CLOUD-BASED STUDENT INFORMATION CHATBOT SYSTEM

A major project report submitted in partial fulfilment of the requirement for the award of a degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

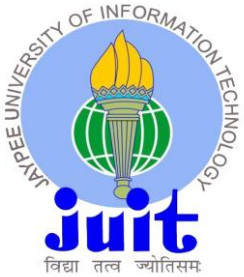
Yash Srivastava (201423)

Kartikay Narula (201427)

Under the guidance & supervision of

Mr. Prateek Thakral

Assistant Professor (Grade-II)



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan-173234 (India)

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **"Cloud-based Student Information Chatbot System"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my work carried out over a period from August 2023 to May 2024 under the supervision of **Mr. Prateek Thakral** (Assistant Professor (Grade-II), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Student Name: Yash Srivastava
Roll No.: 201423

Student Name: Kartikay Narula
Roll No.: 201427

This is to certify that the above statement by the candidate is true to the best of my knowledge.

Supervisor Name: Mr. Prateek Thakral
Designation: Assistant Professor (Grade II)
Department: Computer Science and Engineering and Information Technology
Dated:

CERTIFICATE

This is to certify that the work being presented in the project report titled "**Cloud-based Student Information Chatbot System"** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Yash Srivastava & Kartikay Narula with Roll Number 201423 & 201427 respectively during the period from August 2023 to May 2024 under the supervision Mr. Prateek Thakral (Assistant Professor (Grade-II), Department of Computer Science & Engineering and Information Technology)

Student Name: Yash Srivastava
Roll No.: 201423

Student Name: Kartikay Narula
Roll No.: 201427

The above statement is correct to the best of my knowledge.

Supervisor Name: Mr. Prateek Thakral
Designation: Assistant Professor - (Grade-II)
Department: Computer Science and Engineering and Information Technology

ACKNOWLEDGEMENT

Firstly, we express our heartiest thanks and gratefulness to almighty God for his divine blessing making it possible to complete the project work successfully. We are grateful to our supervisor **Mr. Prateek Thakral (Assistant Professor (Grade II))**, Department of Computer Science & Engineering and Information Technology) for carrying out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, who have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Student Name: Yash Srivastava
Roll No.: 201423

Student Name: Kartikay Narula
Roll No.: 201427

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The latest development of the Internet has brought the world into our hands. Everything happens on the Internet from passing information to purchasing something. The Internet made the world a small circle. This project is also based on the Internet, this paper is on the System that solves the diverse queries of students, parents, and faculty members. Development of a Cloud-based Student Information Chatbot System, harnessing cloud computing and natural language processing (NLP) to resolve the information accessibility within educational institutions.

The proposed chatbot will function as a cognitive interface, enabling smooth communication for consumers looking for class schedules, test results, necessary materials, and administrative notifications. Our main goal is to employ cloud infrastructure to overcome traditional limitations while guaranteeing scalability, and robustness, and accommodating different user engagement levels. Additionally, through sophisticated authorization and authentication procedures, utmost attention will be paid to data protection and privacy.

This creative project, which aims to improve accessibility and participation, fits well with how modern education is changing. Our project aims to revolutionize educational communication paradigms by promoting technology-driven solutions, which will significantly alter the accessibility and interchange of knowledge.

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

A major step toward transforming education and moving it toward a time of seamless communication and information access has been taken with the introduction of the Cloud-Based Student Information Chatbot System. This revolutionary system represents the nexus of cutting-edge technology capabilities with a deep understanding of the problems ingrained in educational institutions.

This system is a dynamic and transformational platform that can effectively address and bridge gaps in maintaining student data and resolving a variety of queries by utilising the combined strengths of the MERN stack and Natural Language Processing (NLP) [1]. This creative method is essentially the result of combining cutting-edge technology with a deep comprehension of the complex issues that the educational field faces. The system aims to provide an intuitive, adaptable, and efficient experience that meets the various needs and preferences of teachers, parents, and students in educational institutions by utilising cloud computing capabilities and incorporating state-of-the-art natural language processing algorithms.

This chatbot system's architecture has been carefully designed to deliver prompt answers to a wide range of questions. When it comes to questions regarding class schedules, test results, necessary resources, or important administrative updates, the system's architecture prioritizes responsiveness to a range of user needs in addition to timely information delivery.

Furthermore, the design philosophy of the system is centred on overcoming the constraints that are inherent in conventional communication channels like emails and portals. This chatbot technology ushers in a new age of stakeholder interaction with information through an intelligently built conversational interface. It is intended to transform communication in learning environments by creating a setting that is not only instructional but also dynamic, entertaining, and incredibly effective.

In conclusion, the Cloud-Based Student Information Chatbot System is a paradigm change in the way that information is accessible, shared, and customised inside educational institutions. It is more than just a technology improvement in the field of education.

1.2 PROBLEM STATEMENT

There are ongoing issues in the educational environment linked to efficiently organising and sharing important student-related data. Conventional communication channels sometimes don't offer a smooth and prompt way to answer the various questions that teachers, parents, and students have. Despite their widespread use, online portals and emails have difficulty providing real-time, individualised replies, which results in inefficiencies and user discontent.

There is a communication gap in educational institutions as a result of the limits of the present methods, which impede responsiveness, fast data retrieval, and individualised contact. Students frequently find difficulties in getting timely access to relevant academic materials, while parents and staff members face challenges in receiving timely information or effectively resolving problems.

The primary need is for an advanced solution that ensures data security, accessibility, and effective stakeholder engagement in addition to addressing these communication gaps. This gap is intended to be filled by the Cloud-Based Student Information Chatbot System, which offers an extensive and intelligent platform that can handle a variety of queries while guaranteeing data confidentiality and privacy.

1.3 OBJECTIVE

1. NLP-driven Query Resolution: Implementing an NLP-powered chatbot capable of understanding and resolving diverse student inquiries promptly and accurately.
2. Intuitive User Interface: Developing an intuitive, user-friendly interface that ensures seamless navigation and interaction for users of varying technical abilities.

3. Scalability and Resilience: Utilising cloud infrastructure to guarantee scalability and resilience, accommodating varying roles of user engagement without compromising performance.
4. Robust Authentication Mechanisms: Implementing robust user authentication and authorization mechanisms to safeguard student data and system integrity.
5. Personalised Information Delivery: Making the system deliver personalised student-related information promptly and accurately upon request.
6. Academic Schedule Management: Developing features allowing students to access and manage their academic schedules efficiently.
7. Examination Outcome Retrieval: Providing a mechanism for students to retrieve examination outcomes promptly through the chatbot interface.
8. Access to Academic Resources: Facilitating easy access to vital academic resources and administrative updates, ensuring students, parents, and faculty are well-informed.
9. Responsive Communication Channel: Establishing a responsive communication channel that caters to inquiries from students, parents, and faculty.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

Beyond technological innovation, the Cloud-Based Student Information Chatbot System is significant because it is a catalyst that has the potential to completely change the way that information is distributed within educational institutions. Its appearance signifies a turning point, tackling the innate inadequacies found in traditional methods of communication. This system's transformational potential is fundamental to its relevance. Through the integration of cutting-edge technology and a user-centred design methodology, it seeks to address the deficiencies. afflicting the current channels of communication within educational environments. These channels are unable to satisfy the changing needs of teachers, parents, and students since they are frequently marked by inefficiencies in information transmission, delays, and a lack of personalization.

The critical necessity to develop a groundbreaking platform beyond simple information sharing inspired the creation of this system. It aims to provide a communication channel that breaks through obstacles to accessibility, engagement, and security while raising the bar for educational communication to never-before-seen levels.

The essential need to improve accessibility and stakeholder participation lies at the heart of this drive. The system aims to provide an inclusive platform that meets the varied demands and preferences of its users through its clever design and flexibility. Through easy access to calendar management, critical information acquisition, and academic resources, students may create a proactive and engaged learning environment.

Furthermore, the system stands out as a trailblazer in educational technology due to its dedication to protecting data security and privacy. This system prioritizes strict security controls and privacy rules to protect sensitive student information from unauthorised access or breaches, a critical concern for educational institutions.

At its core, the goal of completely changing the paradigm of educational communication is what motivates the Cloud-Based Student Information Chatbot System. It seeks to provide a safe, interesting, and easily navigable environment that sets a new benchmark for educational communication channels in addition to expediting the distribution of knowledge.

1.5 ORGANIZATION OF PROJECT REPORT

Chapter 1: Introduction

- **1.1** The opening chapter of our project report, contains the overview of the whole project.
- **1.2** The Introduction chapter includes the complete overview of the project, the opening chapter also contains the problem statement, it also contains the main usage of the project, its advantages, use cases.
- **1.3** Objective of the project creating a LLM based NLP chatbot integrating with cloud and student portal as well as admin portal.
- **1.4** The motivation for the team to work on this project is also mentioned in the Introduction chapter highlighting its significance as well.

Chapter 2: Literature Survey

- 2.1 Overview of Relevant Literature: Summarising existing studies and information relevant to the project's scope.
- 2.2 Key Gaps in the Literature: Identifying areas where existing literature lacks information or requires further exploration.

