

Table of Contents

Chapter No	Title of the Chapter	Page No
1	Introduction	5
2	Description of Tools	8
3	Design & Development	12
4	Demonstration/Evaluation and Results	18
5	Conclusion & Future Work	21
	References	22

Chapter – 1 Introduction

The introduction highlights the importance of chatbots in modern web development, particularly in the context of university websites. Chatbots serve as valuable tools for providing real-time information to prospective students, current students, and other stakeholders. By leveraging advanced technologies like JavaScript, HTML, CSS, and AI language models, chatbots can deliver accurate and user-friendly responses to a wide range of queries.

1.1 Overall Background and System

An AI integrated chatbot is a computer program designed to simulate conversation with human users, primarily over the internet. These chatbots leverage artificial intelligence (AI) technologies such as natural language processing (NLP), machine learning (ML), and sometimes deep learning to understand, interpret, and respond to user queries in a human-like manner. AI chatbots can handle complex interactions and improve over time through learning from interactions.

The integration of AI into chatbots is driven by the need for more efficient, responsive, and scalable communication systems. Traditional rule-based chatbots, which rely on predefined scripts, are limited in their ability to handle nuanced human conversations. AI chatbots, on the other hand, can understand context, manage ambiguity, and provide more accurate and personalized responses. This advancement reduces the need for human intervention and enhances user experience.

AI integrated chatbots are utilized across various industries and sectors:

- **Customer Service:** Businesses use AI chatbots to provide 24/7 support, answer frequently asked questions, and resolve common issues.
- **Healthcare:** Chatbots assist patients by providing information on symptoms, scheduling appointments, and reminding them of medications.
- **Education:** Educational institutions deploy chatbots to answer student queries, assist with admissions, and provide learning support.
- **E-commerce:** Online retailers use chatbots for product recommendations, order tracking, and handling returns and exchanges.
- **Banking and Finance:** Banks implement chatbots for handling customer inquiries, processing transactions, and providing financial advice.

Applications:

- **Customer Support:** Automating responses to common customer queries.
- **Virtual Assistants:** Helping users with tasks such as setting reminders, booking appointments, or providing information.
- **E-commerce Assistance:** Assisting customers in navigating websites, finding products, and making purchases.

- **Healthcare Guidance:** Offering preliminary medical advice and booking appointments.
- **Educational Tools:** Providing tutoring or answering questions about courses and enrolment processes.

Importance and Necessity: The importance of AI integrated chatbots lies in their ability to improve efficiency and accessibility in communication. They provide instant responses, operate 24/7, and can handle multiple interactions simultaneously. This reduces the workload on human agents, cuts operational costs, and enhances customer satisfaction. In sectors like healthcare and education, chatbots can significantly improve access to information and services, thereby contributing to better outcomes. Moreover, as AI technology advances, chatbots become more adept at understanding and responding to human language, making them indispensable tools in various applications.

1.2 Objective

The primary objective of this project is to develop and implement an AI integrated chatbot that enhances user engagement on the university's website. This chatbot aims to provide prospective students with comprehensive, accurate, and timely information about various aspects of university life. By leveraging artificial intelligence, the chatbot will be able to interact with users in a human-like manner, ensuring that their queries are addressed promptly and efficiently.

1.3 Methodology

Software Tools:

- Java Script/Python
- HTML
- CSS
- Gemini Bard API
- VS Code

Hardware Tools:

No hardware tools

1.4 Outcome(s)

The expected outcomes of this AI integrated chatbot project for the university website include:

1. Functional AI Chatbot:

A fully operational chatbot capable of interacting with users through text (and voice if applicable), providing accurate and relevant responses to queries about academic programs, campus life, and contact details.

2. Increased User Engagement:

Enhanced user engagement on the university website, with longer visitor sessions and higher interaction rates due to the chatbot's ability to provide instant and personalized responses.

3. Improved Information Access:

Users will have easy access to comprehensive information about the university, including academic offerings, campus facilities, extracurricular activities, and administrative contact information.

4. 24/7 Availability:

The chatbot will provide round-the-clock support, ensuring that prospective students from various time zones can obtain information and assistance at any time.

5. Personalized User Experience:

The chatbot will use machine learning to tailor responses based on user preferences and past interactions, delivering a personalized and engaging user experience.

6. Increased Conversion Rates:

By guiding prospective students through the application process and providing detailed, accessible information, the chatbot will help increase the number of completed applications and admissions.

7. Reduced Human Workload:

The chatbot will handle routine inquiries, freeing up university staff to focus on more complex and high-value tasks.

8. Scalable and Adaptable System:

The chatbot system will be scalable to accommodate a growing number of users and adaptable to incorporate new information and respond to emerging trends in user queries.

