Global Journal of Engineering and Technology Advances

eISSN: 2582-5003

Cross Ref DOI: 10.30574/gjeta

Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Implementation of an AI-powered FAQ chatbot using the deep-learning rasa framework

Emmanuel Oyerinde, Samuel Abel, Emmanuel Mgbeahuruike, Obumneme Ukandu *, Oluwaseyi Adediran, Ifeoluwa Ikotun and Mosopefoluwa Adebawojo

Department of Computer Science Babcock University, Nigeria.

Global Journal of Engineering and Technology Advances, 2025, 23(03), 271-284

Publication history: Received on 01 May 2025; revised on 16 June 2025; accepted on 19 June 2025

Article DOI: https://doi.org/10.30574/gjeta.2025.23.3.0193

Abstract

Artificial Intelligence (AI) has become more prevalent in our day-to-day activities, as it saves human time and reduces human workload. A lot of students are clueless and know nothing about some concepts of information pertaining to their study period at Babcock University. When most go to the officials for information, it is likely the officials are in one meeting or the other, a long queue is present and many more situations emerge, which then lead to delays in the access of the required information. The aim of this project is to implement an AI-Powered FAQ chatbot using the deep-learning Rasa framework to address critical challenges faced by students seeking timely and accurate information. Based on research, the Rasa Framework was chosen for the development of the FAQ Chatbot. It is an open-source, flexible chatbot framework that allows complete control over the chatbot's behavior. Rasa Framework is known for its deep learning capabilities, specifically in Natural Language Understanding (NLU) and Dialogue Management (DM). The system is a web application with a chat screen to integrate the chatbot model. The chat screen is a user-friendly, graphical interface that the users of the system can interact with. The tools used include: PyCharm, Microsoft Visual Studio, SmartDraw, Draw.io, Flaticon, ExacliDraw.

Keywords: Artificial Intelligence; Chatbot; Rasa; Natural Language Understanding; Dialogue Management

1. Introduction

In the ever-evolving digital landscape, chatbots have emerged as revolutionary tools, fundamentally reshaping the way individuals engage with technology. Throughout the course of human existence, interaction and interdependence have been inherent, serving as essential elements for survival.

As humanity progressed in its interactions with the environment and advanced in knowledge, the necessity for creating computers became evident. The term "computer" originally referred to a job title held by individuals tasked with computing numbers all day long [1]. Computers which were initially developed for automating mathematical calculations, evolved into a combination of hardware and software [1], with the concept of this automated system credited to Charles Babbage.

Interaction with early computer systems was predominantly uni-directional, where users input data, and computers processed it to generate results. However, the evolution of computer systems has empowered them to engage in conversations, imitating human interactions rather than generating fixed outputs. These advanced systems, known as chatterbots or chatbots, have found widespread applications across various industries, including banking, retail, commerce, healthcare, and education [2]. Recent statistics reveal that a significant 58% of companies operating in the B2B sector actively employ chatbots in their operations, while 42% of B2C companies have embraced this technology

* Corresponding author: Obumneme Ukandu

[3]. This indicates a growing adoption rate, particularly in the B2B context. Additionally, 23% of companies in the customer service sector have incorporated AI chatbots into their operations [4].

Despite the automation and speed that accompanied computers and software programs, chatbots have elevated user interaction to new heights [5]. The primary motivation for chatbot users, including productivity gains, contrasts with the traditional approach of spending significant time studying website content. According to [6], users now prefer the efficiency of interacting with a chatbot for faster answers.

In educational institutions, where natural conversations play a crucial role in information gathering and navigating university life complexities, chatbots address challenges when urgent inquiries arise [7]. Multiple students pose questions to university staff members, faculty, or administrative personnel, often leading to delays and frustration. Chatbots provide a solution by offering immediate answers, mitigating wasted time and transportation expenses.

Chatbots serve diverse roles in education, from virtual tutoring to addressing inquiries related to course registration and enrollment, providing proactive student support, and more [8]. Over time, Chabot's have undergone a remarkable evolution, transitioning from rule-based and inflexible systems into versatile conversational agents. This transformation is driven by advancements in Natural Language Processing (NLP) and Machine Learning and enables chatbots to deliver personalized and sophisticated interactions [8]. Beyond their initial domain-specific functions, chatbots have now transcended into the realm of Artificial General Intelligence (AGI) and Generative Pre-trained Transformers (GPT), capable of handling a broader array of tasks and knowledge domains [9].

In the early stages of conversational AI development, a rule-based approach, mainly reliant on pattern matching, was initially employed [10]. However, its limitations in handling natural language inputs and nuanced queries prompted the need for a more intelligent system. The complexity of language expressions makes adapting and scaling challenging for rule-based models.

This spurred a shift toward a solution capable of not only recognizing patterns but also offering continuous learning for accurate and personalized responses, thereby addressing the intricacies of language and the varied expressions of intent.

Moreover, the issue of vendor lock-in, binding users to specific tools, emphasizes the need for an open-source solution. The Rasa framework, with its deep learning capabilities, provides a strategic choice, mitigating the risks of vendor lock-in.

In light of these specific challenges, the research aims to address the shortcomings of existing systems within the university, proposing the implementation of an AI-powered FAQ Chabot with deep learning capabilities.

