iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60961

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

AI Based Multilingual College Enquiry Voice Bot Using Python

Prof. Kotresh H M[1], Deeksha Srikanth[2], Devika V[3]
[1]*Associate Professor, Electronics and Communication, East West Institute of Technology, Bengaluru, India*
[2, 3]*Student, Electronics and Communication, East West Institute of Technology, Bengaluru, India*

Abstract: ***This paper addresses the growing need for seamless cross-language communication in our global society. Leveraging AI, it offers a versatile conversational agent capable of understanding and responding to inquiries in multiple languages. Through speech recognition and advanced NLP, the system interprets user intent and retrieves relevant information from a college database. Integrated language translation services enable communication in various languages namely-Kannada, English, Tamil and Telugu enhancing accessibility. Responses are dynamically translated into the user's preferred language using text-to-speech conversion. This project demonstrates AI's potential in bridging linguistic barriers and improving access to educational resources. Future enhancements could include personalized responses and expanded language support in five South Indian languages, further enriching the user experience and solidifying the bot's role as a valuable educational tool.***
Keywords: Artificial Intelligence, Multilingual, Natural language processing, Admission.

I. INTRODUCTION

In today's era defined by globalization and cultural diversity, effective communication across language barriers is paramount, particularly in education. However, traditional methods often fall short in meeting the needs of linguistically diverse populations. The "Multilingual College Inquiry Voice Assistant Bot using Python" project addresses this challenge by leveraging AI to create an accessible solution. It aims to develop a conversational agent capable of understanding user inquiries and providing college-related information in multiple languages. By utilizing speech recognition, users can interact naturally, streamlining the information-seeking process and enhancing accessibility.

At the core of the project lies advanced Natural Language Processing (NLP), enabling the bot to decipher user intent and extract key information effectively. This allows for tailored responses, addressing specific user needs, such as college locations, course offerings, and admission requirements. Integration with language translation services ensures seamless communication in various languages, fostering inclusivity and breaking down barriers to information access. Additionally, the bot interfaces with a comprehensive database, offering accurate and up-to-date information on colleges, academic programs, faculty details, and campus amenities.

The project embodies the transformative potential of AI-driven conversational agents in revolutionizing communication and information access in education. By providing a user-friendly, multilingual interface in five South Indian languages namely-Kannada, English, Tamil and Telugu for accessing college-related information, it empowers individuals from diverse linguistic backgrounds to make informed decisions about their educational journeys. Its significance extends beyond convenience, representing a fundamental shift in how educational information is disseminated and accessed, removing barriers of literacy and technological proficiency.

Moreover, the project holds implications for fostering cultural exchange and mutual understanding in the educational landscape. By facilitating communication across linguistic boundaries, it creates opportunities for collaboration, knowledge-sharing, and cross-cultural dialogue among students, educators, and institutions worldwide. In doing so, it enriches the educational experience and cultivates a more inclusive and interconnected global community. As we strive for a more equitable and accessible education system, the project exemplifies the transformative power of technology in shaping the future of learning, embodying the spirit of innovation, collaboration, and inclusivity in the quest for knowledge in the 21st century.

II. METHODOLOGY

The block diagram illustrates the working of the College Voice Bot in Figure 1. Integrating these modules into a cohesive framework, the proposed model aims to create a versatile voice assistant bot, empowering users to access college-related information effortlessly in multiple languages.

©IJRASET: All Rights are Reserved

1) *Speech Recognition Module:* Utilizes libraries like Speech Recognition or Google Cloud Speech-to-Text to convert spoken words into text, allowing natural language interaction with the bot.
2) *Natural Language Processing (NLP) Module:* Incorporates techniques like intent recognition, entity extraction, and sentiment analysis to understand and interpret user queries effectively.
3) *Language Translation Module:* Integrates services like Google Translate API or Microsoft Translator API to translate user queries and responses into multiple languages, enhancing accessibility.
4) *College Information Retrieval Module:* Retrieves college information from a database or API based on user input, including college profiles, courses offered, and admission requirements.
5) *Response Generation Module:* Generates appropriate responses, formats them, translates into the user's preferred language, and converts them into speech for seamless communication.
6) *User Interface Module:* Designs user-friendly interfaces including a command-line interface (CLI) for testing and a graphical user interface (GUI) for deployment in web or mobile applications.
7) *Error Handling Module:* Implements robust error handling mechanisms to manage unexpected inputs and errors, providing informative messages to guide users.
8) *Testing and Validation Module:* Conducts thorough testing to ensure functionality, accuracy, and reliability across various inputs, languages, and scenarios.
9) *Deployment Module:* Deploys the system in a production environment, hosting it on a server or cloud platform to handle user requests in real-time.
10) *Maintenance and Updates Module:* Regularly maintains and updates the system based on user feedback, monitoring performance for optimization and enhancement.