Chapter – 2 Description of Tools

This section delves into the various tools and technologies used in the development of the Amrita University chatbot. It emphasizes the crucial roles played by JavaScript, HTML, CSS, and the Bard API in creating an interactive, visually appealing, and responsive chatbot interface. The integration of these technologies ensures a seamless user experience and enables the chatbot to communicate with backend servers efficiently.

2.1 Python

Python is a high-level, interpreted programming language renowned for its simplicity, readability, and versatility. It has become a staple in the fields of AI and machine learning due to its extensive collection of libraries and frameworks, such as TensorFlow, Keras, PyTorch, Scikit-learn, and NLTK. These libraries facilitate the development and deployment of sophisticated AI models without needing to implement complex algorithms from scratch. Python's design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code compared to other languages like Java or C++. This makes Python an ideal choice for developers who want to focus on solving problems rather than struggling with language intricacies.

In this project, Python serves as the primary language for developing the AI chatbot. Its robust ecosystem allows for efficient implementation of natural language processing (NLP) and machine learning algorithms, which are critical for building an intelligent chatbot that can understand and respond to user queries. Python's extensive support for data manipulation and analysis, through libraries like Pandas and NumPy, facilitates the preprocessing and handling of large datasets, which is essential for training machine learning models. Moreover, Python's simplicity and readability help maintain a clear and manageable codebase, which is crucial for collaborative development and future maintenance.

Python also benefits from a rich community and comprehensive documentation, which provide invaluable resources for troubleshooting and learning. The Python community continuously contributes to a vast array of open-source libraries and tools that can be leveraged to enhance the capabilities of the AI chatbot. For instance, the use of NLP libraries such as NLTK and SpaCy allows the chatbot to perform tasks like tokenization, sentiment analysis, and entity recognition. These capabilities enable the chatbot to comprehend user input more effectively and provide relevant and accurate responses.

Additionally, Python's integration capabilities with other technologies make it a versatile tool for this project. It can easily interface with web development technologies through frameworks like Flask and Django, enabling seamless integration of the AI chatbot into the university's web infrastructure. This interoperability ensures that the chatbot can be embedded within the university's existing systems, providing a cohesive and user-friendly experience for prospective students and other users.

2.2 JavaScript

JavaScript is a versatile, high-level programming language that is essential for web development. It is primarily used for creating interactive and dynamic web pages, making it indispensable for front-end development. JavaScript enables the creation of responsive user interfaces that can handle user interactions seamlessly, providing a more engaging user experience. Its ability to manipulate the Document Object Model (DOM) allows developers to update the content and structure of web pages in real-time, responding to user actions without needing to reload the entire page.

In this project, JavaScript is used to enhance the chatbot's user interface. It enables dynamic content updates, ensuring that the chatbot can provide real-time responses and feedback to user inputs. By leveraging JavaScript, the development team can implement features such as real-time message updates, user input handling, and interactive elements like buttons and forms. This ensures that the chatbot interface is not only functional but also engaging and user-friendly. JavaScript's asynchronous capabilities, through technologies like AJAX and Fetch API, allow the chatbot to communicate with the backend server without disrupting the user experience.

JavaScript works in conjunction with HTML and CSS to create a responsive and user-friendly chatbot interface embedded in the university website. This trio forms the core of web development, with HTML providing the structure, CSS handling the presentation, and JavaScript enabling interactivity. In the context of the AI chatbot, JavaScript facilitates the seamless integration of various interactive features, ensuring that users have a smooth and intuitive experience when interacting with the chatbot. The ability to handle events, such as user clicks and form submissions, allows the chatbot to respond promptly and appropriately to user queries.

Furthermore, JavaScript's vast ecosystem, including libraries and frameworks like React, Angular, and Vue.js, provides additional tools to enhance the development process. These frameworks offer advanced functionalities and design patterns that help in building complex user interfaces more efficiently. For example, using React can help in managing the state of the chatbot interface, ensuring that it remains consistent and responsive as users interact with it. This contributes to creating a polished and professional user experience, which is crucial for engaging prospective students and providing them with accurate and timely information.

2.3 HTML

HTML (Hyper Text Markup Language) is the standard language for creating web pages and web applications. It defines the structure of web content, including elements such as headings, paragraphs, links, images, and forms. HTML provides the foundational structure that web browsers use to display content. Each element in HTML represents a different part of the webpage, allowing developers to organize content logically and semantically. This structured approach is essential for ensuring that web pages are accessible and easy to navigate for users.

In the context of the AI integrated chatbot, HTML is used to design the layout of the chatbot interface on the university website. It structures the content in a way that ensures the chatbot is correctly displayed and accessible to users. By using HTML, the development team can create a well-organized and easily navigable interface, which is crucial for a positive user experience. Elements such as divs, buttons, and input fields are used to build the chatbot's layout, ensuring that it is functional and aesthetically pleasing. The semantic nature of HTML also helps in making the chatbot accessible to users with disabilities, as screen readers and other assistive technologies can interpret the structured content effectively.