Chapter 3: System Development

- 3.1 Requirements and Analysis: Detailing the needs and analysing various aspects before the system's development.
- 3.2 Project Design and Architecture: Planning and outlining the overall structure and design approach for the system.
- 3.3 Data Preparation: Preparing and organising the data required for the system.
- 3.4 Implementation: Actual coding, using specific tools, and techniques, accompanied by code snippets and algorithms.
- 3.5 Key Challenges: Discuss encountered challenges during development and their resolution.

Chapter 4: Testing

- 4.1 Testing Strategy: Explaining the strategy and tools employed for system testing.
- 4.2 Test Cases and Outcomes: Presenting specific cases tested and their resulting outcomes.

Chapter 5: Results and Evaluation

- 5.1 Results: Displaying findings, interpretations, and conclusions drawn from the outcomes.
- 5.2 Comparison with Existing Solutions: Comparing the project's outcomes with existing solutions if applicable.

Chapter 6: Conclusions and Future Scope

- 6.1 Conclusion: Summarising key findings, recognizing limitations, and contributions to the project's field.
- 6.2 Future Scope: Suggesting potential future developments or areas for further enhancements or extensions.

CHAPTER 2: LITERATURE SURVEY

2.1 OVERVIEW OF RELEVANT LITERATURE

Table 2.1: Literature Survey on A Scalable Self-Learning Chatbot for Classroom

Title	A Scalable Self-Learning Chatbot for Classroom [16]
Journal/Conference	IEEE Annual Computing and Communication Workshop and Conference (CCWC) [1]
Year	2022
Summary	The paper titled "A Scalable Self-Learning Chatbot for Classroom," presented at the IEEE Annual Computing and Communication Workshop and Conference (CCWC) in 2022, introduces a sophisticated Classroom Assistant chatbot. This chatbot is built using Dialogflow, Natural Language Processing (NLP), and the Google Cloud Platform (GCP) and is specifically designed to enhance the management of student queries, improve learning experiences, and support educators within the Slack environment. By leveraging advanced AI technologies, the chatbot aims to streamline communication and provide timely assistance to both students and teachers, thereby fostering a more interactive and efficient educational setting.[16] Despite its innovative approach and the potential benefits it offers, the paper identifies several notable limitations in the chatbot's functionality. One significant limitation is the potential for compatibility issues with other platforms. Since the chatbot is deeply integrated into Slack, adapting its functionalities to other communication tools or educational platforms may present challenges.This over-reliance on Slack could restrict its usability and effectiveness in educational institutions that utilize different platforms, thus limiting its broader applicability.

	Additionally, the complexities involved in achieving seamless integration with other systems and platforms pose another hurdle. Integrating the chatbot into various educational environments requires overcoming technical and operational barriers, which could impede its widespread adoption. These complexities might deter institutions from attempting to incorporate the chatbot into their existing ecosystems. The paper also highlights concerns about restricted user engagement. While the chatbot functions effectively within the Slack environment, its ability to engage users in other contexts remains uncertain. This limitation suggests that the chatbot may not be as effective in fostering interaction and engagement outside of its current platform, potentially diminishing its overall impact. These identified limitations underscore the need for the chatbot to evolve beyond its current dependencies and enhance its adaptability. To ensure its versatility and usability across diverse educational environments or communication platforms, addressing these challenges is crucial. Developing solutions to overcome compatibility issues, simplifying integration processes, and improving user engagement mechanisms would significantly broaden the chatbot's applicability. By doing so, the Classroom Assistant could facilitate its seamless integration into a variety of educational contexts, thereby maximizing its potential benefits and ensuring it can meet the needs of a wider range of users.

Table 2.2: Literature Survey on Web-Based College Chatbot - SDABot

Title	Web-Based College Chatbot – SDABot [13]
Journal/Conference	International Journal for Research in Applied Science and Engineering Technology
Year	2021

Summary	The paper titled "Web-Based College Chatbot - SDABot," published in 2021 in the International Journal for Research in Applied Science and Engineering Technology, introduces a real-time chatbot specifically designed for college environments. This innovative chatbot is developed using Flask, SQLite Database, Machine Learning, and Search-Based Algorithms, and is intended for seamless integration into college websites. The chatbot aims to provide an interactive and adaptable platform for student engagement, facilitating the dissemination of information and enhancing the overall student experience.[13] Despite its promising design and potential benefits, the paper identifies several limitations inherent to the chatbot's functionality. One significant limitation is its reliance on a static knowledge base. The chatbot's responses are dependent on pre-programmed information, which may not always be up-to-date or comprehensive. This static nature can lead to challenges in maintaining the accuracy and relevancy of the information provided, especially as the college's data and policies evolve over time. Another limitation highlighted is the potential difficulty in synchronising the chatbot with website updates. As college websites are frequently updated with new information and resources, ensuring that the chatbot's knowledge base remains aligned with these changes can be complex and time-consuming. Failure to synchronise effectively may result in the chatbot providing outdated or incorrect information to users. The paper also points out the chatbot's susceptibility to occasional inaccuracies. Given its dependency on a predefined knowledge base and algorithms, there is a risk of the chatbot delivering incorrect or misleading responses. This issue underscores the need for regular

	updates and continuous monitoring to ensure the reliability of the information provided. Additionally, the chatbot faces constraints in conversational depth. While it can handle basic queries and provide straightforward information, its ability to engage in more complex or nuanced conversations is limited. This constraint reduces its effectiveness in addressing more detailed or sophisticated inquiries from students. Moreover, the chatbot's operational scope is limited. While it serves as a useful tool within the college website, its functionality may not extend to other platforms or applications, restricting its utility to a single environment. This limitation suggests that the chatbot might not fully meet the diverse needs of the college community across different contexts. Addressing these challenges is critical to enhancing the chatbot's overall effectiveness. Efforts to maintain and update the chatbot's knowledge repository regularly would ensure that the information provided is accurate, relevant, and comprehensive. Additionally, expanding the chatbot's conversational abilities and operational scope could significantly improve its utility as a valuable resource within college environments.

Table 2.3: Literature Survey on A Smart Virtual Assistant for Students

Title	A Smart Virtual Assistant for Students [14]
Journal/Conference	3rd International Conference on Applications of Intelligent Systems
Year	2020
Summary	The paper titled "A Smart Virtual Assistant for Students," presented at the 3rd International Conference on Applications of Intelligent Systems in 2020, introduces a cutting-edge virtual assistant designed to meet the diverse needs of college students. This virtual

	assistant is developed using Natural Language Processing (NLP), integrated with a Learning Management System (LMS), secured through TLS Encryption, and meticulously crafted following the Software Development Life Cycle (SDLC). The primary aim of this advanced assistant is to alleviate student challenges, reduce dropout rates, enhance counsellor outreach, and provide comprehensive chatbot services, thereby enriching the educational experience.[14] However, the paper also highlights several limitations that affect the virtual assistant's overall effectiveness. One prominent issue is the presence of language barriers, which can hinder effective communication. The virtual assistant may struggle to cater to students who speak different languages or dialects, potentially limiting its inclusivity and accessibility across a diverse student body. Another significant limitation is the absence of human interaction, which can lead to a sense of detachment among users. While the virtual assistant can handle numerous queries and tasks, the lack of a human touch may result in reduced emotional engagement and support, which are often critical in educational and counselling contexts. Technical constraints also pose challenges to the virtual assistant's operational capabilities. These constraints might affect the assistant’s performance, responsiveness, and ability to handle complex queries or scenarios, thereby limiting its utility and reliability. User adoption is another area of concern. Despite its advanced features, the virtual assistant may face resistance from students who are reluctant to use new technology or prefer traditional methods of seeking information and support. Encouraging widespread

	adoption and demonstrating the assistant's benefits are crucial to overcoming this resistance. Furthermore, there are significant concerns regarding user privacy. The virtual assistant's ability to handle sensitive student information necessitates robust privacy measures to safeguard user data. Ensuring that these privacy concerns are adequately addressed is essential to build trust and promote the virtual assistant's usage. These limitations underscore the formidable challenges in enhancing the virtual assistant's functionality and acceptance. Ensuring inclusivity across diverse linguistic backgrounds requires advanced language processing capabilities and support for multiple languages. Balancing automated responses with opportunities for human interaction can help maintain user engagement and emotional support. Overcoming technical constraints involves continuous improvement and optimization of the virtual assistant's capabilities. Promoting user adoption necessitates effective communication of the assistant's benefits and user-friendly design. Lastly, implementing stringent privacy measures is vital to protect user data and foster trust.