[11] created a review enquiry chatbot on suitable chatbot framework using evidence from implementation of RASA and IBM Watson frameworks, [12], [13] developed a self-learning chatbot for university FAQs, integrating it into a web application. [14] took a similar approach but focused on integrating their chatbot with Facebook Messenger to provide information on curriculum, admissions, and schedules.

Some researchers explored more specific applications of chatbots in education. [15] Created a digital learning tool aimed at increasing students' research knowledge, while [16] designed "FLOKI," an AI chatbot specifically for maritime trainees learning collision avoidance regulations. Beyond the practical applications, [17] conducted a study examining the ethical implications of chatbots in education and research, highlighting both benefits and potential drawbacks. This research underscores the importance of responsible implementation of AI technologies in educational settings.

[18] took a different approach by assessing an existing educational chatbot named PLUTO, evaluating its effectiveness in addressing educational matters and providing immediate responses to users.

Previous studies have shown the lack of support for dynamic data which results in a static FAQ knowledge base. Also, no admin panel was implemented to manage the conversations between users and Chabot. There also exists a limitation of clarity on methodology, tools used, and development techniques.

Hence, this system provides room for handling dynamic data, an admin panel, user feedback, and learning (feedback mechanism). Our system also shifts the focus toward current and prospective students. The research provides a well-defined methodology section in Chabot development.

2. Literature review

2.1. Chatbot Overview

A chatbot is an automated tool designed to imitate and interpret human conversation, whether written or spoken [19]. Its main function is to replicate human interactions through text or voice communication. These systems can vary from basic programs that address simple inquiries to advanced digital assistants that adapt and improve over time by processing large volumes of data. Chatbots operate through Artificial Intelligence (AI), leveraging predefined rules to provide responses and solve user queries [20]. They utilize Natural Language Processing (NLP) and Machine Learning (ML) to comprehend user input and deliver appropriate answers [21]. Generally, chatbots are categorized into two types: task-based and data-driven. Task-based chatbots focus on specific, structured tasks, while data-driven chatbots, often referred to as virtual assistants, offer more flexible and personalized user experiences [22].

Today, chatbots are widely employed across multiple fields. In Information Technology (IT), they enhance service management; in business, they handle customer inquiries and guide users to the right resources [23]. They are also used in e-commerce for customer support, in education to assist with online learning, and in healthcare to provide tailored health advice, services, and even preliminary diagnostics. Additionally, they have found applications in entertainment and robotics. Notable examples of chatbot technologies include popular virtual assistants like Siri, Alexa, and Google Assistant, as well as messaging platforms such as WeChat and Facebook Messenger [19].

2.2. Strength and Weakness of Existing Systems

The development and deployment of chatbots represent a significant advancement in Artificial Intelligence (AI) and Machine Learning (ML). Current chatbot systems offer several benefits that enhance their appeal across industries. One major advantage is their ability to provide round-the-clock customer service and support, offering users immediate assistance at any time [24]. Over time, maintaining chatbots proves more cost-effective than hiring and training a full customer support team. Additionally, with continuous improvements in AI and Natural Language Processing (NLP) technologies, chatbots are becoming increasingly adept at interpreting user queries and delivering accurate responses tailored to customer needs [25].

Despite these strengths, existing chatbot systems also face notable limitations. Many struggle with handling queries outside their training data, often leading to inadequate or incorrect responses [26]. Language processing challenges, including difficulties with accents and grammatical errors, further hinder their performance. Chatbots typically require extensive dialogue data to be properly trained and often find it difficult to manage non-linear, back-and-forth conversations. Their limited understanding of user inputs, lack of emotional intelligence, and sometimes impersonal interactions diminish the user experience. Moreover, the setup and maintenance of advanced chatbot systems can be costly and complex, while concerns around data privacy and security continue to pose significant risks for both users and providers.

2.3. Trends in Chatbot

As a result of improvements in AI, machine learning, and natural language processing, the trends represent the changing landscape of chatbots. Technology advancement causes the emergence of new trends which include:

- Multimodal Chatbots: Chatbots that can handle multiple modes of communication and various sources to extract information, such as text, voice, images, and gestures [27]. Integrating speech recognition, computer vision, and text-based chatbots for a seamless multimodal experience.
- Emotional Intelligence: Creating chatbots that can detect and respond to user emotions, enabling a greater empathy and context-aware interactions.
- Enhanced Personalization: Personalizing chatbot interactions to deliver highly tailored experiences to individual users.
- Continuous Learning: Chatbots that can learn and adapt from user interactions over time to improve themselves.
- Hybrid Models: Combining rule-based and AI-driven approaches to create more versatile and robust chatbots.
- Explainable AI: Developing chatbots that can explain their decisions and reasoning to users, enhancing transparency and trust.
- Chatbots in Healthcare: Expanding the use of chatbots for healthcare applications, such as remote patient monitoring, mental health support, medical ethics, and medication reminders [20].
- Chatbots for Education: Utilizing chatbots for personalized learning, tutoring, and educational support.

- Contextual Awareness: Enhancing chatbots' ability to understand and remember user context for more natural and fluid conversations.
- Security and Fraud Detection: Implementing robust security measures in chatbots to prevent security threats like fraud, malicious activities and provide solutions if need be.

Integration of Augmented Reality (AR) and Virtual Reality (VR): To allow more engaging and interactive experiences through the combination of AR/VR and conversational AI.

3. Methodology

In the development of any system, especially a complex one, the process of decomposition is crucial. Decomposition involves breaking down a problem or system into parts that are easier to conceive, understand, program, and maintain. Hence, the importance of system analysis and design.