HTML's role extends beyond mere content structuring; it also facilitates the integration of various multimedia elements and interactive features. For instance, incorporating multimedia content like images and videos can enhance the chatbot's responses, providing richer and more engaging information to users. HTML forms and input elements allow users to interact with the chatbot, submitting queries and receiving responses in real-time. These interactive capabilities are essential for creating an effective and engaging user experience, which is a primary goal of the AI integrated chatbot project.

Furthermore, HTML's compatibility with other web technologies ensures that the chatbot can be seamlessly integrated into the university's existing web infrastructure. By working in conjunction with CSS and JavaScript, HTML provides a robust foundation for building a cohesive and interactive chatbot interface. This interoperability ensures that the chatbot not only looks good but also performs well across different browsers and devices. This cross-compatibility is crucial for ensuring that all users, regardless of their device or browser, can access and benefit from the chatbot's capabilities.

2.4 CSS

CSS (Cascading Style Sheets) is a style sheet language used to describe the presentation of a document written in HTML. It controls the visual and layout aspects of web pages, including colors, fonts, spacing, and positioning. CSS allows developers to separate content from design, making it easier to maintain and update the look and feel of a website. By applying styles to HTML elements, CSS ensures that web pages are visually appealing and consistent with branding guidelines. This separation of content and presentation is crucial for maintaining a clean and organized codebase, which is easy to manage and update.

For the AI integrated chatbot, CSS is crucial in designing an attractive and consistent user interface. It ensures that the chatbot aligns with the university's branding, providing a visually appealing and intuitive experience for users. CSS is used to style the chatbot's interface elements, making them aesthetically pleasing and ensuring they function well on different devices and screen sizes. By using responsive design techniques, CSS ensures that the chatbot interface adapts to various screen resolutions, providing an optimal viewing experience on desktops, tablets, and smartphones. This responsiveness is essential for reaching a broad audience, including prospective students who may access the chatbot from different devices.

CSS also allows for the implementation of animations and transitions, which can enhance the user experience by making interactions with the chatbot more engaging and dynamic. For example, smooth transitions when opening the chatbot window or subtle animations when the chatbot sends a response can make the interface feel more interactive and lively. These visual enhancements contribute to a more polished and professional user experience, which is important for maintaining a positive impression of the university's digital presence.

Additionally, CSS frameworks like Bootstrap and Materialize provide pre-designed components and styles that can speed up the development process. These frameworks offer a consistent and responsive design out of the box, reducing the time and effort required to create a visually appealing interface. For the AI integrated chatbot, using a CSS framework can ensure that the interface looks professional and functions well across different devices and browsers. This consistency and professionalism are crucial for creating a user-friendly and engaging experience, which is key to the success of the chatbot in providing information and assistance to prospective students.

2.5 Bard API

The Bard API is a language model API similar to Olama, offering advanced text processing and generation capabilities. By integrating the Bard API, chatbots can significantly enhance their response quality, making interactions more precise and contextually appropriate. This improvement is crucial for creating a seamless user experience, especially in environments requiring detailed and nuanced conversations.

One of the key benefits of the Bard API is its ability to refine chatbot responses. This refinement allows the chatbot to handle complex queries with greater sophistication, providing users with accurate and comprehensive information. As a result, the interactions become more engaging and informative, leading to higher user satisfaction. This capability is particularly valuable in educational settings where users may have intricate questions about programs, admissions, and campus life. Furthermore, the Bard API's advanced text processing capabilities make it a powerful tool for ongoing improvements. As the chatbot interacts with more users, it can learn from the diverse range of questions and responses, continually enhancing its ability to provide precise and relevant information. This adaptability ensures that the chatbot remains a valuable resource over time, capable of evolving alongside the changing needs and expectations of prospective students. By leveraging the Bard API, Amrita University can maintain a cutting-edge communication tool that not only meets but exceeds user expectations, fostering a positive and informative experience for all users.

For institutions like Amrita University, using the Bard API can transform how prospective students interact with their chatbot. By delivering more nuanced and sophisticated responses, the chatbot can better address the diverse and detailed inquiries of prospective students. This not only improves the overall user experience but also helps the university present itself as a technologically advanced and responsive institution, ultimately aiding in student recruitment and engagement.

Chapter 3: Design and Development

The design and development stages outline the structured approach followed by Amrita University in creating the chatbot prototype. This section covers the initial project initiation, HTML page development, feedback collection through Google Forms, and the integration of frontend and backend components. The AI chatbot functional flowchart provides a visual representation of the development process, highlighting the key steps involved in processing user queries and generating appropriate responses.