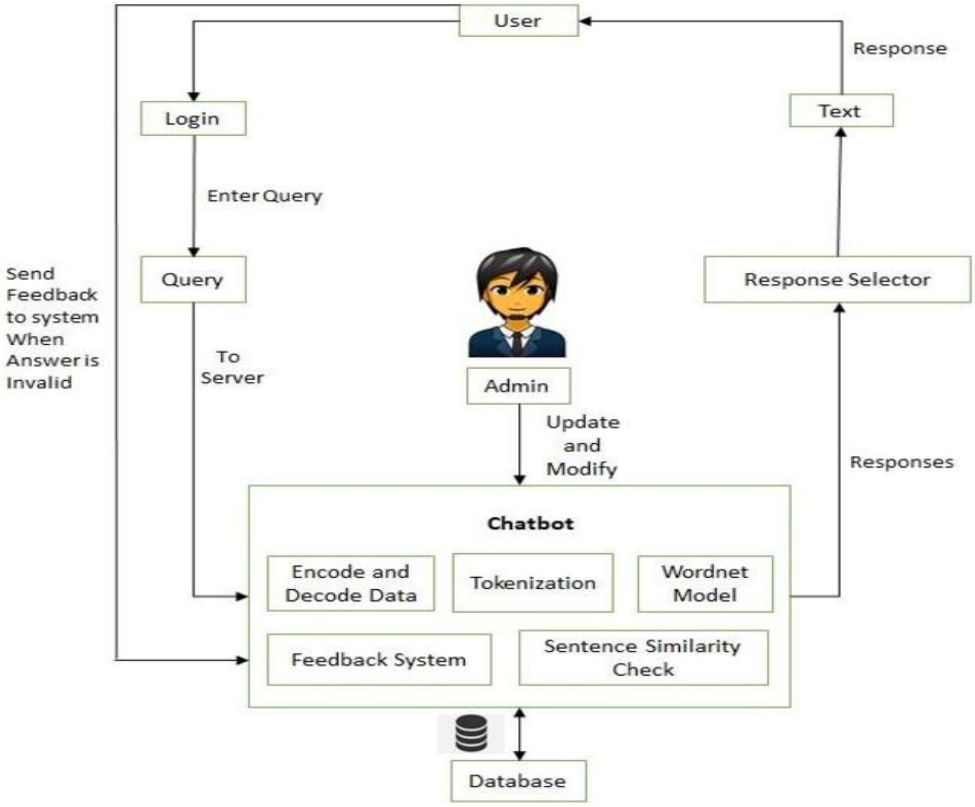


Fig. 1. Block Diagram of the College Enquiry Voice Bot.

III. DESIGN

Designing an effective system is inherently a creative process, where attention to detail and adherence to key design principles are paramount. For the development of the "AI-based Multilingual College Inquiry Voice Assistant Bot using Python," a robust design approach is essential to ensure its usability, scalability, and overall effectiveness. This involves considering various fundamental design concepts that underpin the system's architecture and functionality.

User-Centric Design lies at the heart of this endeavor. It emphasizes the importance of designing the system with the end-user in mind, prioritizing aspects such as ease of use, intuitiveness, and accessibility. Understanding the target users' needs, preferences, and technological proficiency is crucial for creating a design that resonates with them and enhances their overall experience.

Modularity and Scalability are key principles that drive the system's architecture. By adopting a modular design approach, the system becomes more flexible and adaptable to future enhancements, language support expansions, and the integration of additional features. Breaking down the system into smaller, independent modules promotes flexibility and ease of maintenance.