3.1. Data Collection

The essence of collecting data was to help understand the requirements and pain-points of students who will use this product upon completion of the project. A google form was used for this purpose. This was to ensure that the form could be easily accessed and filled by students across the university.

3.2. Data Organization

The collected data was organized based on their similarity and relevance. This helped to group similar requirements and prevent repetition or redundancy.

3.3. Intent Classification, Training, and Development

The organized data was converted to various intents which aided the system (Chabot) to understand what the user needs and respond by providing accurate and reliable information. To achieve effective intent classification, the Chabot model required diverse training examples. This training phase was crucial for the model's ability to generalize. The Chabot model was integrated into a Python Flask application, which was deployed on a cloud Platform as a Service (PaaS) provider like Render or Heroku.

3.4. Implementation of the Chatbot

The implementation of the Chabot is broken down into the development of the Chabot model and the web application. Following the design of the Chabot, this cannot be completed without implementing the model. Several tools were used for this purpose and they include:

- ***Python:*** This is the primary language for implementing Rasa and developing custom actions for the Chabot. This language is also popular and well-known for its brevity and simplicity. No wonder there are so many python libraries for machine learning and NLP tasks.
- ***Rasa Framework:*** This is an open-source conversational AI framework used for building the Chabot's natural language understanding and dialogue management.

3.4.1. How Rasa Works

Rasa consists of four main parts namely: Natural Language Understanding (NLU), Rasa Core, Channel and Action, and additional Helper functions like Tracker Store, Lock Store, and Event Broker. When a user interacts with the Chabot, the NLU component takes the lead in understanding the user's intent by classifying it accurately and extracting essential details (entities). This ensures a precise grasp of what the user is saying. Following that, Rasa Core manages the conversation, picking the best response and action based on the chat's history. Importantly, Rasa Core smoothly communicates with the action server, allowing the Chabot to perform custom actions in response to user queries and vice versa. This teamwork of components lets Rasa provide a smart and context-aware conversational AI experience. Figure 1 illustrates the architecture of a Rasa chatbot. The Client sends a message to Rasa NLU for intent classification and entity extraction, and passes it to Rasa Core for dialogue management, which may trigger the Action Server to perform custom actions, with responses sent back to the Client.

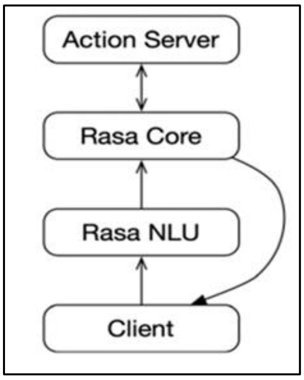

Figure 1 Core Working Process of Rasa [28]

3.5. Implementation of Web Application

The development of the web application was broken down into the user module and the administrator module.

- ***User Module:*** The user module serves as the Chabot application, providing a platform for users to interact seamlessly. Users can pose questions, and the system promptly generates relevant responses.
- ***Administrator Module:*** The administrator module is categorized into two key roles:
  - Faculty-Level Administrator: Assigned to each faculty in Babcock University, these administrators are responsible for managing intents, entities, and utterances specific to their respective faculties.
  - Super-Level Administrator: This role holds the overarching managerial position within the system. Super-Level Administrators have the authority to oversee and manage all faculty administrators across the entire system.
- ***Web Development Technologies Used:*** The following technologies were used for the development of the web application:
  - Flask: Flask is a well-known and popular Python-based web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.
  - MongoDB: MongoDB is a popular, open-source, document-oriented NoSQL database program. It is designed to be flexible, scalable, and high-performance, making it suitable for a wide range of applications. MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.
  - HTML: HyperText Mark-up Language is the standard markup language for documents designed to be displayed in a web browser. It lays out the structure and gives the content for the web page.
  - CSS: Cascading Style Sheet is a language used for specifying the styles in which a document written in a mark-up language like HTML should be displayed.
  - JS: Javascript is the core technology of the web. It is a scripting language used by developers to create dynamically updated content and make a web page interactive.
  - Software Development Process Model: The adopted software development process model for this system is the incremental model, also known as a multiple-waterfall model. It is apt for this project as it facilitates the breakdown of the entire project into manageable modules, with each module integrating incrementally. Figure 2 displays the Incremental Model of software development, where the system is built and delivered in a series of increments, each involving cycles of analysis, design, implementation, testing, deployment, and maintenance. New functionality is added in each release (Release 1, Release 2, Release 3, etc.), allowing for partial system delivery and earlier user feedback.