Figure 1 showcases the development stages of the chatbot project at Amrita University, starting from the initiation phase where objectives and scope are defined, progressing to Stage 01 where foundational activities are undertaken. It includes the development of an HTML page to establish the core framework of the web application, followed by the creation of a Google Form to collect feedback from the university community. The collected opinions are then evaluated to shape the project's direction, ensuring that user needs and preferences are prioritized.

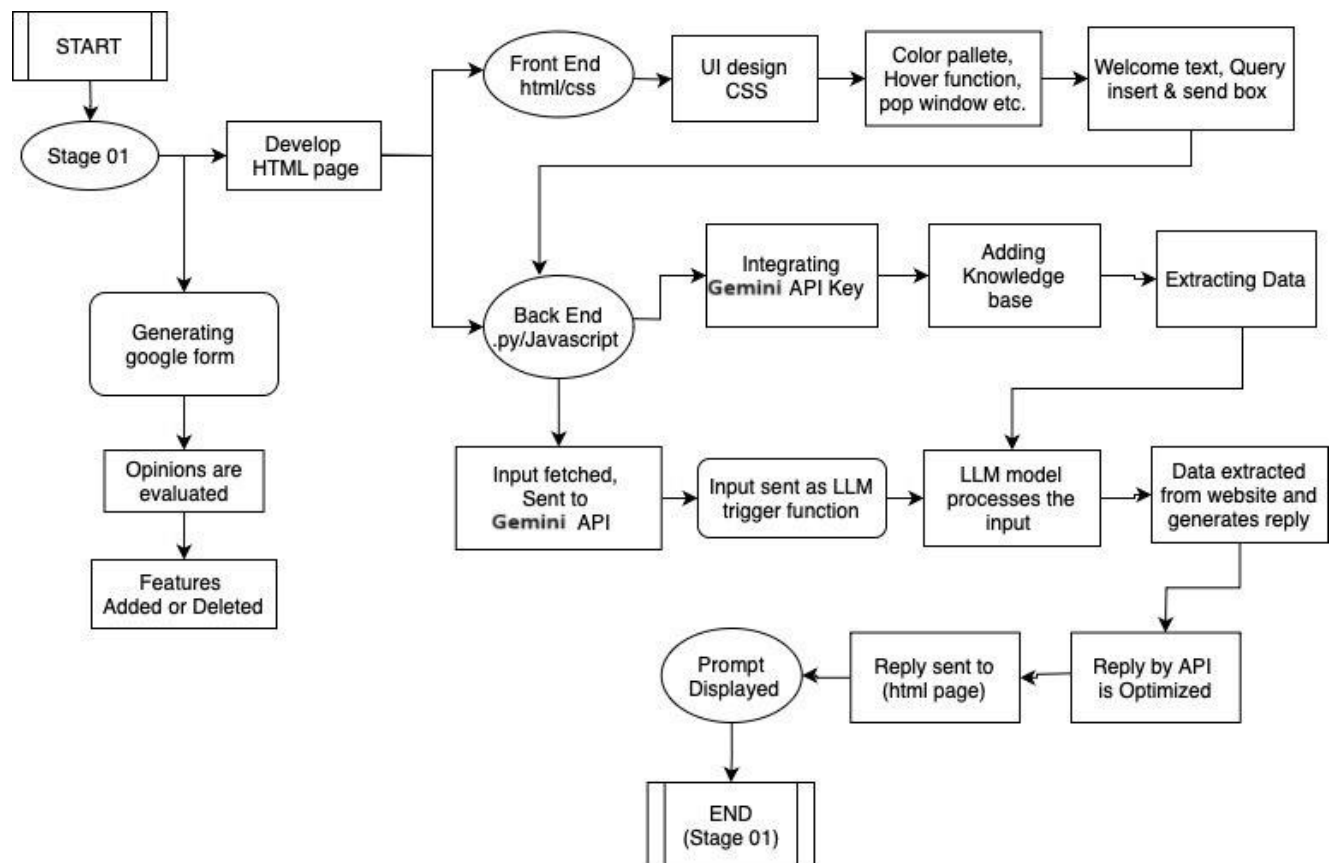


Fig 1: Development Stages of Chatbot

Start and Stage 01

The project at Amrita University begins at the **START** block, representing the initiation phase where the overall objectives and scope are defined. This leads into **Stage 01**, which marks the commencement of the first development phase. Here, the foundational activities are undertaken to lay the groundwork for the project, ensuring all necessary preparations and planning are completed.

Developing HTML Page

The next step involves **Develop HTML Page**, where the team creates the basic structure of the web application using HTML. This step is crucial as it establishes the core framework of the site, which includes setting up the essential elements and layout. This HTML page will serve as the foundation upon which other features and functionalities will be built.

Generating Google Form

After the HTML page is developed, a **Google Form** is generated to collect opinions and feedback from the university community, including students, faculty, and staff. This form is designed to gather valuable input on user needs, preferences, and expectations regarding the new system. It typically includes questions aimed at understanding user requirements and areas for improvement.

