SMART CAMPUS BOT

Ballem SahithiSriya[1], Navanitha[2], Pradyumn[3], Mrs.K.Kusumalatha[4]

[1] UG student Dept. of Electronics and Computers Engineering Sreenidhi Institute of science and Technology,Telangana,India.

[2] UG student Dept. of Electronics and Computers Engineering Sreenidhi Institute of science and Technology,Telangana,India.

[3] UG student Dept. of Electronics and Computers Engineering Sreenidhi Institute of science and Technology,Telangana,India.

[4] Asst.Professor Dept. of Electronics and Computers Engineering Sreenidhi Institute of science and Technology,Telangana,India.

Abstract-A chat-bots aims to make a conversation between both human and machine.The machine has been embedded knowledge to identify the sentences and making a decision itself as a response to answer a question. Chat-bots will becompletely based on a text-based user interface, allowing the user to type commands and receive text as well as text to speech response.Chat-bots are usually stateful services, remembering previous commands in order to provide functionality. It can be utilized securely by an even largeraudience when chat-bots technology is integrated with popular web services. The college inquiry chat-bots will be built using artificial algorithms that analyze user's queries and understand user's message.The response principle is matching the input sentence from a user. The User can ask the question any college-related activities through the chat-bot without physically available to the college for inquiry. The System analyses the question and then answers to the user. With the help of artificial intelligence, the system answers the query asked by the students. The system replies using an effective Graphical User Interface as if a real person is talking to the user. The user just has to register himself to the system and has to login to the system. The chatbots consists of core and interface that is accessing in the Telegram by Ngrok. Natural language processing technologies are used for parsing, tokenizing, stemming and filtering the content of the complaint.

Keywords— "Anaconda Prompt","College System,"Placement Information", "Python", "Rasa", "Telegram User Interface"

I.INTRODUCTION

The integration of technology within educational institutions has revolutionized traditional methods of student engagement and administrative processes. In this digital era, chatbots have emerged as powerful tools for enhancing communication and efficiency in college management systems. This report focuses on the development and implementation of a Telegram-based chatbot tailored specifically for college management tasks, utilizing the Rasa Python framework within the Anaconda environment.

The chatbot serves as a virtual assistant, providing students with instant access to a wide range of services and resources, including course information, exam schedules, academic support, and administrative assistance. By leveraging Natural Language Processing (NLP) algorithms, the chatbot offers an intuitive interface for users to interact with college services seamlessly.

Through an exploration of the chatbot's architecture and functionalities, this report aims to evaluate its effectiveness in improving user experience and operational efficiency within the college ecosystem. Additionally, it discusses potential implications and future avenues for integrating chatbot technology more extensively into college management systems.

In the ever-evolving landscape of educational institutions, the integration of technology has become imperative for efficient management and seamless operations. Recognizing this need, the Smart Campus Bot emerges as a pivotal solution in the realm of college management systems. Developed through the collaborative efforts of Rasa, Anaconda, Python, and a dedicated team, this innovative bot aims to streamline administrative tasks, enhance communication channels, and enrich the overall learning experience within educational institutions.

The convergence of Rasa's natural language processing capabilities, Anaconda's comprehensive data science platform, and Python's versatile programming language empowers the Smart Campus Bot with unparalleled functionality and adaptability. By leveraging these cutting-edge technologies, the bot serves as a dynamic interface between students, faculty, and administrative enrollment and timetable management to resource allocation and event coordination.

Through this report, we delve into the architecture, features, and implementation of the Smart Campus Bot, shedding light on its transformative potential in revolutionizing college management systems. From its inception to deployment, we examine the intricacies of its development process, highlighting key milestones, challenges, and future prospects.

Join us on a journey through the realms of innovation and efficiency as we explore the capabilities of the Smart Campus Bot and its role in shaping the future of educational institutions. In the ever-evolving landscape of educational institutions, the integration of technology has become imperative for efficient management and seamless operations. Recognizing this need, the Smart Campus Bot emerges as a pivotal solution in the realm of college management systems. Developed through the collaborative efforts of Rasa, Anaconda, Python, and a dedicated team, this innovative bot aims to streamline administrative tasks, enhance communication channels, and enrich the overall learning experience within educational institutions.

