user's queries and matching them to a set of predefined responses. The chatbot was designed to handle a variety of common campus-related interactions.

These included answering frequently asked questions such as "How to apply for admission?", providing course-related information like "What courses are offered in Computer Science?", and offering department contact details. Additionally, the system supported navigation assistance to help users locate facilities within the campus, such as "Where is the library?".

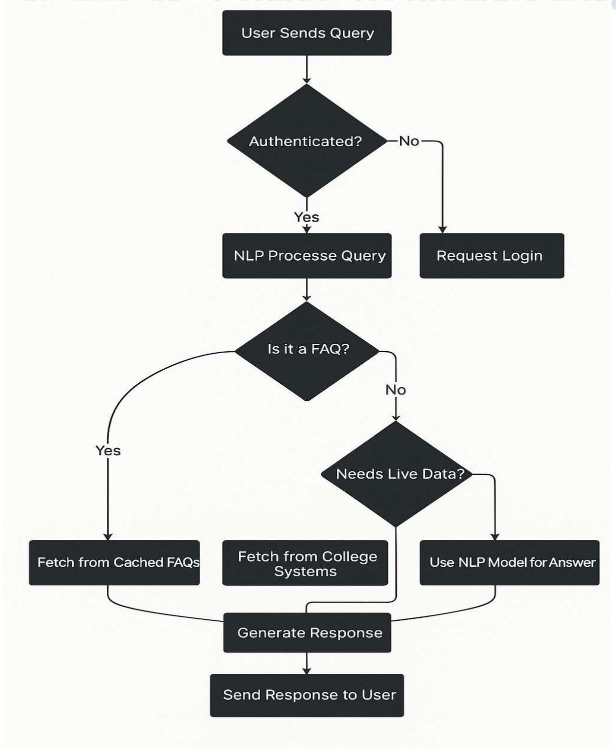


Fig 3: logic Flow Diagram

E. Model Training & Testing

To enable the chatbot to understand and respond intelligently to user queries, we implemented a structured model training and testing process. First, we performed intent labeling, where sample queries like

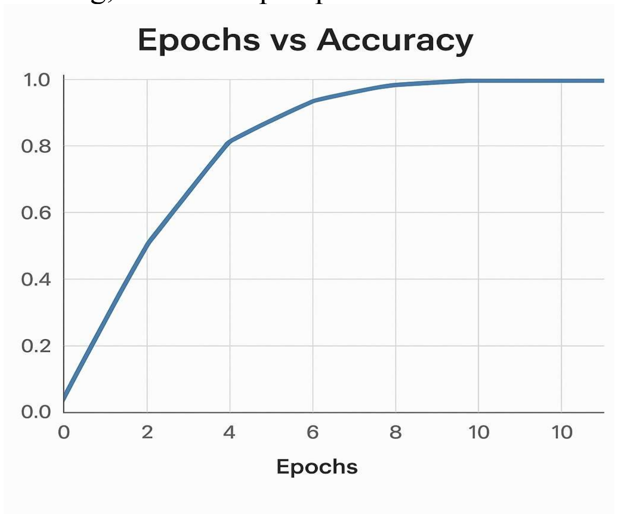


Fig 4: graph based on accuracy of code

"How to apply for scholarships?" were manually tagged with corresponding categories to train the NLP model on recognizing user intent. This labeled dataset was then fed into Dialogflow/Rasa, where the AI engine learned to classify questions into predefined intents and entities.

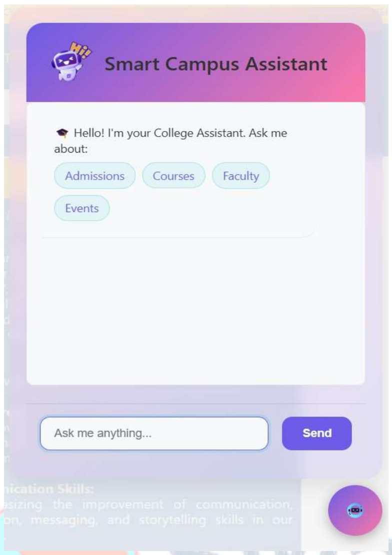
During the testing phase, we validated the model's accuracy using a separate set of test queries, refining responses to ensure it could handle variations in phrasing.This iterative process helped optimize the chatbot's ability to deliver precise and context-aware answers before deployment.

F. Deployment & Integration

The chatbot was tested using real queries collected during the requirement phase. The phases include Response accuracy, Load time, User experience, System compatibility.

The chatbot was integrated into the Django web framework, where backend logic was managed through views and templates, while AJAX and JavaScript facilitated real-time messaging without page reloads. The final deployment featured the chatbot as a floating/fixed widget, ensuring visibility and usability across the college website's main sections.