Table 2.4: Literature Survey on Smart College Chatbot Using ML and Python

Title	Smart College Chatbot Using ML and Python [5]
Journal/Conference	International Conference on System, Computation, Automation
Year	2020
Summary	The paper titled "Smart College Chatbot Using ML and Python," presented at the 2020 International Conference on System, Computation, Automation, and Networking, investigates the development of a specialized chatbot tailored for college environments. This innovative chatbot is developed using Natural Language Processing (NLP), Machine Learning (ML), the Natural

	Language Toolkit (NLTK), and WordPress. The paper delves into the chatbot's design, user interactions, and the integration of Artificial Intelligence (AI) components, showcasing its potential to enhance student engagement and streamline information dissemination in academic settings. [5] Despite its innovative approach and potential benefits, the paper identifies several limitations that affect the chatbot's overall effectiveness. A notable limitation is the absence of an in-depth discussion on system scalability. The paper does not provide detailed insights into how the chatbot would handle increased user volumes, particularly in high-demand scenarios. This omission raises concerns about the chatbot's ability to scale efficiently and maintain performance as the number of users grows. Without a robust scalability plan, the chatbot might struggle to deliver consistent and reliable service during peak usage times, potentially diminishing its utility and user experience. Another significant limitation is the lack of comprehensive security measures. The paper does not elaborate on the implementation of robust security protocols necessary to safeguard user data. This gap signifies potential vulnerabilities within the chatbot ecosystem, which could be exploited by malicious actors to compromise user privacy and data integrity. Ensuring the security of user information is paramount, especially in educational environments where sensitive personal data is often handled. The absence of detailed security strategies suggests a need for heightened attention to developing and implementing stringent security measures to protect user data and build trust among users.[10] Addressing these limitations is crucial to enhancing the chatbot's effectiveness and reliability. Detailed plans for system scalability are necessary to ensure the chatbot can handle increased user

	volumes without degradation in performance. This involves optimizing the backend infrastructure, employing load balancing techniques, and possibly leveraging cloud-based solutions to dynamically allocate resources based on demand. Moreover, implementing comprehensive security measures is essential to protect user data and maintain privacy. This includes employing encryption techniques for data transmission and storage, regular security audits, and adherence to best practices in cybersecurity. Ensuring that these protocols are in place will help safeguard user information and enhance the overall trust and credibility of the chatbot. By addressing these challenges, the "Smart College Chatbot Using ML and Python" can significantly improve its scalability, security, and overall effectiveness. This will enable it to better serve the needs of college environments, providing a reliable, secure, and user-friendly tool for student engagement and information dissemination.

Table 2.5: Literature Survey on AI-Based Student Bot

Title	AI-Based Student Bot for Academic Information System using Machine Learning [11]
Journal/Conference	International Journal of Scientific Research in Computer Science, Engineering, and Information Technology
Year	2019
Summary	The paper titled "AI-Based Student Bot for Academic Information System using Machine Learning," published in the International Journal of Scientific Research in Computer Science, Engineering, and Information Technology in 2019, introduces the innovative StudentBot system. This system is developed through the integration of Natural Language Processing (NLP), Dialogflow,

	Node.js, and MySQL Api.ai, providing a comprehensive chat-based interface designed to address queries related to college information. Leveraging advanced technologies such as NLP, AIML (Artificial Intelligence Markup Language), and Dialogflow, StudentBot facilitates interactive conversations and efficiently manages a database to handle academic inquiries. [11] Despite its promising advancements, StudentBot encounters several limitations that affect its overall functionality and effectiveness. One notable limitation is its relatively limited knowledge base. The current scope of information that StudentBot can provide is restricted, which may hinder its ability to fully satisfy diverse student inquiries. Expanding the knowledge repository is essential to ensure the chatbot can address a wider range of questions and provide more comprehensive support to users. Another challenge is interpreting diverse language nuances. While NLP technologies are employed, the system may struggle with understanding and accurately responding to the various ways in which students phrase their questions. Enhancing language comprehension capabilities is crucial for improving the chatbot's accuracy and reliability in interpreting and responding to user queries. The lack of customization options is also a significant limitation. Users have different needs and preferences, and the inability to customize the chatbot's responses or interface limits its flexibility and user satisfaction. Enabling greater customization would allow the chatbot to better cater to individual user requirements and improve the overall user experience. Additionally, StudentBot's scope of functionalities is somewhat restricted. While it can handle basic academic inquiries, its ability

	to provide more complex information or perform advanced tasks is limited. Broadening the scope of its functionalities to encompass a wider range of academic information and user needs would significantly enhance its utility and effectiveness within educational environments. The paper highlights these limitations while also shedding light on the promising nature of AI-based academic chatbots like StudentBot. It emphasizes the critical areas for future enhancements and improvements to maximize the chatbot's utility. Addressing these limitations involves expanding the knowledge base, refining language comprehension capabilities, enabling greater customization, and broadening the scope of functionalities. By focusing on these areas, the development of StudentBot can continue to evolve, offering more robust and comprehensive support to students and enhancing its overall effectiveness within the educational landscape.

Table 2.6: Literature Survey on ChatBot for College Website

Title	ChatBot for College Website [12]
Journal/Conference	International Journal of Innovative Technology and Exploring Engineering
Year	2019
Summary	The paper titled "ChatBot for College Website," published in 2019 in the International Journal of Innovative Technology and Exploring Engineering, introduces a sophisticated chatbot designed for college websites. This chatbot leverages advanced technologies such as Deep Learning, Long Short Term Memory (LSTM), Recurrent Neural Network (RNN), TensorFlow, Word embedding, Sequence-to-Sequence model, and Flask. Developed through supervised learning methodologies, the chatbot achieved a notable accuracy rate of 92.89% with a final loss of 0.012 by the 150th

	epoch, indicating its high performance in understanding and responding to user queries. [12] Despite these impressive achievements, the paper highlights several inherent limitations within the chatbot's functionalities. One significant limitation is the deficiency in emotional intelligence. The chatbot struggles to understand and respond to the emotional nuances of user interactions, which can be crucial in educational environments where empathy and emotional support are important. Another limitation is the chatbot's reliance on extensive datasets. Developing and maintaining a comprehensive dataset for training the chatbot is resource-intensive and can be challenging. The effectiveness of the chatbot heavily depends on the quality and breadth of the data it is trained on, which can limit its adaptability and performance in dynamic or less predictable scenarios. The substantial training requirements present another challenge. Training the chatbot to achieve high accuracy involves considerable computational resources and time. This intensive training process can be a barrier to frequent updates and improvements, potentially leading to outdated responses as information changes. Furthermore, the chatbot exhibits limitations in human-like interaction. While it can handle basic queries effectively, its interactions may lack the natural flow and depth of a conversation with a human, which can impact user satisfaction and engagement. The paper also notes challenges in comprehending context. The chatbot may struggle with maintaining context over longer conversations or understanding nuanced context-specific queries, which can result in inaccurate or irrelevant responses.

	Managing ambiguities is another area where the chatbot encounters difficulties. Ambiguous questions or statements can confuse the chatbot, leading to incorrect answers or requests for clarification, which can frustrate users. Addressing these limitations is crucial for advancing the chatbot's sophistication and effectiveness. Enhancing emotional intelligence involves incorporating techniques that allow the chatbot to detect and appropriately respond to emotional cues. Reducing dependency on extensive datasets can be achieved by developing more efficient data augmentation and transfer learning methods. Optimizing training processes might involve using more efficient algorithms or hardware to decrease training time and resource consumption. Improving the chatbot's ability for human-like interaction can involve refining natural language generation models to produce more conversational and context-aware responses.

Table 2.7: Literature Survey on Chatbot: A User Service for College

Title	Chatbot: A User Service for College [17]
Journal/Conference	International Journal of Scientific Research in Computer Science, Engineering, and Information Technology
Year	2019
Summary	The paper titled "Chatbot: A User Service for College," published in 2019 in the International Journal of Scientific Research in Computer Science, Engineering, and Information Technology, introduces an advanced chatbot tailored to handle college-related queries. This chatbot leverages cutting-edge technologies, including Natural Language Processing (NLP), Microsoft Speech Recognition and Synthesis, XAMPP, Tomcat, Eclipse, Java Programming Language, and AWS (Amazon Web Services).

	Designed to cater to the needs of college students and staff, the chatbot employs NLP, AIML (Artificial Intelligence Markup Language), tokenization, and lemmatization techniques to facilitate interactions. It also offers functionalities such as providing maps and hostel-related information.[17] Despite its innovative design and functionalities, the paper highlights several limitations inherent in the chatbot's capabilities. One major challenge is language comprehension. The chatbot may struggle to understand and accurately respond to queries in diverse languages or dialects, which can limit its accessibility and effectiveness in a multicultural college environment. Context awareness is another limitation. The chatbot often finds it difficult to maintain context over extended conversations or to understand the nuanced context of queries. This limitation can result in responses that are irrelevant or incorrect, diminishing the user experience. The chatbot also exhibits deficiencies in emotional intelligence. While it can handle factual queries, it lacks the ability to understand and respond appropriately to the emotional tone of user interactions. This can be particularly important in educational settings where students may seek empathetic and supportive responses. Processing complex queries poses another significant challenge. The chatbot may struggle with intricate or multi-faceted questions, leading to inadequate or incomplete responses. This limitation underscores the need for more advanced query processing capabilities to handle a broader range of inquiries effectively.