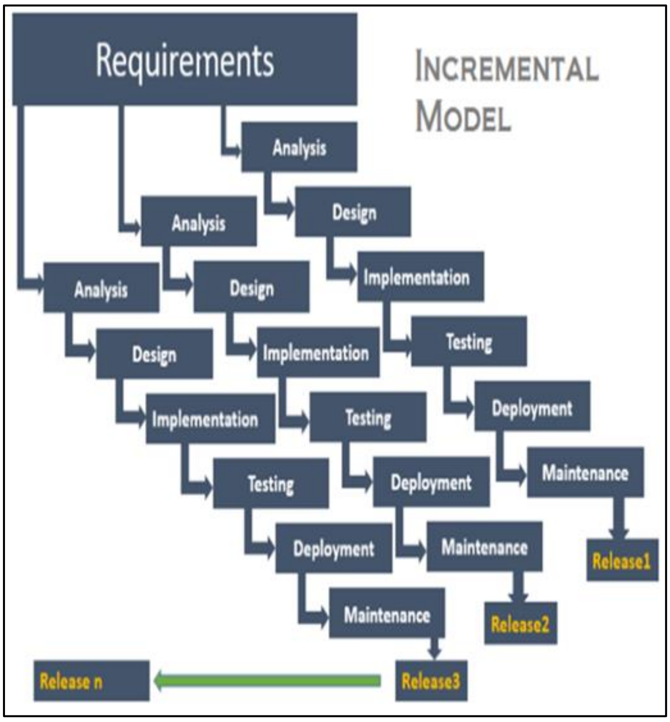

Figure 2 The Incremental Process Model

- System Analysis: The initial phase focused on understanding the system and gathering essential requirements. This involved a thorough examination of the components necessary for the system's functionality.
- System Design: Following the analysis, the design phase set out to conceptualize the proposed system. Emphasis was placed on determining how the various components interact, including the logical flow of the system. Of paramount importance was the design of the user interface to ensure seamless user interaction.
- Implementation: This phase involved translating the established requirements and design into executable programs. The collective code formed the fundamental source code for the software.Testing: This phase involved validating the software against the initially identified requirements. This was to ascertain that the product aligns precisely with the needs identified during the requirement gathering phase.
- Deployment: Upon successful testing, the deployment phase initiates the process of delivering the software to customers.
- Maintenance: This phase involves constantly seeking out ways to improve or optimize the system. Also, as the needs and requirements of customers change from time to time it is of utmost importance to get their feedback on the performance of the system and making modifications where and when needed.

3.6. Requirement Analysis

Requirement analysis, also known as requirements engineering, is the process of determining and documenting the needs and expectations of a new or modified product. These needs and expectations can be referred to as requirements and they can be categorized into two main types: Functional Requirements and Non-Functional Requirements.

3.6.1. Functional Requirements

Functional requirements define what a system is supposed to do. They are like a to-do list for a system, such as a chatbot. They describe what the system should do, its functions, features, and capabilities. These requirements lay out specific tasks or services the system must perform, acting as a guide for developers to ensure the Chabot behaves as intended.

The functional requirements of the system are based on the user roles which are:

- User
- Administrator

Functional Requirements for User Role:

- Users shall be able to register with their username, email, password, faculty, and department.
- Users shall be able to login using username or email along with a password
- Users shall be able to upload a profile picture
- Users shall be able to view and edit their profiles
- Users shall be able to create multiple chat tabs
- Users shall be able to pose queries to the chatbot using texts
- Users shall be able to provide their sentiments/feedback based on the chatbot's response
- Users shall be able to send questions the chatbot could not answer to the relevant departmental admin.
- Users shall be able to view chat history

Functional Requirements for Admin Role:

- Admin shall be able to log in to the system with username or email and password.
- Admin shall be able to upload a profile picture.
- Admin shall be able to view and edit profiles.
- Admin shall be able to create new intents and modify existing ones.
- Admin shall be able to modify, or review training data for each intent.
- Admin shall be able to view real-time information about the performance of the Chabot.
- Admin shall be able to receive questions the Chabot could not answer.

3.6.2. Non-functional requirements

Non-functional requirements, on the other hand, define how the system should perform. They specify the quality attributes and constraints that the system must satisfy. These requirements are not directly related to the specific behavior or functions of the system but rather focus on the overall performance, usability, security, reliability, and other quality attributes of the system. Non-functional requirements are often used to define the performance, usability, and other quality attributes of the system and are essential for ensuring that the system meets the desired quality standards. Some of the non-functional requirements are listed below.

- Security: One of the key metrics that will be used to evaluate this chatbot is the accuracy of the information it provides. Hence, data integrity is quite imperative to prevent unauthorized access to an administrator account which could lead to compromise. Implementing secure login mechanisms for admin access would be expedient.
- Performance: the system should ensure quick response times and minimal downtime for seamless user interaction.
- Accuracy and Reliability: the system should provide accurate and reliable responses through regular knowledge base updates.
- User-Friendly Interface: the system should have an easy-to-navigate interface for users and admins.
- Maintainability: the system should be maintainable in terms of writing clean, simple, and readable code.
- Availability: the system should be available at all times for uninterrupted service.

3.6.3. Hardware Requirements

The minimum hardware requirement for the effective development and implementation of this system are as follows:

- RAM:
  - Minimum: 4GB.
  - Recommended: 8GB or more for a smoother experience.
- Processor:
  - Minimum: Dual-core processor.
  - Recommended: Quad-core processor or better.
- Hard Disk:
  - Minimum: 250GB with at least 50GB of free space.
  - Recommended: 500GB or more.