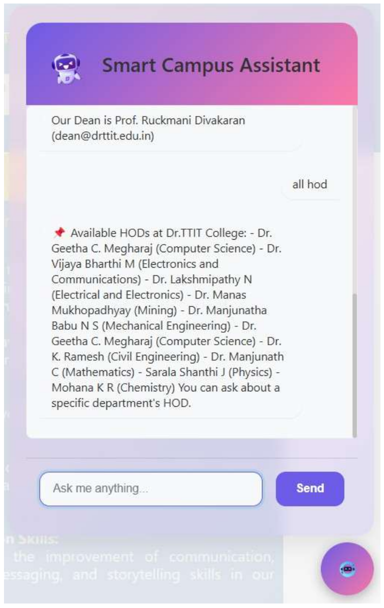
IV. RESULT



🎓 Hello! I'm your College Assistant. Ask me about:

Fig 5: Base Template

The above diagram represents the base template for the project we have been working on



Our Dean is Prof. Ruckmani Divakaran (dean@drttit.edu.in)

all hod

📌 Available HODs at Dr.TTIT College: - Dr. Geetha C. Megharaj (Computer Science) - Dr. Vijaya Bharthi M (Electronics and Communications) - Dr. Lakshmipathy N (Electrical and Electronics) - Dr. Manas Mukhopadhyay (Mining) - Dr. Manjunatha Babu N S (Mechanical Engineering) - Dr. Geetha C. Megharaj (Computer Science) - Dr. K. Ramesh (Civil Engineering) - Dr. Manjunath C (Mathematics) - Sarala Shanthi J (Physics) - Mohana K R (Chemistry) You can ask about a specific department's HOD.

Fig 6: Model Testing and Training

V. CONCLUSION

This whole thing started because we noticed a common issue students and visitors face — not knowing where to go or who to ask for information on the college website. Instead of building something overly complex or relying on third-party platforms, we decided to create a simple, in-house chatbot solution. We chose to use Python for the backend and Django for easy integration with the existing DR.TTIT website. For the interface, we went with basic HTML, CSS, and JavaScript to keep it lightweight and accessible. The chatbot uses a rule-based system to match user questions with predefined responses. It's not perfect — it doesn't understand everything — but for most common questions like admissions, courses, or department contacts, it does the job well.

VI. FUTURE WORK

The current version of the chatbot effectively addresses common student queries related to admissions, courses, departments, and general college information. However, there are several directions in which the system can be enhanced in the future. One key improvement would be the integration of voice input and output, making the chatbot more accessible, especially for visually impaired users or those who prefer voice interactions.

Another important enhancement would be the inclusion of multilingual support, allowing users to communicate with the chatbot in regional languages such as Kannada, Hindi, or Tamil, thereby increasing inclusivity. Developing an admin dashboard could enable staff members to dynamically update chatbot responses— such as event schedules or deadlines— without the need for technical changes in the codebase.

REFERENCES

[1] GWENDAL DANIE; JORDI CABOT; LAURENT DERUELLE;MUSTAPHA DERRASXatkit: A

Multimodal Low-Code Chatbot Development Framework. J. Neotrop,2021, 1, e2966919.

[2] Mohammad Amin Kuhail; Mohamed Bahja; Ons Al-Shamaileh; Justin Thomas; Amina Alkazemi; Joao Negreiros Assessing the Impact of Chatbot-Human Personality Congruence on User Behavior.Eng. 2024, 38, 71761 - 71782.

[3] Maria Virvou; George A. Tsihrintzis;Wang, X.N. Artificial Intelligence Chatbots and Social Robots in Education. FEPER Framework for Efficiency. 2024, 58, 17-19.

[4] Ibrahim Arpaci.A Multianalytical SEM-ANN Approach to Investigate the Social Sustainability of AI Chatbots Based on Cybersecurity and Protection Motivation Theory. 2023, 1714 - 1725, 101183. [CrossRef]

[5] Ranci Ren;John W.;Castro Adrián Santos;Oscar DSilvia T. Acuña. Using the SOCIO Chatbot for UML Modelling: A Family of Experiments,2021, c9d5-5390, 1435. [CrossRef]

[6] Bilal Babayigit; Habibelahi Rahmani; Mohammed Abubaker; TrBot: A Turkish Deep Learning Chatbot Utilizing Seq2Seq Model. Trans. Chin. Soc. Agric. Mach. 2025, 43, 49552 - 49566.

[7]G.K.SANTHOSHRAM;VMUTHUMANIKANDA

N;Visistant: A Conversational Chatbot for Natural Language to Visualizations With Gemini Large Language Models implemented. October. 2024.3465541

[8] Jui-Hsuan Lee; Eric Hsiao-Kuang Wu; Yu-Yen Ou; Yueh-Che Lee; Cheng-Hsun Lee; Chia-Ru Chung;Anti-Drugs Chatbot: Chinese BERT-Based Cognitive Intent Analysis, Chin. Soc. Agric. Eng. 2023, 36, 514 - 521.

[9] ÁNGEL ANTONIO MARTÍNEZ-GÁRATE;JOSÉ ALFONSOAGUILAR-CALDERÓN;CAROLINA;TR IPP-BARBA;ANÍBAL;ZALDÍVAR-COLADO;Model Driven Approaches for Conversational Agents Development: A Systematic Mapping Study Chin. Soc. Agric. Eng. 2023, 3293849,144–151.