	The reliance on extensive training data is also a concern. The chatbot's performance is heavily dependent on the quality and comprehensiveness of the training data it is provided with. Ensuring continuous and up-to-date training data is a resource-intensive process, and gaps in this data can lead to reduced accuracy and reliability. Security measures are another critical area of concern. Ensuring the privacy and security of user data is paramount, especially in an educational setting where sensitive personal information is handled. The paper indicates that there are concerns regarding the robustness of the chatbot's security protocols, which need to be addressed to safeguard user data and build trust. Addressing these limitations is essential for enhancing the chatbot's overall effectiveness and user satisfaction. Improving language comprehension can involve incorporating more sophisticated NLP models that support multiple languages and dialects. Refining context-awareness capabilities may require the integration of advanced memory networks or contextual understanding algorithms to maintain conversation continuity and relevance. Enhancing emotional intelligence involves developing techniques to detect and respond to the emotional tone of user interactions, which can improve the chatbot's ability to provide empathetic support. Optimizing query processing capabilities can be achieved by implementing more advanced AI algorithms capable of handling complex and multi-faceted queries. Reducing reliance on extensive training data involves employing data augmentation techniques and leveraging pre-trained models that can generalize better from less data. Finally, fortifying security measures is crucial to protect user data. This includes implementing

	robust encryption methods, regular security audits, and adherence to best practices in cybersecurity.

Table 2.8: Literature Survey on A Multi-Model And AI-Based College Bot

Title	A Multi-Model And AI-Based College Bot Management System (Aicms) For Professional Engineering Colleges [15]
Journal/Conference	International Journal of Innovative Technology and Exploring Engineering
Year	2019
Summary	The paper titled "A Multi-Model And AI-Based College Bot Management System (AICMS) For Professional Engineering Colleges," published in 2019 in the International Journal of Innovative Technology and Exploring Engineering, introduces an Intent-based College Bot. This chatbot is developed using advanced technologies such as Natural Language Processing (NLP), Dialogflow, Facebook Messenger, Google API.ai, and Natural Language Understanding (NLU) [1]. Primarily operating through Dialogflow on Facebook, the chatbot is designed to handle predefined queries and deliver corresponding responses, catering to the needs of professional engineering colleges.[15] Despite its innovative design, the paper identifies several limitations inherent to the chatbot's functionalities. One significant limitation is its reliance on a rule-based methodology. This approach restricts the chatbot's flexibility and adaptability, as it can only respond to queries that fit predefined rules and patterns. Evolving from a rule-based to a more dynamic approach would enhance the chatbot’s ability to handle a broader range of queries and provide more nuanced responses. Maintaining the quality of training data is another challenge highlighted in the paper. The effectiveness of the chatbot heavily depends on the quality and comprehensiveness of its training data.

	Ensuring that this data remains accurate, relevant, and up-to-date is crucial for the chatbot's performance. Poor quality or outdated data can lead to incorrect or irrelevant responses, undermining user trust and satisfaction. The limitations in language support also pose a significant challenge. The chatbot may struggle to effectively support multiple languages, limiting its accessibility to non-English speaking users. Expanding language support capabilities is essential to make the chatbot more inclusive and useful for a diverse student body. Dependencies on external components are another critical limitation. The chatbot relies on various external systems and services, such as Dialogflow and Facebook Messenger, which can introduce vulnerabilities and dependencies. Any issues or changes in these external components can directly impact the chatbot's functionality and reliability. Reducing these dependencies would enhance the chatbot's robustness and ensure more consistent performance. Addressing these limitations is crucial for advancing the chatbot's capabilities. Transitioning towards more adaptable and intelligent models involves incorporating machine learning techniques that allow the chatbot to learn from interactions and improve over time, rather than relying solely on predefined rules. Enhancing the quality and diversity of training data requires continuous data collection, curation, and updating processes to ensure the chatbot's responses remain accurate and relevant. Improving language coverage can be achieved by integrating more sophisticated NLP models that support multiple languages and dialects, making the chatbot more accessible to a wider audience. Mitigating dependencies on external systems involves developing

	more autonomous solutions and ensuring that critical functionalities are maintained in-house to reduce vulnerability to external changes.

Table 2.9: Literature Survey on Natural Language Processing based Jaro-Chatbot

Title	Natural Language Processing based Jaro-The Interviewing Chatbot [9]
Journal/Conference	3rd International Conference on Computing Methodologies and Communication (ICCMC)
Year	2019
Summary	The paper titled "Natural Language Processing based Jaro-The Interviewing Chatbot," presented at the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) in Erode, India, and subsequently added to IEEE Xplore on 29th August 2019, spotlights a groundbreaking system designed to transform recruitment processes. The innovative "Jaro" system revolutionizes interviews by leveraging sophisticated Natural Language Processing (NLP) technology. This system autonomously generates interview questions by analyzing candidates' resumes, thereby standardizing and streamlining the interview preparation process.[9] However, despite its pioneering approach, the development of Jaro reveals several limitations. One significant concern is the potential for biases within the NLP model. If the training data contains inherent biases, these can be perpetuated in the interview questions generated by Jaro, potentially leading to unfair assessments of candidates. Another limitation is the challenge of accurately assessing candidates' soft skills solely through automated means. Soft skills, such as communication, empathy, and teamwork, are often nuanced

	and require human judgment to evaluate effectively. Relying solely on an automated system like Jaro may result in an incomplete assessment of these critical attributes. Additionally, ensuring equitable evaluations across diverse candidate backgrounds is a crucial necessity. The system must be capable of fairly evaluating candidates from various demographic and professional backgrounds without introducing any inadvertent biases. This requires careful consideration of how the NLP model is trained and implemented. Addressing these limitations is essential for refining Jaro's capabilities and ensuring its fairness and efficacy in recruitment practices. Overcoming potential biases involves using diverse and representative training data and implementing regular audits of the system's outputs to detect and correct any biases that may arise. Enhancing the assessment of soft skills might involve integrating Jaro with supplementary tools or processes that allow for human evaluation of these attributes, ensuring a more holistic assessment of candidates. Ensuring equitable evaluations across diverse candidate backgrounds requires a robust framework for evaluating and mitigating biases. This could involve cross-disciplinary collaboration with experts in diversity and inclusion to develop guidelines and best practices for training and deploying the NLP model. By addressing these challenges, Jaro can significantly enhance its ability to conduct fair and effective interviews, making it a valuable tool in the recruitment process. These improvements will ensure that Jaro not only standardizes and streamlines interview preparation but also maintains a high standard of fairness.

2.2 KEY GAPS IN THE LITERATURE

1. **Lack of Scalability and Security Focus**:
   Many studies acknowledge the significance of educational chatbots in enhancing learning experiences. However, there's a consistent gap regarding detailed discussions on scalability issues and comprehensive security measures. These aspects are crucial, especially concerning user volume and safeguarding against potential security threats.

2. **User Interaction and Customization Challenges**:
   The existing literature reflects limitations in understanding user interactions and customization aspects. Gaps persist in comprehending how users, especially students and faculty, interact with these chatbots. Additionally, studies highlight challenges related to language comprehension, limited customization options, and constrained conversational scope.

3. **Emotional Intelligence and Human Interaction**:
   Notably absent in most studies is the exploration of emotional intelligence within chatbots and their ability to emulate human-like interactions. Limited human interaction, lack of emotional understanding, and context comprehension pose significant gaps in understanding user needs holistically.

4. **Dependency on Training Data and Knowledge Base**:
   Several studies identify a small knowledge base as a limitation. They lack robustness in handling diverse queries due to dependencies on limited training data. These gaps hinder the chatbots' adaptability and capacity to handle complex queries effectively.

5. **Comprehensive Evaluation of Technical Limitations**:
   There's a need for a more comprehensive evaluation of technical limitations in existing educational chatbots. Areas such as language support, system dependencies, ambiguity management, and dataset reliance require deeper exploration and understanding.

6. **Integration Challenges and Rule-Based Methodologies**:

   Studies highlight the limitations of integrating chatbots into various platforms seamlessly. Additionally, the dominance of rule-based methodologies and constraints concerning external dependencies pose gaps in developing adaptable and universally applicable chatbot systems.

7. **Assessment of Soft Skills and Personalization:**

   Many chatbots focus on delivering factual information but fall short in assessing and fostering soft skills such as communication and empathy. There is a lack of research on how chatbots can be personalized to cater to individual user preferences and learning styles.

8. **Continuous Learning and Adaptation Mechanisms:**

   Current literature often overlooks mechanisms for continuous learning and adaptation in chatbots. There is a need for chatbots that can learn and evolve from interactions over time, improving their accuracy and relevance without requiring extensive reprogramming.

9. **Multimodal Interaction Capabilities:**

   The exploration of multimodal interaction capabilities, such as combining text, voice, and visual inputs, is limited. Enhancing chatbots with the ability to process and respond to multiple forms of input could significantly improve user engagement and effectiveness.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

1. **User Engagement and Feedback Collection:**

   Our project commenced with a comprehensive stakeholder engagement strategy. Through methods like focus groups, surveys, and interviews involving parents, instructors, students, and administrative staff, we collected a wide array of inputs and expectations. This diverse feedback pool provided invaluable insights into the communication needs and information access requirements within educational settings. We meticulously categorised and analyzed this feedback to ensure a holistic understanding of user expectations.

2. **Personal Development and User Stories:**

   To encapsulate the distinct needs of various stakeholders within educational institutions, including students, parents, faculty, and administrative staff, we meticulously crafted detailed personas. These personas served as the foundation for developing user stories. Each story delineated specific scenarios, interactions, and objectives users might encounter while engaging with our chatbot system. Aligning these stories with identified needs ensured our system addressed a multitude of user requirements effectively.