Natural Language Understanding (NLU) forms the backbone of the system's functionality. Robust natural language processing (NLP) techniques are essential for accurately interpreting and understanding user queries. This involves sophisticated entity extraction, intent recognition, sentiment analysis, and handling variations in language structure and syntax to ensure precise responses. Multilingual Support is a critical aspect considering the diverse linguistic backgrounds of potential users. Comprehensive language support is achieved by integrating language translation services and language-specific processing modules. This ensures that the system can provide accurate and culturally relevant responses in multiple languages, taking into account linguistic nuances and cultural differences. Integration with External Data Sources facilitates access to up-to-date information about colleges, courses, and academic programs. Seamless integration with external data sources, such as college databases or APIs, ensures that the system can retrieve relevant information efficiently and reliably. Feedback Mechanisms are incorporated to gather user feedback and monitor system performance. Analyzing user interactions, errors, and feedback allows for iterative refinement of the system's functionality, accuracy, and user experience. Privacy and Security measures are implemented to safeguard user data and ensure compliance with data protection regulations. Encryption, authentication, and access control mechanisms are utilized to protect sensitive information and maintain user confidentiality. Error Handling and Recovery mechanisms are developed to gracefully manage unexpected inputs, errors, and system failures. Informative error messages, fallback options, and assistance prompts guide users and mitigate user frustration. Documentation and Support materials are provided to assist users in understanding and effectively using the system. Comprehensive documentation, tutorials, and user guides, coupled with technical support channels, ensure that users can access assistance when needed. The input design phase focuses on creating an intuitive and responsive voice input interface, implementing robust NLU techniques, ensuring comprehensive language support, handling errors gracefully, and incorporating mechanisms for user feedback and personalization. The output design phase emphasizes presenting information to users in a clear, understandable, and user-friendly manner. This involves synthesizing speech, translating responses into multiple languages, formatting responses for clarity, personalizing responses based on user context, providing visual feedback, handling errors effectively, and guiding users through the interaction flow.

By adhering to these fundamental design concepts and principles, the "AI-based Multilingual College Inquiry Voice Assistant Bot using Python" can be developed into a robust, user-friendly, and scalable system that effectively meets the needs of individuals seeking college-related information in multiple languages.

IV. IMPLEMENTATION

The implementation of the "AI-based Multilingual College Inquiry Voice Assistant Bot using Python" involves several key components and steps in the figure shown below.

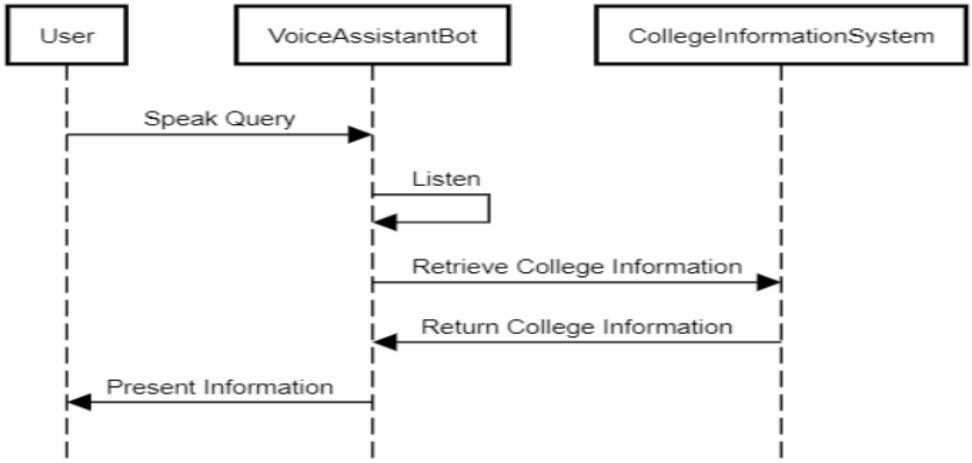


Fig. 2. Sequential diagram of the working of College Enquiry Voice Bot

When a user speaks a query to the voice bot, the Speech Recognition Module comes into play. The user's spoken words are captured by the system and converted into text using speech recognition libraries like Speech Recognition or Google Cloud Speech-to-Text. This allows for natural language interaction with the bot, enabling users to articulate their queries in their preferred language.

Once the user's query is converted into text, the Natural Language Processing (NLP) Module interprets and understands the query. It employs advanced NLP techniques such as intent recognition, entity extraction, and sentiment analysis to discern the user's intent and extract relevant information from their input. This ensures that the bot comprehends the user's query accurately and effectively.

After understanding the user's query, the system proceeds to the College Information Retrieval Module. This module retrieves detailed information about colleges from a database or API based on the user's input. The system queries the database or API to gather data such as college profiles, available courses, admission requirements, and other pertinent details related to the user's inquiry. Once the college information is retrieved, it is passed back to the voice bot system. The Response Generation Module then takes over, generating appropriate responses based on the retrieved information and the user's query. It formats the response text and translates it into the user's preferred language using the Language Translation Module. Additionally, the response is converted into speech using text-to-speech synthesis technology for seamless communication with the user.