Evaluating Opinions

Once the feedback is collected through the Google Form, these **Opinions are evaluated**. This evaluation involves analyzing the responses to identify common themes, suggestions, and specific needs. The insights gained from this feedback are critical in shaping the project's direction, helping the development team to prioritize features and functionalities that will most benefit the users at Amrita University.

Figure 2, on the other hand, presents the AI Chatbot Functional Flowchart, depicting the sequence of user interactions with the chatbot. It emphasizes the process where users tap buttons to initiate queries and receive real-time responses, illustrating the chatbot's efficient handling of user inputs and its ability to provide prompt and relevant feedback. Together, these figures highlight both the systematic development process and the functional operation of the chatbot system.

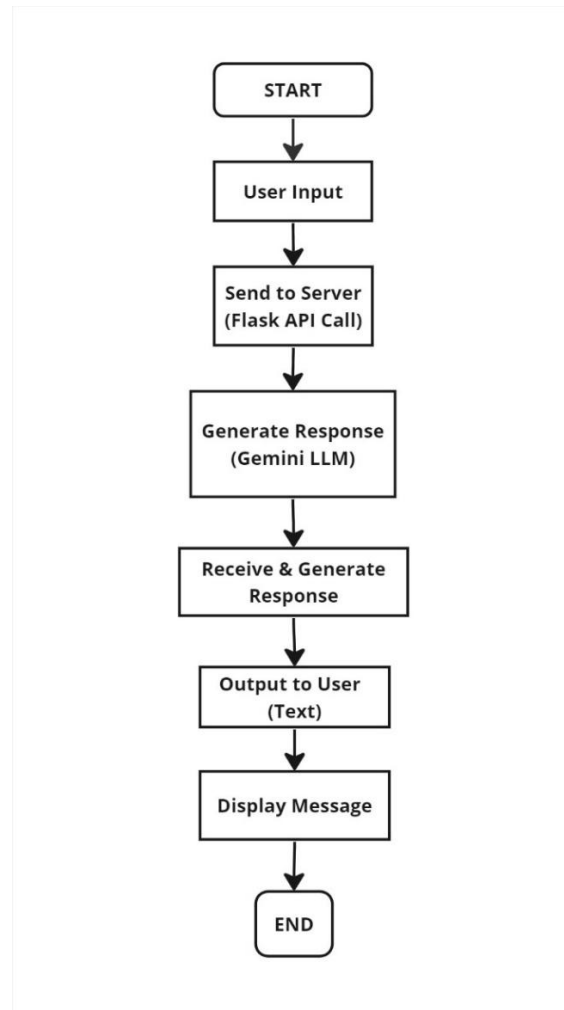


Fig 2: AI Chatbot Functional Flowchart

Front End Development (HTML/CSS)

The **Front End (HTML/CSS)** development phase focuses on enhancing the user interface of the HTML page. This involves using CSS to style the HTML elements, ensuring the website is visually appealing and user-friendly. The design process considers the needs of the university's diverse user base, making the interface intuitive and accessible.

UI Design with CSS

UI Design CSS is an integral part of this phase, where detailed styling is applied to create a cohesive and aesthetically pleasing user interface. The CSS code defines the visual aspects such as layout, colors, fonts, and spacing, ensuring a professional and attractive design that aligns with Amrita University's branding.

Adding Design Elements

Incorporating elements like a **Color palette, Hover function, pop window etc.**, adds interactivity to the user interface. These design elements make the website more engaging and user-friendly. For instance, a well-chosen color palette enhances visual appeal, while hover functions provide immediate feedback to user actions, improving the overall user experience.

Adding Functional Elements

Adding **Welcome text, Query insert & send box** involves integrating functional components that facilitate user interaction. The welcome text provides a friendly introduction, guiding users on how to navigate the site. The query insert box allows users to input their questions or concerns, and the send box enables them to submit these queries for further processing.

Back End Development (Python/JavaScript)

The **Back End (Python/JavaScript)** development is focused on the server-side logic of the application. This involves using programming languages like Python or JavaScript to handle data processing, API integrations, and other server-side functionalities. The backend is responsible for managing the data flow between the user interface and the server, ensuring smooth and efficient operations.

Integrating Gemini API Key

Integrating the Gemini API Key is a critical step for enabling secure communication with the Gemini API. This integration involves setting up the necessary authentication mechanisms to allow the backend to interact with the API securely. The API key acts as a credential, authorizing the backend to send requests and receive responses from the Gemini API.

Adding Knowledge Base

Adding a Knowledge Base involves incorporating a repository of information that the chatbot can use to generate responses. This knowledge base contains pre-defined answers, facts, and other relevant data, enhancing the chatbot's ability to provide accurate and informative replies to user queries.