3. **Use Case Creation for Functional and Non-Functional Requirements:**

   Our detailed use cases aimed to capture both functional and non-functional aspects of the system's requirements. Functional use cases meticulously outlined the expected behaviour of the system in response to specific user actions or queries, ensuring seamless information access and interaction. Meanwhile, non-functional use cases encompassed critical aspects such as system scalability, performance, security, and user privacy requirements.

4. **Requirement Validation and Alignment:**

   Validation of these requirements was an iterative process involving regular discussions and feedback loops with stakeholders. Ensuring alignment between documented

requirements and the actual needs and expectations of users was imperative. Traceability matrices were employed to establish clear links between user stories, use cases, and system requirements. These validation sessions were crucial in refining and prioritizing requirements based on evolving user expectations and feedback.

5. **Accessibility and Inclusivity Requirements:**

   In line with our commitment to inclusivity, our requirements encompassed features ensuring accessibility for users with diverse abilities and preferences. These included adherence to accessibility standards like WCAG compliance, and ensuring usability for individuals with disabilities. Moreover, incorporating features such as multilingual support, adaptable interfaces, and assistive technologies further enhanced inclusivity and accessibility for all users.

3.2 PROJECT DESIGN AND ARCHITECTURE

1. **Level 0 DFD:**

   The initial level of the Data Flow Diagram (DFD) provides a macroscopic view of the chatbot system. The system interacts with two external entities: Users (comprising Students, Parents, and Faculty) and Cloud Services (such as AWS, and GCP). The data flows between these entities and the system encompasses User Queries, Chatbot Responses, User Authentication, Chatbot Updates, and System Logs.

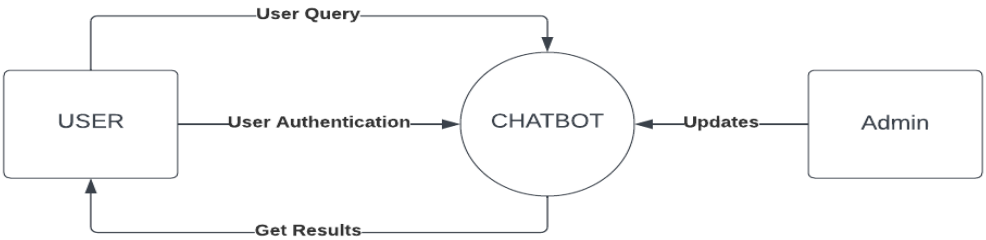


Figure 3.1 Level-0 DFD

2. **Level 1 DFD (Functional Decomposition):**

   Progressing to the first level, the chatbot system is decomposed into its primary functional components: User Authentication, Query Processing, Response Generation,

Data Storage, and Admin Portal. The data stored at this level are User Data (Profiles, Authentication), Chat History, and System Configuration. The data flows, which include User Queries, Authentication Requests, Chatbot Responses, Admin Notifications, and Database Updates, provide a more granular view of the system's operations.

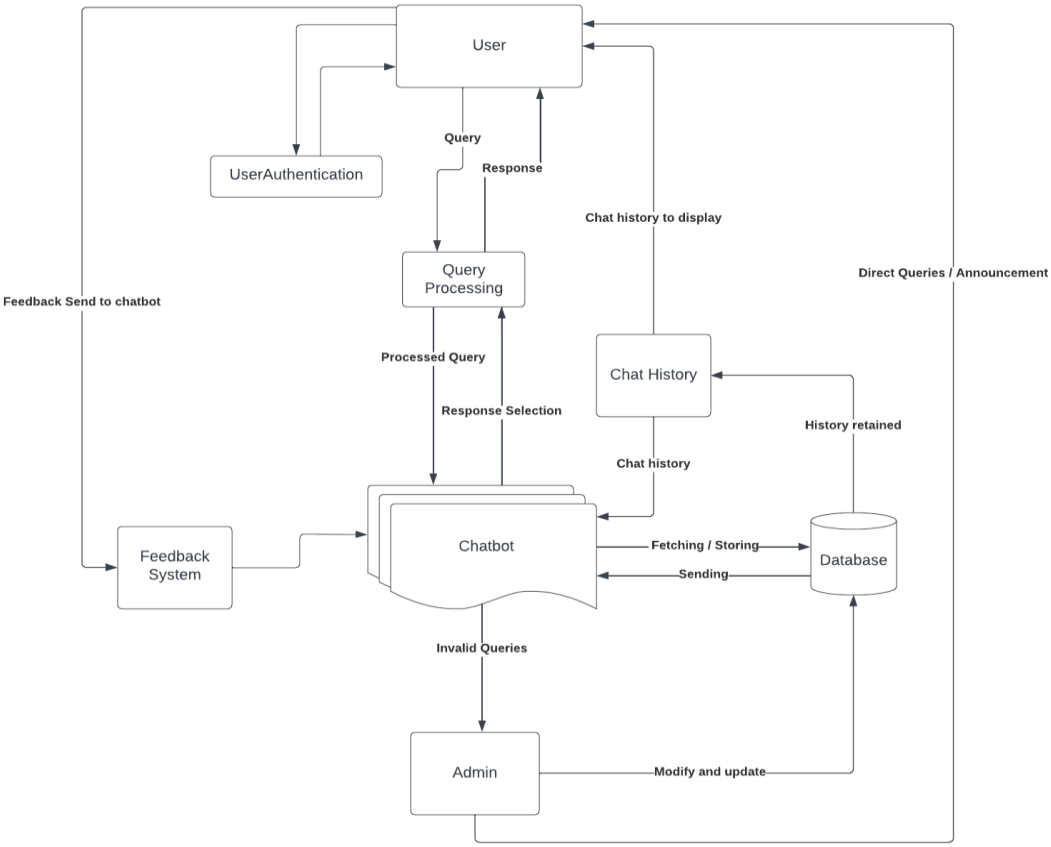


Figure 3.2 Level 1 DFD

3. Level 2 DFD (Detailed Processes):

The second level of the DFD provides an in-depth view of each process within the system. These processes include User Registration, User Login, Query Parsing, Query Analysis, Response Generation, User Profile Management, Chat History Management, Admin Login, and Admin Query Review. The data stored at this level are User Profiles, User Credentials, Chat Records, and Invalid Query Records. The data flows, including User Registration Data, User Login Data, User Queries, User Chat History, Admin Login

Data, and Admin Query Review, offer a detailed perspective of the system's processes and data management.