Finally, the user is presented with the retrieved college information in detail. The voice bot system vocalizes the response, providing the user with comprehensive information about colleges, courses, admission requirements, and any other relevant details based on their initial query. This ensures that the user receives accurate and detailed information tailored to their needs, enabling them to make informed decisions about their educational pursuits. Sequence diagram consists of 5 different blocks namely user, processor, memory, Model and labels as shown in the above figure User will provide the input image through the file’s already saved image is being taken in consideration which is been captured and sent to the processor where preprocessing of data is done which is resizing, reshaping and other parameters and after that those are stored in the memory unit. After pre-processing and storing of image, trained model file is loaded where the featured of the image is extracted for classifying the output. After classifying the output, label is provided.

V. TESTING

The overarching goal of testing is to pinpoint errors or issues that arise throughout the development process, ensuring that each component aligns with user requirements and behaves as expected. Test data, encompassing both input and output scenarios, plays a pivotal role in this process, allowing for the verification of system behavior under various conditions.

At its core, software testing involves the systematic evaluation of objects, variables, functions, and modules to ensure their functionality and integration within the larger system. This iterative process spans from unit testing, where individual components are examined in isolation, to integration testing, where these components are amalgamated to form a cohesive system. Functional testing then scrutinizes the system's ability to meet technical and user-defined requirements.

Verification and *Validation* serve as fundamental pillars of the testing process. Verification entails confirming that the software adheres to specified documentation, while validation ensures alignment with user requirements outlined in the software requirement specification document. These complementary processes ensure that the software functions as intended and satisfies user expectations.

Unit testing involves testing individual software units, evaluating their functionality, and validating internal code branches and flows. Integration testing focuses on the seamless integration of validated components into a unified system, rectifying any issues that may arise during the process. Functional testing encompasses the systematic evaluation of each function to ensure compliance with technical specifications and user manuals.

Test cases are pivotal in this process, serving as blueprints for testing scenarios and aiding in the generation of test data. These cases validate program behavior under different conditions, facilitating comprehensive testing coverage. Despite the time-consuming nature of test case execution, it remains an indispensable phase, essential for identifying and addressing potential errors.

White box testing delves into the inner workings of software, scrutinizing its structure and language for potential issues. In contrast, black box testing evaluates software functionality without prior knowledge of its internal workings, relying solely on input-output analysis. Both approaches are indispensable for ensuring robust software performance.

System testing marks the culmination of the testing process, validating the integration of all components and ensuring alignment with software requirements. Test cases play a crucial role in this phase, verifying program behavior and validating user outputs.

In summary, software testing is a multifaceted process encompassing verification, validation, unit testing, integration testing, functional testing, and system testing. Test cases and test data are instrumental in evaluating program behavior under various conditions, ensuring software reliability and user satisfaction. Through meticulous testing, software developers can identify and rectify errors, ultimately delivering a product that meets user expectations and operates effectively within its intended environment.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. RESULTS