3.6.4. Software Requirements

- Operating System: Cross Platform.
- Web Browsers: Chrome, Mozilla Firefox, Safari

3.7. Flowchart

A flowchart, otherwise known as a process flow diagram, is the diagrammatic representation of the logical flow of the program. It involves the use of shapes, lines, arrows, and other symbols. It is a graphical representation of steps in sequential order and is widely used in presenting the flow of algorithms, workflow, or processes. Flowcharts are used to design and document simple processes or programs and help visualize the process. They are used in various fields to analyze, design, document, or manage a process or program. Flowcharts typically use symbols to represent different aspects of processes and are complemented by other types of diagrams. The symbols used in flowcharts include process steps, decisions, inputs and outputs, documents, and delays or wait times. Flowcharts are a common process analysis tool and are used to develop an understanding of how a process is done, study a process for improvement, and communicate to others how a process is done. Figure 3 shows a user login system where, after successful credential validation, users can either manage intents and entities (create, update, search, delete, and retrain models), update their profile (change username or password), or view the dashboard. After completing an action, users are given the option to perform another action or exit the system. Figure 4 shows the flowchart for the Super-Level Administrator role where, after successful login, users can manage intents and entities, review unanswered queries, manage other admin accounts, or update their profile, with the option to perform multiple actions before exiting. Figure 5 shows how users interact with a web application to ask questions, which are sent via a Flask server to the Rasa NLP engine for responses, while admins manage intents and monitor system performance through analytics, and all conversations are stored in a database.