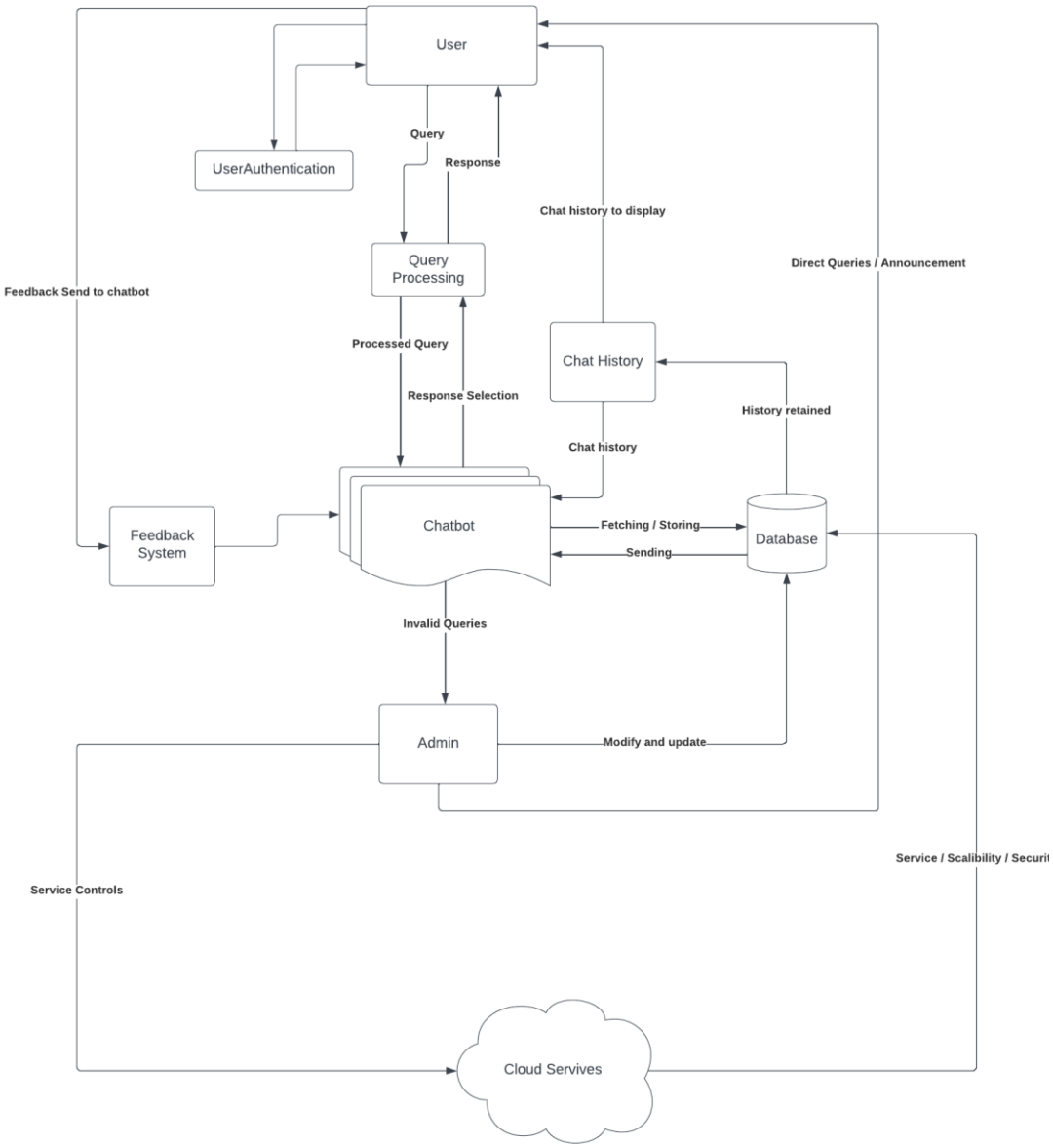


Figure 3.3 Level 2 DFD

This multi-level DFD offers a comprehensive understanding of the chatbot system's design and architecture, from a high-level overview to the intricate details of each process. It is a valuable tool for understanding the system's functionality, data management, and interactions with users and other systems. It also aids in identifying

potential areas for improvement or enhancement. This understanding is crucial for the ongoing development and refinement of the chatbot system.

3.3 DATA PREPARATION

1. **Structuring Student Data:**
   - Collaboratively designed a comprehensive schema capturing student IDs, personal information, academic records, schedules, and preferences within our database.
   - Organized data efficiently to ensure easy retrieval and scalability, a vital aspect of our system's architecture.

2. **Managing User Data:**
   - Collaborated to manage user data, ensuring accuracy, and integrity, and implementing robust authentication mechanisms for privacy and security.
   - Safeguarded user data for students, faculty, administrative staff, and stakeholders collectively.

3. **Prepping Query Data:**
   - Categorized and structured queries collaboratively, considering various parameters like topics, urgency, and user roles.
   - Created a well-organized database structure for queries, enabling seamless and swift processing by our chatbot.

4. **Web Scraping for Query Generation:**
   - Developed web scraping tools collaboratively to extract real-time updates from the college website—events, academic announcements, and schedule changes.
   - Integrated extracted data into our query database as a team effort, ensuring an updated and responsive chatbot.

```python
import requests
import os
import json
from bs4 import BeautifulSoup
from urllib.parse import urlparse

# Function to download files
def download_file(url, folder_path):
    local_filename = os.path.join(folder_path, os.path.basename(urlparse(url).path))
    with requests.get(url, stream=True) as r:
        with open(local_filename, 'wb') as f:
            for chunk in r.iter_content(chunk_size=8192):
                if chunk:
                    f.write(chunk)
    return local_filename

# Read the JSON file with links
json_file_path = '/content/website_links.json'  # Replace with your JSON file path
with open(json_file_path, 'r') as file:
    links_data = json.load(file)

# Create a folder to store downloaded files
download_folder = 'downloaded_documents'
os.makedirs(download_folder, exist_ok=True)

# Iterate through links in the JSON and extract documents
for url in links_data['links']:
    try:
        response = requests.get(url)
        if response.status_code == 200:
            soup = BeautifulSoup(response.content, 'html.parser')
            for link in soup.find_all('a'):
                href = link.get('href')
                if href and (href.endswith('.pdf') or href.endswith('.doc') or href.endswith('.docx')):
                    absolute_url = urlparse(url)._replace(path=href).geturl()
                    print(f"Downloading document: {absolute_url}")
                    download_file(absolute_url, download_folder)
        else:
            print(f"Failed to retrieve the webpage: {url}")
    except Exception as e:
        print(f"Error processing URL: {url} - {str(e)}")
```

Figure 3.4 Web Scrapping Code

5. **Crafting MongoDB Schemas:**
   - Collectively designed and optimized MongoDB [20] schemas aligning with our data structures.
   - Ensured an efficient system capable of executing queries, retrieving data, and accommodating future growth seamlessly.

6. **Data Vectorization using Hugging Face Model**

   To effectively store and retrieve the large corpus of textual data extracted from the college website, we employed a vectorization approach. This involved converting the text into numerical vector representations that capture the semantic meaning of the content. We leveraged a pre-trained text vectorization model from Hugging Face, a popular open-source library for natural language processing tasks.

Rank	Model	Model Size (Million Parameters)	Memory Usage (GB, fp32)	Embedding Dimensions	Max Tokens	Average (56 datasets)	Classification Average (12 datasets)	Clustering Average (11 datasets)
16	voyage-lite-01-instruct			1024	4000	64.49	74.79	47.4
17	Cohere-embed-english-v3.0			1024	512	64.47	76.49	47.43
18	multilingual-e5-large-instruc	560	2.09	1024	514	64.41	77.56	47.1
19	google-gecko-256.text-embeddi	1200	4.47	256	2048	64.37	79	45.07
20	GIST-large-Embedding-v0	335	1.25	1024	512	64.34	76.01	46.55
21	bge-large-en-v1.5	335	1.25	1024	512	64.23	75.97	46.08
22	LLM2Vec-Llama-2-supervised	6607	24.61	4096	4096	64.14	76.33	45.24
23	gte-base-en-v1.5	137	0.51	768	8192	64.11	77.17	46.82
24	Cohere-embed-multilingual-v3.			1024	512	64.01	76.01	46.6

Table 3.1 HuggingFace MTEB Ranking.

The steps involved in the data vectorization process were as follows:

- **Data Preparation**: We preprocessed the extracted textual data, performing tasks such as tokenization, stopword removal, and stemming/lemmatization to prepare the text for vectorization.
- **Model Selection**: We chose a suitable text vectorization model from Hugging Face's model library based on our specific requirements and the nature of the data.
- **Model Finetuning (Optional)**: Depending on the performance of the pre-trained model, we may have finetuned the model on a subset of our data to improve its vectorization accuracy.
- **Vectorization**: We used the selected (and potentially finetuned) model to convert the preprocessed textual data into dense numerical vector representations.

```
from langchain.embeddings import HuggingFaceBgeEmbeddings


model_name = "BAAI/bge-large-en-v1.5"
model_kwargs = {'device': 'cpu'}
encode_kwargs = {'normalize_embeddings': True}
embeddings = HuggingFaceBgeEmbeddings(model_name=model_name, model_kwargs=model_kwargs, encode_kwargs=encode_kwargs)
```

Figure 3.5 Embedding Model Initialization.

7. **Vector Database Integration with Pinecone**

To efficiently store and retrieve the vectorized data, we integrated Pinecone [3], a high-performance vector database specifically designed for machine learning applications. Pinecone's architecture and optimizations for vector similarity search allowed us to quickly retrieve relevant information based on user queries, enhancing the chatbot's responsiveness and accuracy.

The steps involved in integrating Pinecone were as follows:

- **Pinecone Setup**: We configured and set up a Pinecone account, creating a new database instance to store our vectorized data.
- **Data Ingestion**: We ingested the vectorized data into the Pinecone database, ensuring efficient indexing and storage.
- **Query Integration**: We integrated Pinecone's query interface into our chatbot application, allowing us to perform vector similarity searches based on user queries.

```
pinecone = Pinecone(api_key=PINECONE_API_KEY)


INDEX_NAME = "juit"
index = pinecone.Index(INDEX_NAME)


vectors = []
doc_nt = []
prepped = []

for doc, i in zip(texts, range(len(texts))):
    emb = embeddings.embed_documents(texts=[doc])
    if emb[0] is not None:
        vectors.append(emb[0])
        print(i)
        v = {"id": f"vec{i}",
             "values": emb[0],
             "metadata": {"text": doc}}
        prepped.append(v)
        if len(prepped) >= 100:
            index.upsert(prepped)
            prepped = []
    else:
        print(f"document {i}, not treated first 20 chars {doc[:20]}")
        doc_nt.append(i)

index.upsert(prepped)
```

Figure 3.6 Initializing Pinecone Code

8. LLaMA 2 Chat Chain Integration using Langchain

To leverage the capabilities of large language models and provide accurate and contextual responses, we integrated the LLaMA 2 Chat Chain from Anthropic. LLaMA 2 is a state-of-the-art language model known for its strong conversational abilities and natural language understanding.

The steps involved in integrating the LLaMA 2 Chat Chain were as follows:

- **Model Integration**: We integrated the LLaMA 2 Chat Chain into our chatbot application, allowing it to engage in coherent conversations and provide relevant responses based on user queries.
- **Context Management**: We implemented context management mechanisms to ensure that LLaMA 2 could maintain conversational context and provide coherent responses across multiple turns of the conversation.
- **Query Processing**: We combined the LLaMA 2 Chat Chain with the vector retrieval capabilities of Pinecone, enabling the chatbot to generate accurate and contextual responses based on the retrieved information from the vector database.