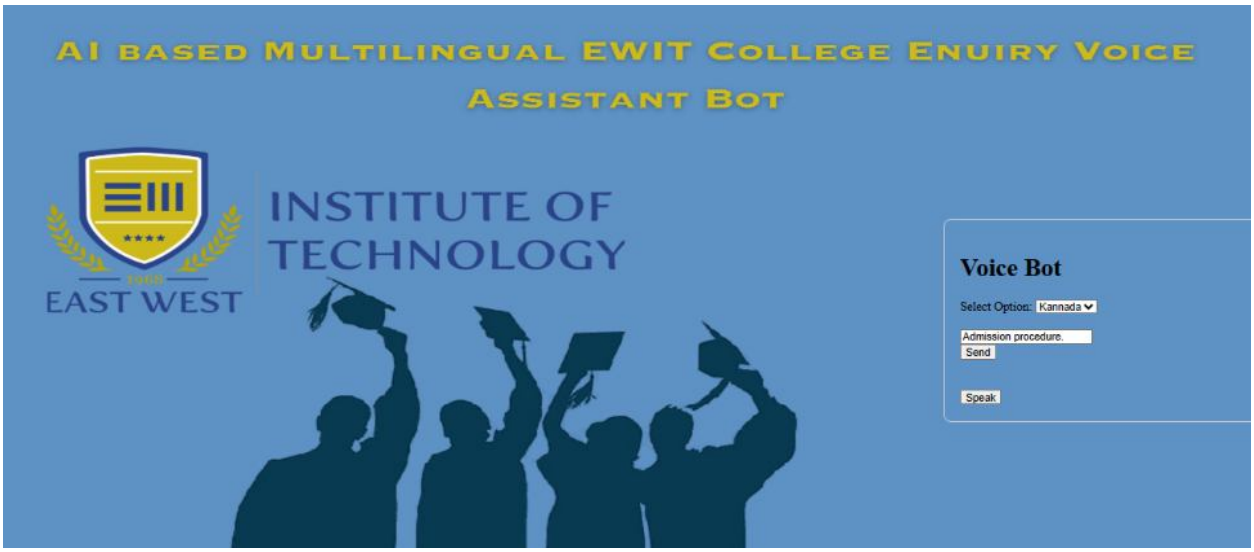


Fig.3. User Query through speech and selects the desired language (Kannada)



Fig. 4. Voice Bot response in Kannada

Fig. 3 shows the user raising a query about admission procedure and expecting an output in Kannada. The selection of the desired language is done through option selection.

The answer to the query is given through a voice by the College Enquiry Voice Bot in the selected language. The answer is also printed after the voice output, in order to make it easy for the user to go through it again. Multiple queries can be asked by the user about the Institution.

REFERENCES

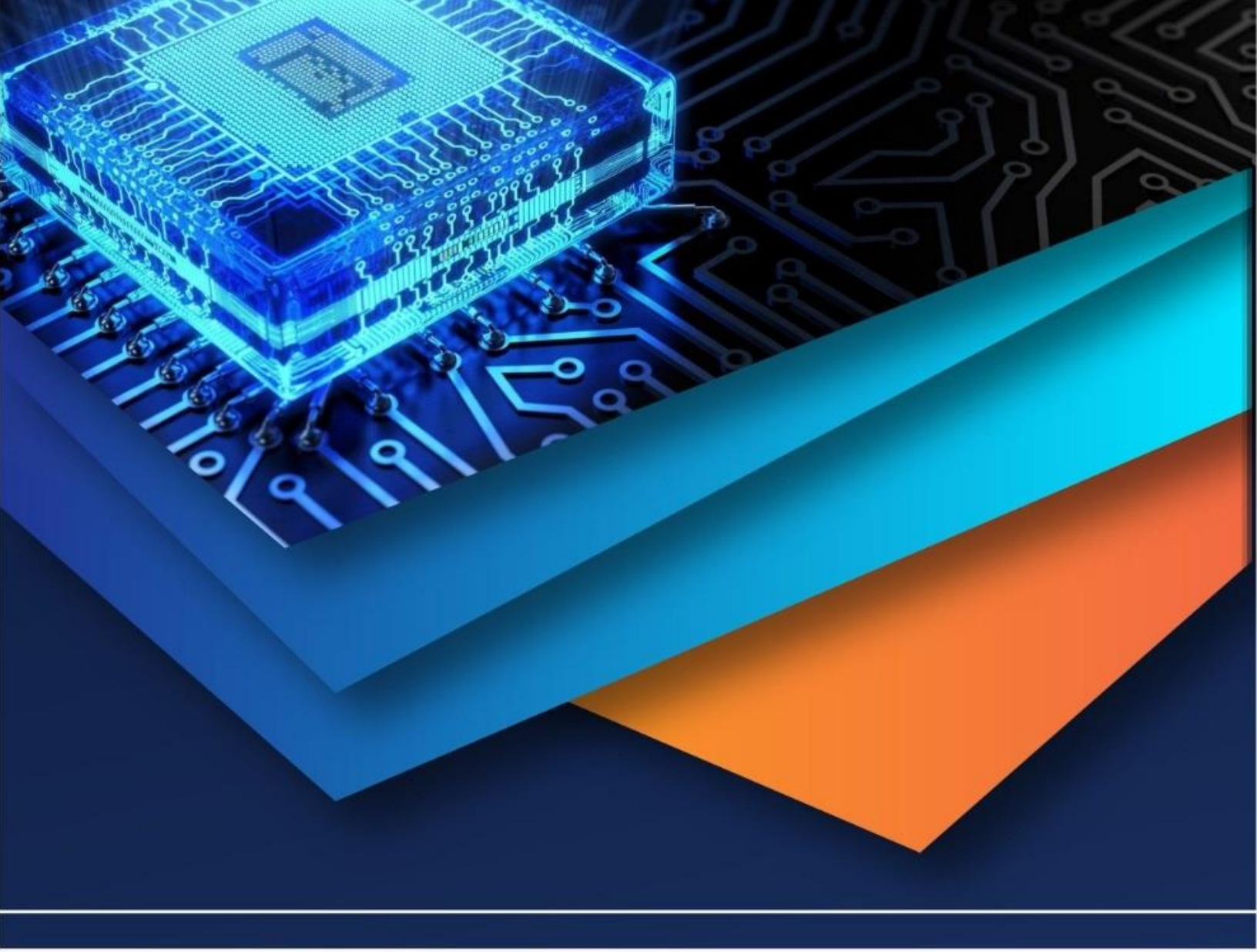
[1] Bhavika R. Ranoliya_, Nidhi Raghuwanshi_ and Sanjay Singh, “Chatbot for University Related FAQs”, IEEE Sept 2017, International Conference on Advances in Computing, Communications and Informatics (ICACCI).