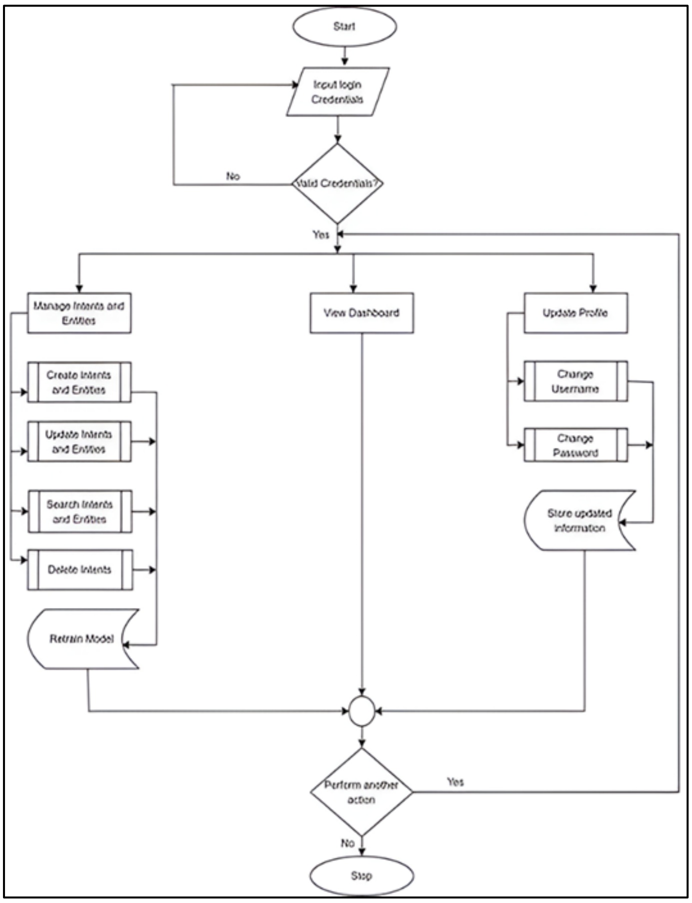

Figure 3 Flowchart Diagram for the Faculty-Level Admin Role

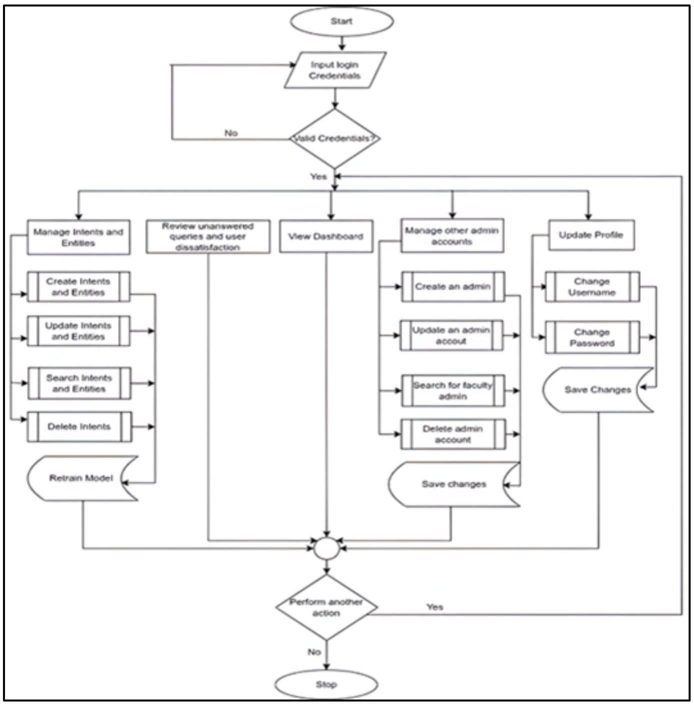

Figure 4 Flowchart Diagram for the Super Level Admin

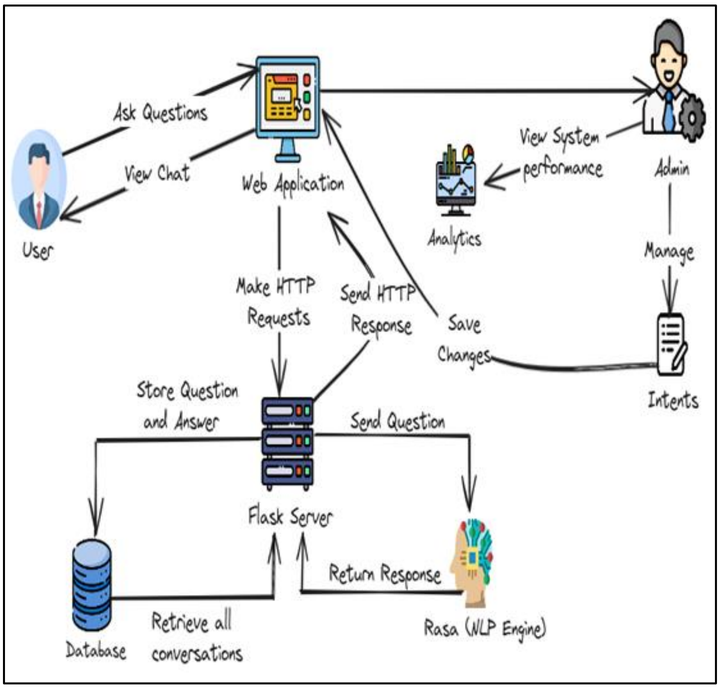

Figure 5 Logical Flow of the System

4. Implementation

The application was built on web technology and it consists of a chat screen, the main feature that integrates with the Chabot model. The chat screen acts as an abstraction layer to provide a user-friendly, graphical interface that the users of the system can interact with. At its core, the model harnesses the power of the Rasa framework, renowned for its deep learning capabilities, specifically in NLU and DM. This sophisticated model is the brain behind the Chabot which makes it adept at processing and interpreting user queries transmitted through the chat screen to the Rasa server to generate accurate and helpful responses that are delivered back to the chat screen where they are displayed to the user.

The web application features a cohesive interface that includes landing page, login, sign-up, and chat screen functionalities, all of which integrate smoothly with an active Rasa Chabot model instance. Whenever a user types in a query and submits it, the system dispatches a POST request to the Rasa server and the response from the server is relayed back to the web application, ensuring a smooth and interactive user experience.

4.1. Landing Page

This page gives a general description of the application and serves as the welcoming interface.

Figure 6 This figure presents the Babcock University Chatbot landing page as a personal campus guide and knowledge base designed to revolutionize school support by providing instant answers to student queries through an interactive chat interface.

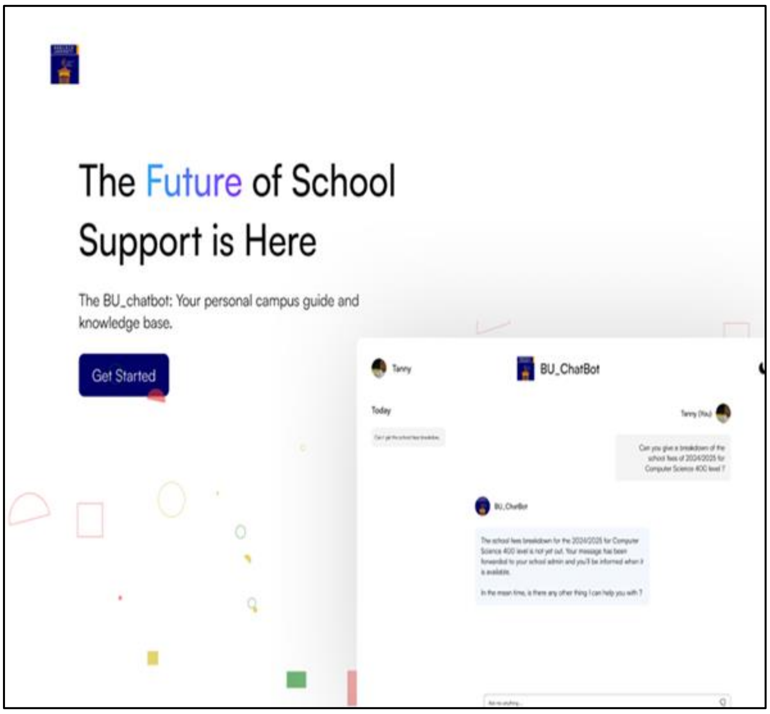

Figure 6 Landing Page

4.2. Login Page

This page allows returning users access to their accounts by entering their credentials. Figure 7 shows the login page for the chatbot, where users can sign in with their email or username and password, create a new account, or recover a forgotten password.

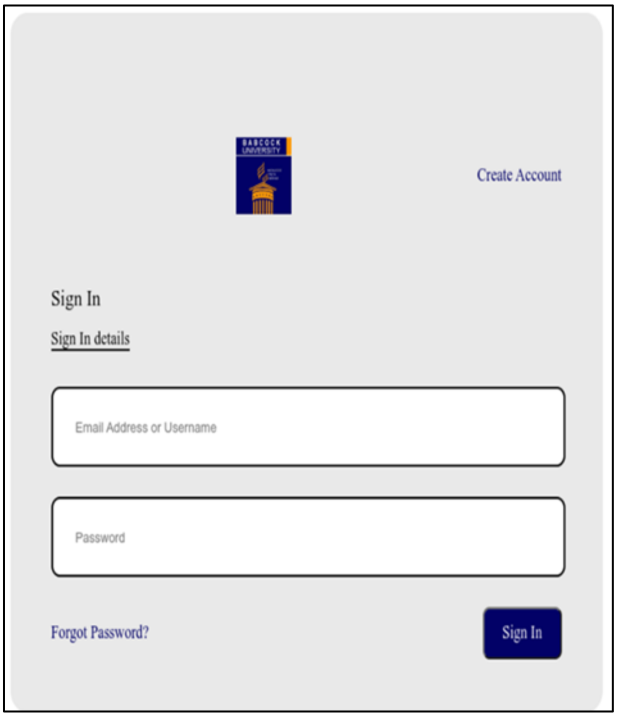

Figure 7 Login Page

4.3. Chat Screen

The chat screen is the core interface for interacting with the AI-powered Chabot. It presents a conversation window where users can type their queries and view the Chabot's responses. Fig 8 shows the chat interface where a user interacts with the Chatbot, asking questions about academic matters and receiving automated, informative responses.