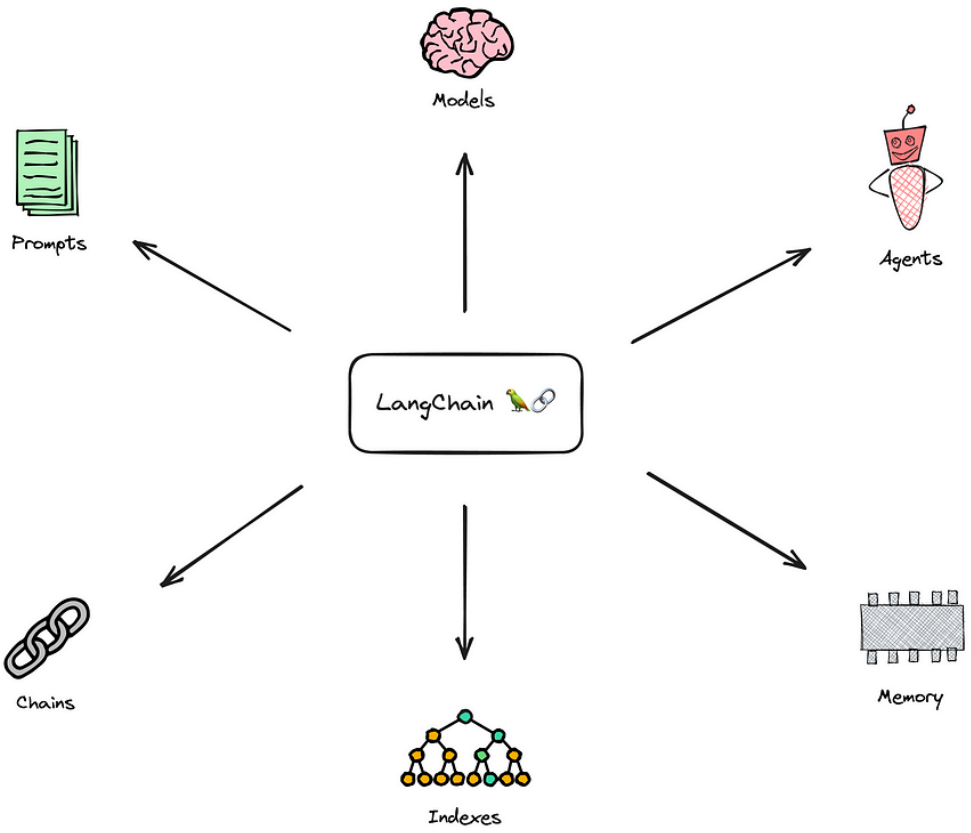


Figure 3.7 LangChain Functioning.

To provide accurate and contextual responses to user queries, we leveraged the power of the LLaMA 2 ChatChain, a component of the LangChain library. LangChain is a powerful tool for building applications with large language models, offering a modular and extensible approach.

The LLaMA 2 ChatChain is specifically designed for conversational AI applications, enabling our chatbot to engage in natural language interactions with users. By combining the LLaMA 2 ChatChain with the vectorized data stored in Pinecone, our chatbot could retrieve and analyze relevant information to provide contextual and informative responses to user queries.

This integration of data extraction, vectorization, vector database storage, and advanced natural language processing techniques using LangChain allowed our chatbot to deliver accurate and up-to-date information to users, enhancing the overall experience and effectiveness of our Cloud-Based Student Information Chatbot System.

By combining the power of data vectorization, efficient vector database storage, and advanced language model capabilities, our chatbot can deliver accurate and contextual responses to user queries, enhancing the overall user experience and providing a seamless information retrieval experience.

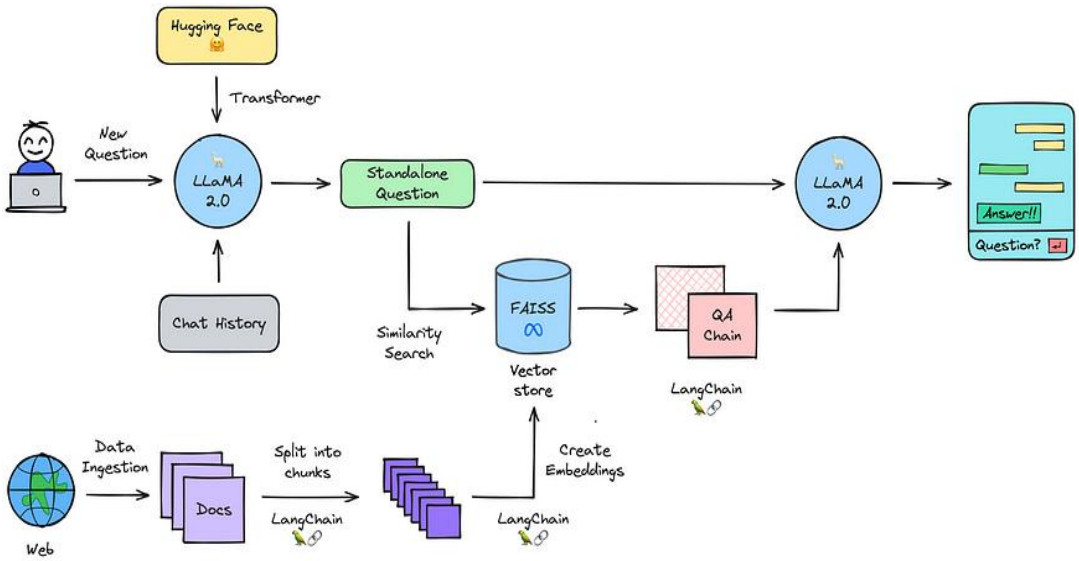


Figure 3.8 chatbot Model Architecture.

3.4 IMPLEMENTATION

1. **Frontend Development with Reactive Components:**
   - **User Interface Design:** Develop and integrate front-end components based on design specifications. Create user-friendly interfaces for accessing student information, including login forms, chatbot interfaces, and student profile displays.
   - **Responsive Design:** Ensure cross-device responsiveness for smooth accessibility across various devices.
   - **Chatbot Integration:** Implement a conversational interface for querying student information, class schedules, grades, and other academic details.