Extracting Data

Extracting Data from websites and other sources involves setting up mechanisms to fetch relevant information in real-time. This ensures that the chatbot has access to up-to-date data, which is crucial for generating accurate and timely responses to user queries. The data extraction process includes web scraping and API calls to gather necessary information.

Input Processing and API Interaction

When a user submits a query, the system **fetches the input** and sends it to the Gemini API for processing. This step involves capturing the user's query and forwarding it to the API, which then processes the input using advanced language models.

Input as LLM Trigger Function

The **input is sent as a trigger function** to the Language Learning Model (LLM) within the Gemini API. This trigger initiates the processing of the input, leveraging the LLM's capabilities to analyze the query and generate a relevant response.

LLM Model Processing

The **LLM model processes the input**, utilizing its sophisticated algorithms and machine learning techniques to understand the context and content of the query. It generates a response based on the available data and knowledge base, ensuring the reply is accurate and contextually appropriate.

Data Extraction and Reply Generation

During this step, the **data extracted from the website** is used by the LLM to formulate a comprehensive response. The LLM processes this data to generate a reply that addresses the user's query effectively, ensuring the information provided is detailed and useful.

Reply Handling

The **reply generated by the API** undergoes optimization to ensure clarity and relevance. This optimization process involves refining the language, structure, and content of the response to enhance its quality, making it more user-friendly and informative.

Sending Reply to HTML Page

The optimized reply is then sent back to the HTML page, where it is displayed to the user. This involves updating the front-end interface with the new information, ensuring the user receives the response promptly and in a format that is easy to understand. The system dynamically updates the webpage, seamlessly integrating the response into the existing layout without requiring a page refresh. This real-time update mechanism enhances the user experience by providing instant feedback and maintaining the flow of interaction. The clear presentation of the reply, formatted for readability and relevance, ensures that users can easily comprehend and act on the information provided.

Prompt Display

Finally, the **prompt is displayed** to the user on the web page. This marks the completion of the interaction cycle, providing the user with the information they requested. The user interface is updated to show the generated response, concluding the query processing workflow.

End Stage 01

The process culminates in the **END (Stage 01)** block, signifying the completion of the first stage of the project. This stage includes the initial setup, development, and integration tasks necessary to create a functional prototype that can handle user interactions and provide accurate responses. The successful completion of this stage lays the foundation for further development and enhancements in subsequent stages.

Chapter 4: Demonstration/Evaluation and Results

This section showcases the user interaction process with the chatbot, demonstrating how users can tap on buttons to initiate queries and receive real-time responses. The images illustrate the smooth flow of information exchange between users and the chatbot, emphasizing the importance of prompt and user-friendly interfaces in providing satisfactory experiences. By highlighting these interactions, the section underscores the efficiency and responsiveness of the chatbot system, which is crucial for maintaining user engagement and satisfaction.

Figures 3, 5, and 7 illustrate the user tapping on a button, each followed by a response generated in **Figures 4, 6, and 8**, respectively, demonstrating the system's reaction to the tap actions. These figures reveal the immediacy of the chatbot's responses, reflecting its capability to provide quick feedback. This rapid interaction ensures that users can efficiently navigate through the system, making their experience seamless and intuitive.

Figures 9, 10, and 11 depict responses generated after query processing, showcasing the system's ability to handle and respond to queries. These figures highlight the chatbot's proficiency in managing more complex interactions, where it processes and generates detailed responses based on user input. The depiction of these query responses emphasizes the depth of the chatbot's understanding and its potential to deliver comprehensive answers, thereby enhancing the overall user experience