[2] Godson Michael D’silva1, Sanket Thakare, Sharddha More, and Jeril Kuriakose, “Real World Smart Chatbot for Customer Care using a Software as a Service (SaaS) Architecture”, IEEE Feb 2017, International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)

[3] Naveen Kumar M, Linga Chandar P C, Venkatesh Prasad A, Sumangali K, “Android Based Educational Chatbot for Visually Impaired People”, IEEE Dec 2016, International Conference on Computational Intelligence and Computing Research (ICCIC)

[4] Bayu Setiaji; Ferry Wahyu Wibowo, “Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling” , IEEE Jan 2016, 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS).

[5] Ly Pichponreay, Chi-Hwan Choi, Jin- Hyuk Kim, Kyung-Hee Lee, Wan-Sup Cho, “Smart Answering Chatbot based on OCR and Overgenerating Transformations and Ranking”, IEEE Aug 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN) S. J. du Preez, M. Lall and S. Sinha, "An intelligent web-based voice chat bot," EUROCON 2009, EUROCON '09 IEEE.
[6] B. Denny, "Deep Learning for Chatbots," in WILDML: Artificial Intelligence, Deep Learning, and NLP vol. 2019, ed, 2016.
[7] Harsh Pawar, Pranav Prabhu, Ajay Yadav, Vincent Mendonca, Joyce Lemos, “College Enquiry Chatbot Using Knowledge in Database”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653; IC Value: 45.98, SJ Impact Factor: 6.887, Volume 6, Issue IV, April 2018, pp 2494- 2496.
[8] A. V. Justo, J. C. dos Reis, I. Calado, R. Bonacin, and F. R. Jensen, "Exploring Ontologies to Improve the Empathy of Interactive Bots," in 2018 IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 261-266.
[9] J. Jordán, S. Heras and V. Julian, “A customer support application using argumentation in multi-agent systems," in 14th International Conference onInformation Fusion pp. In Press. (2011) Bibtex, Aug 2011.
[10] G. B. Satrya P. T. Daely and S. Y. Shin," Android forensics analysis: Private chat on social messenger," in Eighth International Conference on Ubiquitous and Future Networks 2016 (ICUFN 2O16), TU Wien, Vienna, Austria, Aug 2016.
[11] Bogdan Ionescu, Cristian Gadea and Bogdan Solomon, "A chat-centric collaborative environment for web-based real-time collaboration," in IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics (SACI), Timsoara, Romania, Aug 2015.
[12] Fernando A. Mikic Fonte, Martín Llamas Nistal, Juan C. Burguillo Rial and Manuel Caeiro Rodríguez, "NLAST: A natural language assistant for students", *IEEE Global Engineering Education Conference (EDUCON)*, pp. 709-713, April 2016.
[13] Shabina Sayed, Rushabh Jain, Burhanuddin Lokhandwala, Fakhruddin Barodawala and Murtuza Rajkotwala, "Android based Chat-Bot", *International Journal of Computer Applications*, vol. 137, no. 10, pp. 29-32, March 2016.
[14] Imran Ahmed and Shikha Singh, "AIML Based Voice Enabled Artificial Intelligent Chatterbot", *International Journal of u- and e-Service Science and Technology (IJUNESST)*, vol. 08, no. 02, pp. 376-384, 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)