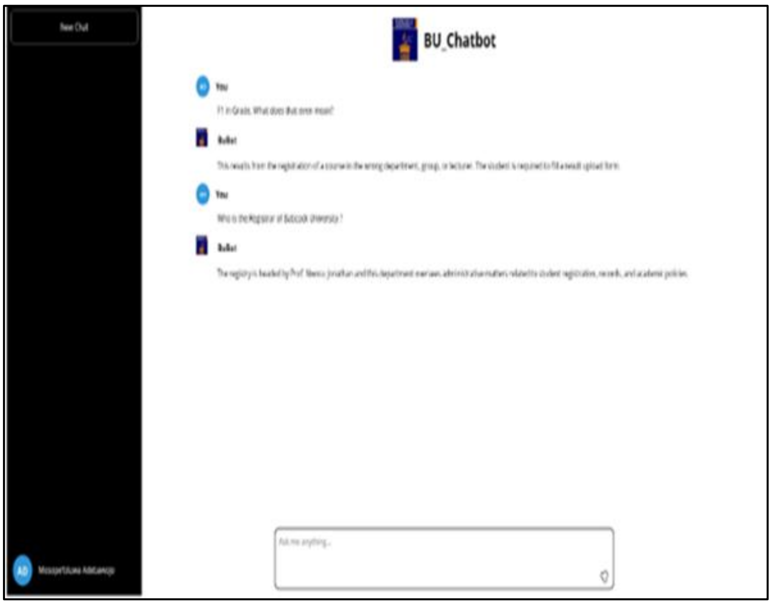

Figure 8 Chat Screen

4.4. Admin Dashboard

Admin can manage the system, the users, and the information provided to users through this page. Figure 9 displays an admin dashboard interface for the chatbot, showing the total number of registered users and navigation options for managing intents, entities, the knowledge base, and training.

Figure 9 Admin Dashboard Overview for User and Content Management

4.5. System Testing

This gives a detail of the testing phase of the system. The different parts of the system were linked to enable navigation from one end to another. Standard and consistent coding practices were followed. Testing was also made to ensure the system worked according to specifications in the form of design, action, efficiency, and safety. The system was also tested for errors, which were corrected to increase the efficiency and performance of the system. The system was deployed on different computers and browsers to ensure its effectiveness.

4.6. Unit Testing

The individual parts of the system were tested for errors and functionality. For example, the separate parts of the server and the different web pages, were each tested to ensure they worked as they are required to. This was done for easy identification of errors and to avoid complex or unknown sources of error in the integrated system.

4.7. Integration Testing

This testing is done on the system as a whole. All parts have been combined and connected to make the system itself. Various tests were conducted to ensure the system is complete and functional. This test was also done to confirm that the system does what it was developed for accurately and efficiently.

5. Conclusion

The study was able to design an AI-powered FAQ chatbot model, implement the model, and also test the model. The chatbot's development ensured that important information was provided to Babcock University students. Information such as the main school officials, application process, registration process, and much more. Chatbots can make a significant difference in helping people achieve their tasks if built and used for their intended purpose. NLP and machine learning play an important role in chatbot development, enabling an improved understanding of user input and the generation of contextually relevant responses. However, they face challenges such as NLU, context management, ethical considerations, and user privacy concerns. Future research may consider ensuring that as the technology evolves, more features are being added to aid its effectiveness and efficiency.

Compliance with ethical standards

Acknowledgments

Appreciation goes to my colleagues and contributing authors for their support and contributions to this research work. Appreciation also goes to the technicians who assisted in the implementation.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] C. Thompson, "The Gendered History of Human Computers," Smithsonian Magazine, Jun. 2019. [Online]. Available: https://www.smithsonianmag.com/science-nature/history-human-computers-180972202/

[2] B. Shawar and E. Attwell, "Chatbots: Are they really useful?," LDV Forum, vol. 22, pp. 29-49, 2007.

[3] D. Ritter, "Chatbot statistics crucial to know in 2023," Dashly. Accessed: Oct. 23, 2023. [Online]. Available: https://www.dashly.io/blog/chatbot-statistics/

[4] N. Bleu, "29 Top Chatbot Statistics For 2023: Usage, Demographics, Trends," Blogging Wizard. Accessed: Oct. 23, 2023. [Online]. Available: https://bloggingwizard.com/chatbot-statistics/

[5] A. Ahmadi, "ChatGPT: Exploring the Threats and Opportunities of Artificial Intelligence in the Age of Chatbots," Asian Journal of Computer Science and Technology (AJCST), vol. 12, no. 1, pp. 25–30, May 2023.