The convergence of Rasa's natural language processing capabilities, Anaconda's comprehensive data science platform, and Python's versatile programming language empowers the Smart Campus Bot with unparalleled functionality and adaptability. By leveraging these cutting-edge technologies, the bot serves as a dynamic interface between students, faculty, and administrative staff, facilitating a myriad of tasks ranging from course enrollment and timetable management to resource allocation and event coordination.

Through this report, we delve into the architecture, features, and implementation of the Smart Campus Bot, shedding light on its transformative potential in revolutionizing college management systems. From its inception to deployment, we examine the intricacies of its development process, highlighting key milestones, challenges, and future prospects.

Join us on a journey through the realms of innovation and efficiency as we explore the capabilities of the Smart Campus Bot and its role in shaping the future of educational inst

II.LITERATURE SURVEY

According to the work of Polignano et al., Health Assistant Bot is a conversational AI that can learn a patient's symptomatology through talk, refer them to specialists, and monitor their medications and other health data. With the bot, the consumer may have a conversation in natural language to learn about their health, describe their symptoms, find a physician, and discuss potential solutions. Using AI methods for side effect management and, by extension, diagnosis, is a novel part of their procedure. A recommendation algorithm then analyses the data to find the nearest specialist who can treat the client's problem based on the client's location and medical history. Throughout the course of the brainstorm, we put this Health Assistant Bot through its paces both offline and online. In the first, it was utilized to determine if any hidden parts were there, and in the second, it analyzed information from 102 people who had frequent conversations with the expert. These results confirmed the system's viability as a tool for improving people's health. The major shortcoming of this bot is that it does not provide in-depth explanations of diseases and symptoms, nor does it report the possibility that a certain ailment can produce a given symptom. Computer-based illness symptom prediction is difficult and error-prone due to a lack of descriptive data.[1]

Patelet et al. have created UNIBOT, an AI-powered and web-based interactive university chat-bot, which places a premium on client involvement and is accessible from anywhere. With very minor tweaks, the Chat-bot might be incorporated into any scholarly website. The database has a wealth of data that may be linked back to individual students or the University as a whole. The Chat-bot is available to anyone with access to the University's website. This effort makes use of ML and AI technologies. The PHP programming language is widely used in the chat-bot development industry. The user inputs a query on a subject covered in college, and the Bot returns an answer based on a comparison of the two. The interface is very similar to that of a chat app. For present and future college students, this is a great way to save time and energy when researching topics like tuition rates, registration dates, and more. It has been argued that UNIBOT does not provide a sufficient amount of information on universities. The chat-bot can be improved by using Natural Language Processing (NLP).[2]

The college inquiry chat-bots will be built using artificial algorithms that analyze user's queries and understand user's message. The response principle is matching the input sentence from a user. The User can ask the question any college-related activities through the chat-bot without physically available to the college for inquiry. The System analyses the question and then answers to the user. With the help of artificial intelligence, the system answers the query asked by the students. The system replies using an effective Graphical User Interface as if a real person is talking to the user. The user just has to register himself to the system and has to login to the system. The chat-bots consists of core and interface that is accessing the core in (MySQL).Natural language processing technologies are used for parsing, tokenizing, stemming and filtering the content of the complaint[3]

The user can have a conversation with a chatbot embedded in the site, which will then provide the necessary data. Attendance, personal details, and grades are just some of the data it provides. The administrators need to compile data relevant to the educational institutions in an Excel document. This Excel document is compatible with database information since it follows a standard format. Additionally, an Excel-to-database Python script has been provided. This new information will be used by the chatbot.[4]

III.METHODOLOGY

We chose the Rasa Python framework within the Anaconda environment, leveraging its natural language processing (NLP) capabilities. Develop modules for interpreting user queries, retrieving data from the college database, processing queries, and generating responses. Integrate with the Telegram platform for user interaction and deploy the chatbot on a server with access to the college database. Test extensively to ensure functionality, accuracy, and security, and provide ongoing maintenance and updates as needed**.**Begin by setting up the development environment using Anaconda Prompt, a command-line interface for managing Python environments. Install Anaconda, which simplifies package management and dependency handling. Create a new virtual environment to isolate dependencies for the bot project, ensuring compatibility and avoiding conflicts with other Python projects.