Fig 3: Tap on the Button



Fig 4: Response generated after Tap action

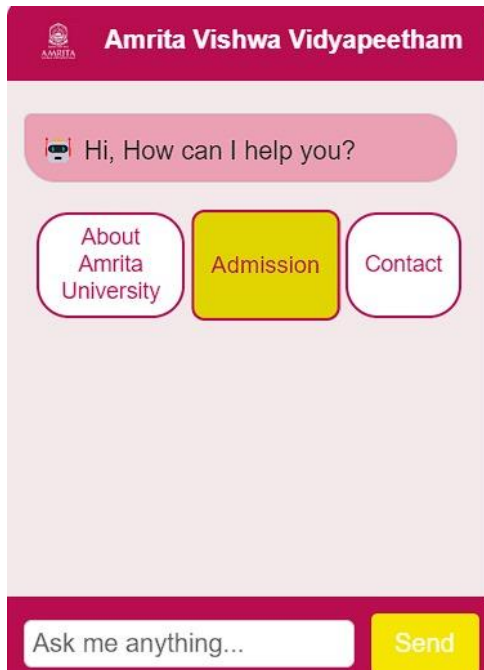


Fig 5: Tap on the Button



Fig 6: Response generated after Tap action

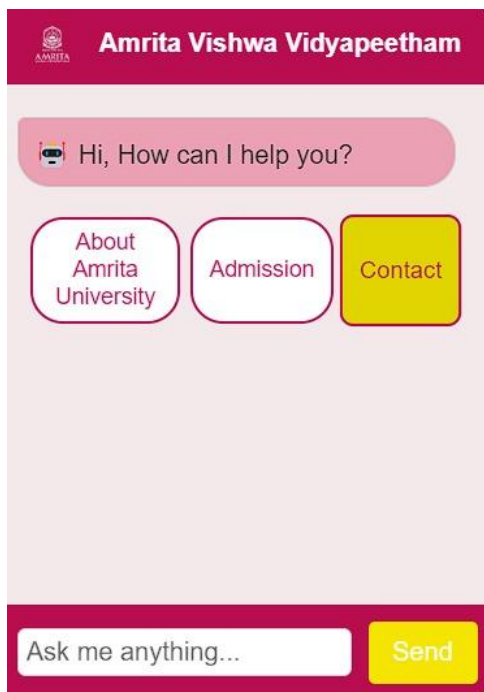


Fig 7: Tap on the Button

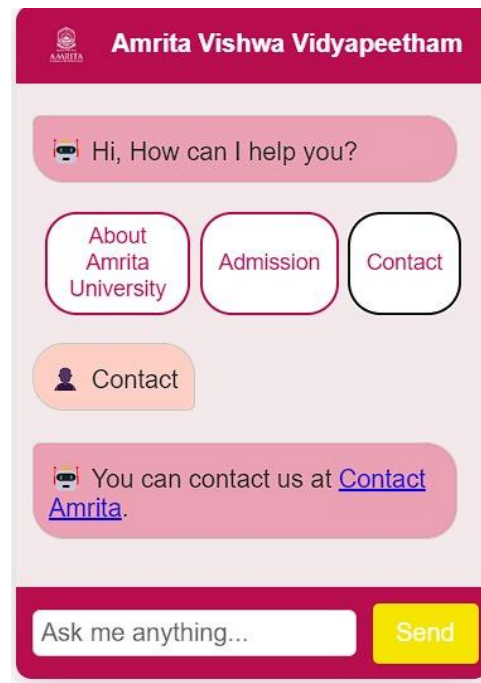


Fig 8: Response generated after Tap action

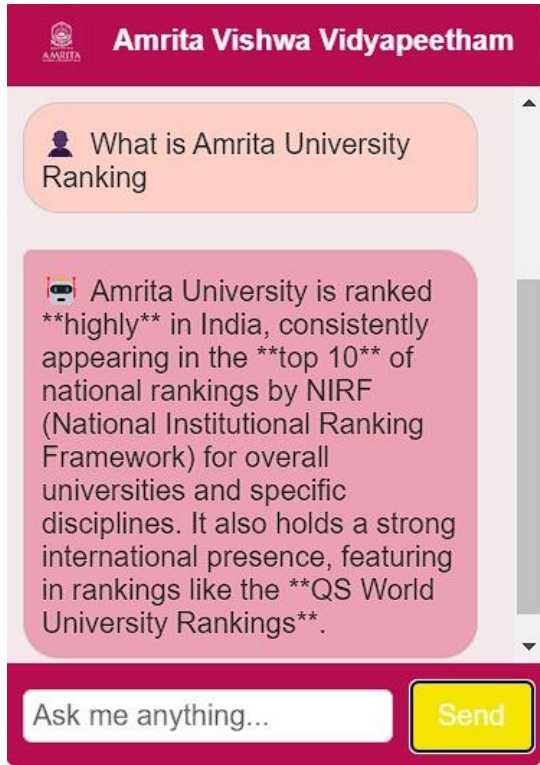


Fig 9: Response generated after Query processing

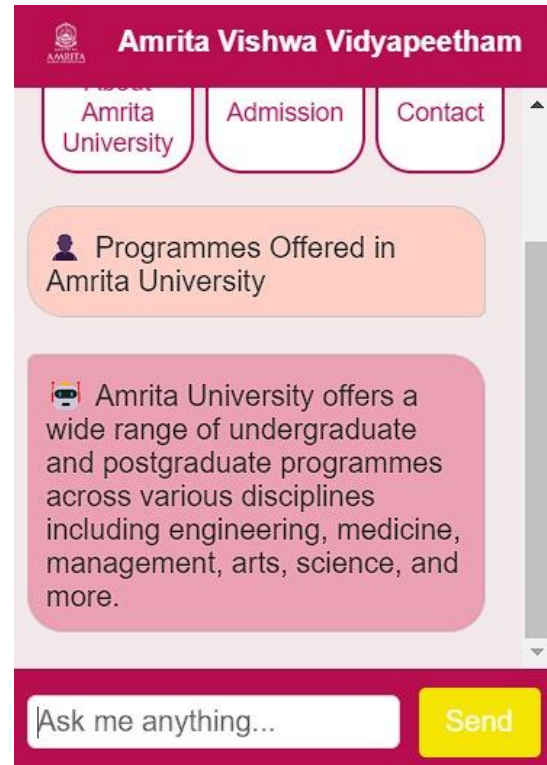


Fig 10: Response generated after Query Processing



Fig 11: Response Generated after Query Processing

Chapter 5: Conclusion and Future work

The conclusion reflects on the successful development of the Amrita University chatbot using Flask and Google's Gemini API. It highlights the chatbot's ability to deliver real-time, accurate, and user-friendly information to users. The future work section outlines plans for enhancing the chatbot's responses, knowledge base, and features to provide more detailed and contextually appropriate information. This continuous improvement ensures that the chatbot remains a valuable resource for the university community, meeting their evolving needs and expectations.

Conclusion

The development of the Amrita Vishwa Vidyapeetham chatbot using Flask and Google's Gemini API represents a significant advancement in providing real-time, accurate, and user-friendly information. This chatbot serves as a valuable tool for prospective students, current students, and other stakeholders by offering instant responses to queries about the university, admissions, and contact details. The integration of a sophisticated generative AI model ensures that the responses are both relevant and precise, enhancing the overall user experience. By employing a combination of backend and frontend technologies, the project successfully delivers a seamless and interactive platform for user engagement.

Future Work

By following this comprehensive plan, the university chatbot can be significantly enhanced to provide detailed, relevant, and contextually appropriate responses. Continuous improvement through feedback and regular updates will ensure the chatbot remains a valuable resource for users, offering an improved interactive experience that meets the specific needs of the university community.

To enhance the university chatbot and ensure it provides more specific and detailed responses, future work will focus on fine-tuning the model, creating a detailed knowledge base, and implementing additional features to improve performance and user experience. This involves collecting comprehensive university-specific data, such as course catalogs, faculty details, admission procedures, event schedules, campus facilities, and FAQs, and properly annotating this data for training purposes. The model will be fine-tuned using this annotated data, with continuous evaluation and iteration based on performance metrics and user feedback. A comprehensive knowledge base will be developed, managed through a content management system (CMS), and integrated with the chatbot to dynamically retrieve specific information, with regular updates to ensure the chatbot's responses remain current and accurate.