[6] R. A. Akinyemi, W. Ajayi, and A. Atuman, "Automation of Customer Support System (Chatbot) to Solve Web Based Financial and Payment Application Service," AJCST, vol. 12, no. 2, pp. 1–17, Aug. 2023.

[7] E. Adamopoulou and L. Moussiades, "An overview of chatbot technology," IFIP Advances in Information and Communication Technology, pp. 373-383, 2020. DOI: 10.1007/978-3-030-49186-4_31.

[8] Z. H. Sain, R. Serban, M. A. Agoi, and S. H. Sain, "Leveraging ChatGPT to Enhance Debugging: Evaluating AI-Driven Solutions in Software Development," AJCST, vol. 13, no. 1, pp. 41–44, Apr. 2024.

[9] R. Sharma and V. Patel, "Enhancing Software Development with AI-Integrated Debugging Tools," International Journal of Computer Science, vol. 22, no. 1, pp. 123–137, 2023.

[10] G. Bansal, V. Chamola, A. Hussain, M. Guizani, and D. Niyato, "Transforming conversations with AI—a comprehensive study of ChatGPT," Cognitive Computation, pp. 1–24, 2024.

[11] N. Kalyani and T. Sultana, "A Review on the Determinants of a Suitable Chatbot Framework—Empirical Evidence from Implementation of RASA and IBM Watson Assistant Frameworks," 2022.

[12] A. Caleb and M. Nnaemeka, "Automated Self-Learning Chatbot for University Related FAQs," Bachelor's thesis, Babcock University, 2022.

[13] M. Senthil Kumar, J. Dharshani, K. Divyabharathi, and S. Sneha, "An Automated Chatbot for an Educational Institution using Natural Language Processing," vol. 10, no. 5, 2022.

[14] J. Grudin and R. Richard, "Chatbots, Humbots, and the Quest for Artificial General Intelligence," in Proc. 2019 CHI Conf. on Human Factors in Computing Systems (CHI '19), ACM, 2019, pp. 209–219. DOI: 10.1145/3290605.3300439.

[15] Patchara, "Chatbot Development As a Digital Learning Tool to Increase Students' Research Knowledge," vol. 14, no. 2, 2021.

[16] V. Karad, "Research Paper on Chatbot Development for Educational Institute," vol. 411038, 2021.

[17] K. Chokri, "Chatbots in Education and Research: A Critical Examination of Ethical Implications and Solutions," vol. 15, no. 5614, 2023.

[18] V. Lonare, S. Jiwane, S. Janbandhu, Y. Khandale, and T. H. Khan, "Research Paper on PLUTO: A Smart Chatbot," vol. 10, no. 7, 2022.

[19] E. Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology," IFIP Advances in Information and Communication Technology, vol. 584, no. 1, pp. 373–383, May 2020, doi: https://doi.org/10.1007/978-3-030-49186-4_31.

[20] U. Nzenwata, O. Ukandu, and O. Bakare, "Artificial Intelligence: Positive or Negative Innovation," International Journal of Emerging Trends in Engineering Research, vol. 11, no. 7, pp. 245–252, Jul. 2023, doi: https://doi.org/10.30534/ijeter/2023/031172023.

[21] E. Kasthuri and S. Balaji, "Natural language processing and deep learning chatbot using long short term memory algorithm," Materials Today: Proceedings, vol. 81, no. 2, May 2021, doi: https://doi.org/10.1016/j.matpr.2021.04.154.

[22] H. Abdulla, A. M. Eltahir, S. Alwahaishi, K. Saghair, J. Platos, and V. Snasel, "Chatbots Development Using Natural Language Processing: A Review," IEEE Xplore, Jul. 01, 2022. https://ieeexplore.ieee.org/abstract/document/10017592

[23] R. Shah, S. Lahoti, and K. Lavanya, "An intelligent chat-bot using natural language processing," International Journal of Engineering Research, vol. 6, no. 5, p. 281, 2017, doi: https://doi.org/10.5958/2319-6890.2017.00019.8.

[24] L. K. Fryer, K. Nakao, and A. Thompson, "Chatbot learning partners: Connecting learning experiences, interest and competence," Computers in Human Behavior, vol. 93, no. 15, pp. 279–289, Apr. 2019, doi: https://doi.org/10.1016/j.chb.2018.12.023.

[25] W. Floyd et al., "Current Strengths and Weaknesses of ChatGPT as a Resource for Radiation Oncology Patients and Providers," International Journal of Radiation Oncology*Biology*Physics, vol. 118, no. 4, Oct. 2023, doi: https://doi.org/10.1016/j.ijrobp.2023.10.020.

[26] A. Følstad and C. Taylor, "Investigating the user experience of customer service chatbot interaction: a framework for qualitative analysis of chatbot dialogues," Quality and User Experience, vol. 6, no. 1, Aug. 2021, doi: https://doi.org/10.1007/s41233-021-00046-5.

[27] D. Chen et al., "Performance of Multimodal Artificial Intelligence Chatbots Evaluated on Clinical Oncology Cases," JAMA Network Open, vol. 7, no. 10, p. e2437711, Oct. 2024, doi: https://doi.org/10.1001/jamanetworkopen.2024.37711.

[28] T. Smith and W. Rajalingam, Conversational AI with Rasa. Birmingham, U.K.: Packt Publishing, 2021. [Online]. Available: https://www.packtpub.com/en-us/product/conversational-ai-with-rasa-9781801077057