Chatbot's are generally used to respond quickly to users. Chatbot's, a common name for automated Once the environment is set up, install the Rasa Open Source framework using Anaconda Prompt. Utilize pip, the Python package installer, to install Rasa within the virtual environment. This provides access to the necessary tools and libraries for building conversational AI applications.[6]
Project Initialization: Initialize a new Rasa project[7] by creating a new directory for the bot project. Use the rasa init command in Anaconda Prompt to generate the basic structure and files required for the bot project. This includes configuration files, directories for training data, and predefined templates.
Domain and Training Data: Define the bot's domain by specifying intents, entities, actions, and responses in the domain.yml file. Prepare training data in the form of stories and NLU (Natural Language Understanding) data to train the bot's machine learning models. Organize training data to cover various user intents and scenarios the bot will encounter.
Training the Bot: Train the bot's NLU and dialogue management models using the rasa train command in Anaconda Prompt. This process involves training machine learning models based on the provided training data to understand user inputs and generate appropriate responses.
Implement custom actions in Python to handle business logic and external API integrations. Define custom actions in the actions.py file and configure endpoints for the Rasa server to communicate with the custom actions server. Implement error handling and validation logic to ensure robustness and reliability of custom actions.
8. Integration: Integrate the bot with external systems and databases using Python scripts and APIs. Use Anaconda Prompt to install additional Python packages and libraries as needed for integration purposes. Implement secure authentication mechanisms and data encryption to protect sensitive information and ensure secure communication with external systems

Our Proposed System:

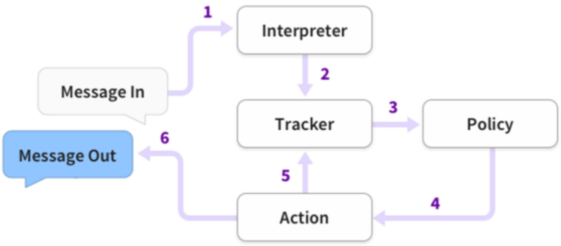
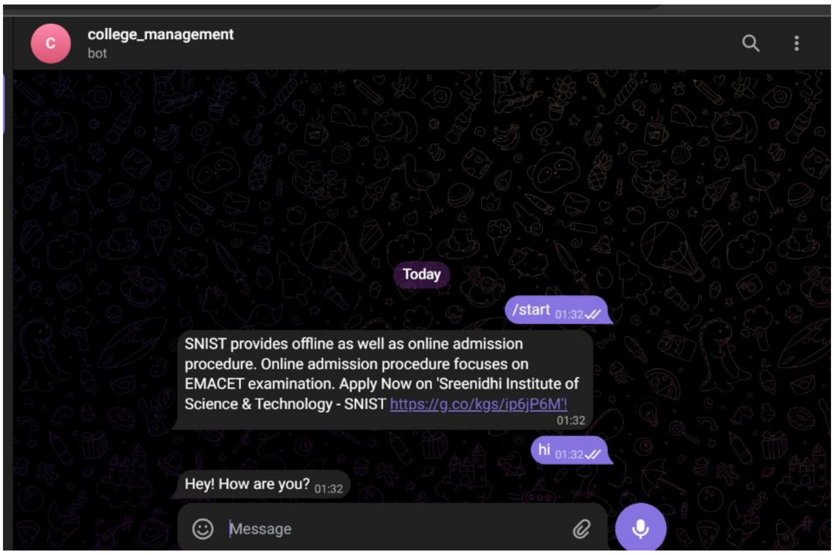


Fig3.1: block diagram for proposed system

From the knowledge gained from the literature survey of existing survey we have decided to propose a system with the features that overcome the limitations of the existing system and develop a chatbot for college students using Rasa.
The proposed system entails the development of a sophisticated chatbot leveraging Anaconda, Rasa, Python, and Telegram interface. Anaconda serves as the foundation for managing Python environments and requisite packages, ensuring a seamless development process. Rasa, an open-source conversational AI platform, empowers the chatbot with advanced capabilities, including natural language understanding, intent recognition, and response generation. Python, renowned for its versatility and ease of use, facilitates the implementation of custom logic and responses, enabling the chatbot to adapt to various user interactions effectively. Integration with the Telegram interface extends the reach of the chatbot to a wider audience, enhancing user engagement and accessibility. By combining these components, the proposed system aims to deliver a robust and versatile chatbot solution capable of meeting the evolving needs of diverse user bases across different domains.