Enhancements to the chatbot's contextual understanding will be achieved through context management systems and state management techniques, allowing for coherent and relevant responses across multi-turn conversations. New features such as quick reply buttons, forms, links to detailed resources, multilingual support, and voice interaction capabilities will be introduced to enhance the user experience. Additionally, robust data privacy measures and compliance checks will be implemented to protect user information and ensure adherence to regulations, with mechanisms for managing user consent. This structured plan aims to create a more effective and user-friendly university chatbot.

References

1. J. S, J. S, S. M and G. R. Hemalakshmi, "College Enquiry Chatbot," *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, Vellore, India, 2023, pp. 1-4.
2. P. M, S. P, K. M, J. M, D. V and G. K. R S, "Student Chabot for University Admission Using Artificial Intelligence," *2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*, GHAZIABAD, India, 2023, pp. 512-515.
3. B. R. Ranoliya, N. Raghuwanshi and S. Singh, "Chatbot for university related FAQs," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India, 2017, pp. 1525-1530.
4. A. D. R. N, S. Sravya Thumati and S. Reyya, "An Efficient Deep Learning Based Chatbot for GRIET," *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballari, India, 2022, pp. 1-5.
5. J. S, J. S, S. M and G. R. Hemalakshmi, "College Enquiry Chatbot," *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, Vellore, India, 2023, pp. 1-4.
6. G. A. Santos, G. G. de Andrade, G. R. S. Silva, F. C. M. Duarte, J. P. J. D. Costa and R. T. de Sousa, "A Conversation-Driven Approach for Chatbot Management," in *IEEE Access*, vol. 10, pp. 8474-8486, 2022.
7. G. Attigeri, A. Agrawal and S. V. Kolekar, "Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis," in *IEEE Access*, vol. 12, pp. 29633-29647, 2024.
8. G. S. Sai Vikas, I. D. Kumar, S. A. Shareef, B. R. Roy and G. Geetha, "Information Chatbot for College Management System Using Multinomial Naive Bayes," *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2021, pp. 1149-1153.
9. L. -K. Lee, Y. -C. Fung, Y. -W. Pun, K. -K. Wong, M. T. -Y. Yu and N. -I. Wu, "Using a Multiplatform Chatbot as an Online Tutor in a University Course," *2020 International Symposium on Educational Technology (ISET)*, Bangkok, Thailand, 2020, pp. 53-56.