ADVANTAGES OF PROPOSED SYSTEM:

- User friendly
- Save the student time
- Open Source Network
- Offers 24/7 Benefit
- Instant Responses
- Personalized Assistance
- Improved Communication
- Cost-Efficiency



/start 01:32

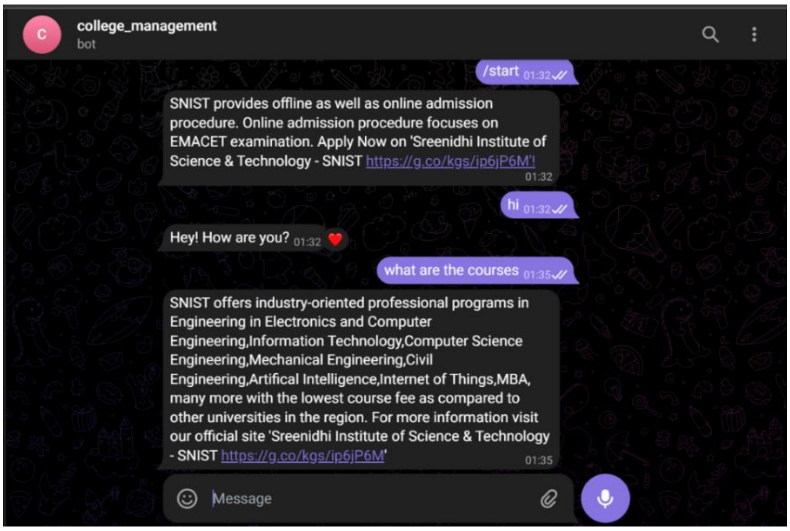
SNIST provides offline as well as online admission procedure. Online admission procedure focuses on EMACET examination. Apply Now on 'Sreenidhi Institute of Science & Technology - SNIST https://g.co/kgs/ip6jP6M'! 01:32

hi 01:32

Hey! How are you? 01:32

Fig:3.1 : Starting of campus bot

Fig:3.1 : We can chat with the smart campus bot with the telegram user interface.



/start 01:32

SNIST provides offline as well as online admission procedure. Online admission procedure focuses on EMACET examination. Apply Now on 'Sreenidhi Institute of Science & Technology - SNIST https://g.co/kgs/ip6jP6M'| 01:32

hi 01:32

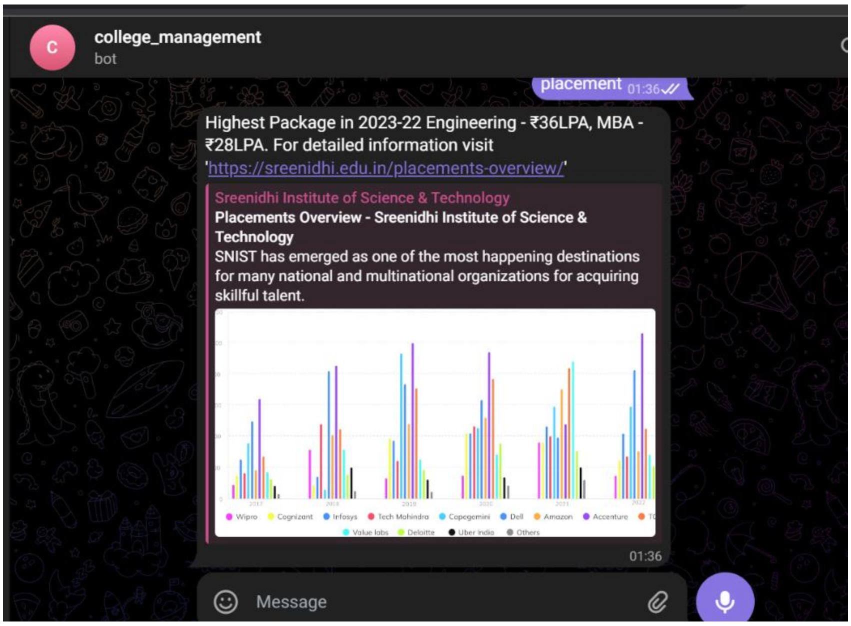
Hey! How are you? 01:32 ❤️

what are the courses 01:35

SNIST offers industry-oriented professional programs in Engineering in Electronics and Computer Engineering,Information Technology,Computer Science Engineering,Mechanical Engineering,Civil Engineering,Artifical Intelligence,Internet of Things,MBA, many more with the lowest course fee as compared to other universities in the region. For more information visit our official site 'Sreenidhi Institute of Science & Technology - SNIST https://g.co/kgs/ip6jP6M' 01:35

Fig3.2:Smart Campus bot

Fig 3.2: Smart campus bot is responding to the student about the Courses information.



placement 01:36

Highest Package in 2023-22 Engineering - ₹36LPA, MBA - ₹28LPA. For detailed information visit 'https://sreenidhi.edu.in/placements-overview/'

Sreenidhi Institute of Science & Technology
Placements Overview - Sreenidhi Institute of Science & Technology
SNIST has emerged as one of the most happening destinations for many national and multinational organizations for acquiring skillful talent.

01:36

Fig3.3:Smart Campus Bot output

Fig: 3.3 represents the Smart Campus bot responding to user about placements information.

IV.CONCLUSION AND FUTURE SCOPE:

CONCLUSION:

In conclusion, the development of a Smart Campus Bot represents a significant advancement in leveraging technology to enhance the efficiency, accessibility, and engagement within educational institutions. By harnessing the power of conversational AI through platforms like Rasa, educational institutions can streamline administrative processes, provide instant support to students and faculty, and facilitate seamless communication across various departments and services. Through the implementation of comprehensive software testing strategies, including unit testing, integration testing, system testing, and acceptance testing, developers can ensure that the Smart Campus Bot meets the highest standards of quality, reliability, and security. This rigorous testing process helps identify and address any potential issues or deficiencies, ensuring that the bot delivers a seamless and user-friendly experience to its users.

Furthermore, the continuous evolution and improvement of the Smart Campus Bot, supported by regular updates, feedback loops, and data-driven insights, enable educational institutions to adapt to changing needs and technological advancements effectively. By embracing innovation and prioritizing the development of intelligent conversational interfaces, educational institutions can stay at the forefront of digital transformation and provide a modern and personalized experience to students, faculty, and staff members.

In summary, the Smart Campus Bot represents a valuable asset for educational institutions, offering enhanced accessibility, efficiency, and engagement while fostering innovation and digitalization in the education sector. Through careful development, testing, and continuous improvement, the Smart Campus Bot has the potential to revolutionize the way educational institutions operate and interact with their stakeholders in the digital age.
Our proposed system provides and interactive method to view the college student data by using a chatbot. It is easy to use as, it can be integrated with Telegram messaging app.It provides the data from excel sheet and uses python script to update the database with reference to the excel file. It provides features like student attendance, student grades, student details, recentannouncements, placement information.
.

FUTURE SCOPE:

Based on the system development and extensibility in future can also implement the Voice recongination and face detection .

V.REFERENCES

[1] M. Polignano, F. Narducci, A. Iovine, C. Musto, M. De Gemmis and G. Semeraro, "HealthAssistantBot: A Personal Health Assistant for the Italian Language," in IEEE Access, vol. 8,2020

[2] N. P. Patel, D. R. Parikh, D. A. Patel and R. R. Patel, "AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT)," 2019 3rd International conference on Electronics,

Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019
[3]https://www.w3schools.com/bootstrap4/default.as p

[4]https://www.w3schools.com/python/default.asp.

Agnese Augello, Giovanni Pilato, Alberto Machi' ICAR - Istituto di Calcolo e Reti ad Alte Prestazioni CNR - Consiglio Nazionale delle Ricerche Viale delle Scienze , 978-0-7695-4859- 3/12$26.00 © 2012 IEEE . "An Approach to Enhance Chatbot Semantic Power and Maintainability: Experiences within the FRASI Project".
[3] https://www.ubisend.com/features

[4] https://sentione.com/resources/whitepapers/chatbo t-guide

[7]https://www.techechelons.com/blog/developmentscope-in-chatbotsapplications#:~:text=With%20efficient%20chatbot%20development%20practices,they%20need%20is%2 0a%20stimulus

[8]https://searchenterpriseai.techtarget.com/feature/The-future-scope-of-chatbots-begins-withaddressing-flaws

[9] https://cobusgreyling.medium.com/eight-thingsdifferentiating-rasa-from-other-chatbot-platforms5f9531690df1

[10] https://medium.com/voice-tech-podcast/how-tocreate-chatbot-using-rasa-82954e141ae7