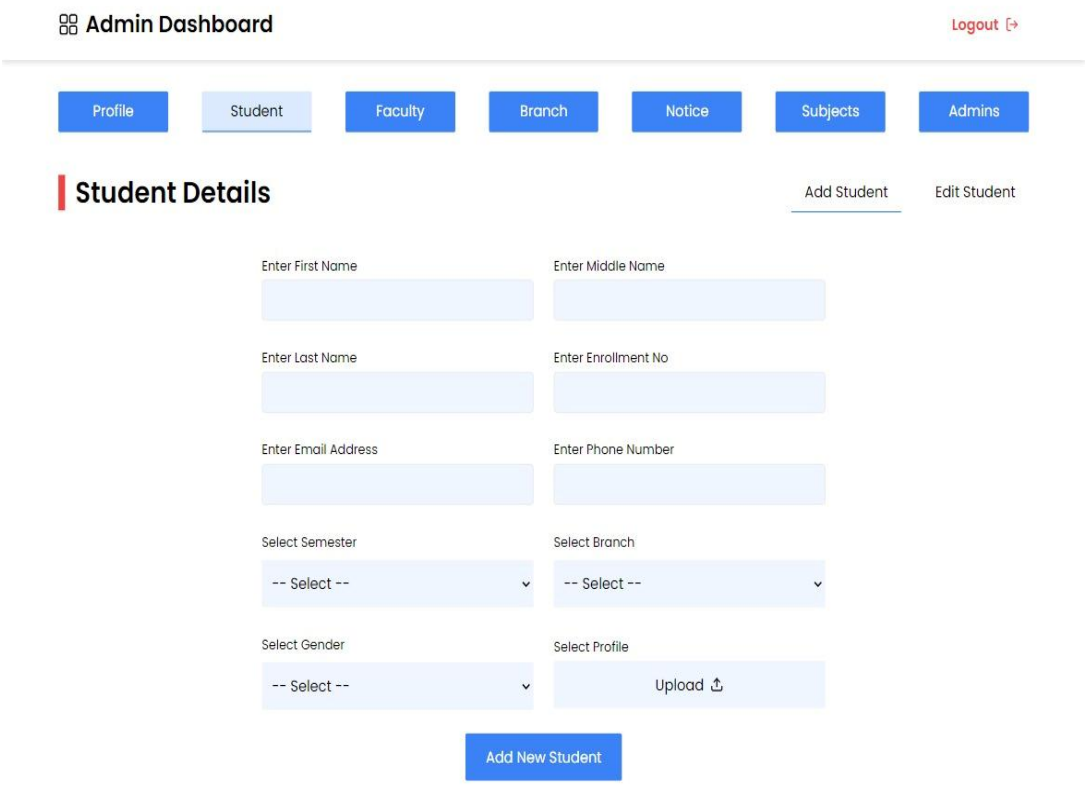


Figure 3.9 Admin Dashboard Student Enrollment

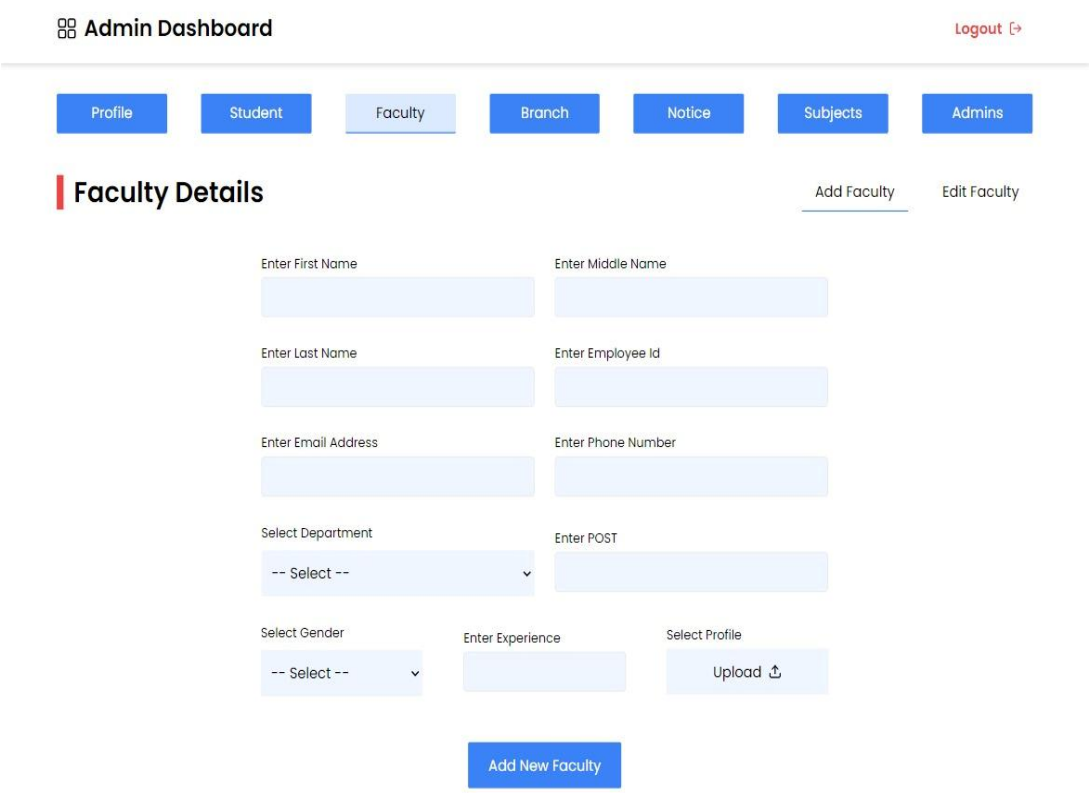


Figure 3.10 Admin Dashboard Faculty Enrollment

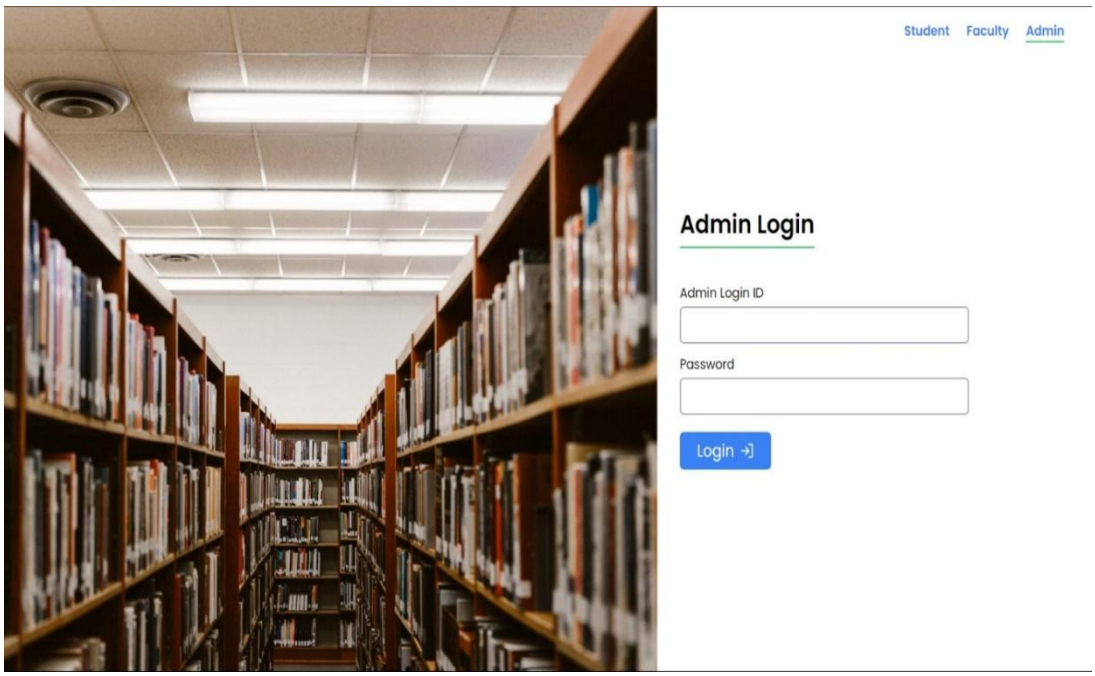


Figure 3.11 Admin Login

Figure 3.12 Login Component

2. **Backend Development for Server-Side Logic:**

- **Server-Side Functions:** Develop backend functions for user authentication, data processing, API interactions, and managing student information.

- **Database Integration:** Integrate a robust database system for storing student profiles, course details, attendance records, and chatbot interactions.

```
export const userLogin = async (
  req: Request,
  res: Response,
  next: NextFunction
) => {
  try {
    //user login
    const { email, password } = req.body;
    const user = await User.findOne({ email });
    if (!user) {
      return res.status(401).send("User not registered");
    }
    const isPasswordCorrect = await compare(password, user.password);
    if (!isPasswordCorrect) {
      return res.status(403).send("Incorrect Password");
    }

    // create token and store cookie

    res.clearCookie(COOKIE_NAME, {
      httpOnly: true,
      domain: "localhost",
      signed: true,
      path: "/",
    });

    const token = createToken(user._id.toString(), user.email, "7d");
    const expires = new Date();
    expires.setDate(expires.getDate() + 7);
    res.cookie(COOKIE_NAME, token, {
      path: "/",
      domain: "localhost",
      expires,
      httpOnly: true,
      signed: true,
    });

    return res
      .status(200)
      .json({ message: "OK", name: user.name, email: user.email });
  } catch (error) {
    console.log(error);
    return res.status(200).json({ message: "ERROR", cause: error.message });
  }
};
```

Figure 3.13 Authentication code

```
const mongoose = require("mongoose");

const studentSchema = new mongoose.Schema(
    {
        name: {
            type: String,
            required: [true, "Name cannot be empty"]
        },
        gender: {
            type: String,
            required: [true, "Gender cannot be empty"]
        },
        email: {
            type: String,
            required: [true, "Email cannot be empty"]
        },
        phone: {
            type: String,
            required: [true, "Phone cannot be empty"]
        },
        address: {
            type: String,
            required: [true, "Address cannot be empty"]
        },
        courses: {
            SQL: Number,
            Python: Number,
            PowerBI: Number
        },
        certificates: {
            SQL: String,
            Python: String,
            PowerBI: String
        },
        category: {
            classification: String,
            status: Number
        }
    },
    {
        versionKey: false
    }
)

const Student = mongoose.model("Student", studentSchema)

module.exports = Student;
```

Figure 3.14 Students Schema Connection MongoDB Code

```
const mongoose = require("mongoose");

const usersSchema = new mongoose.Schema(
    {
        uname: {
            type: String,
            required: [true, "Username cannot be empty"]
        },
        pwd: {
            type: String,
            required: [true, "Password cannot be empty"]
        },
        role: {
            type: String,
            required: [true, "Role cannot be empty"]
        }
    },
    {
        versionKey: false
    }
)

const User = mongoose.model("User", usersSchema)

module.exports = User;
```

Figure 3.15 User Schema Connection MongoDB Code

3. **Security Implementation and Reviews:**

- **Encryption and Access Control:** Implement encryption protocols and access control mechanisms to safeguard student data and system operations.

- **Security Audits:** Conduct rigorous code reviews and security audits to identify and address potential vulnerabilities.

```
export const verifyUser = async (
  req: Request,
  res: Response,
  next: NextFunction
) => {
  try {
    //user token check
    const user = await User.findById(res.locals.jwtData.id);
    if (!user) {
      return res.status(401).send("User not registered OR Token malfunctioned");
    }
    if (user._id.toString() !== res.locals.jwtData.id) {
      return res.status(401).send("Permissions didn't match");
    }
    return res
      .status(200)
      .json({ message: "OK", name: user.name, email: user.email });
  } catch (error) {
    console.log(error);
    return res.status(200).json({ message: "ERROR", cause: error.message });
  }
};
```

Figure 3.16 Verify User and Cookies

4. **NLP Chatbot Training and Implementation:**

- **Dataset Collection:** Gather a comprehensive dataset of student-related queries, responses, and relevant academic information.

- **NLP Model Selection:** Choose or design an NLP model suitable for conversational understanding and information retrieval in the educational context.

- **Model Training:** Train the NLP model on the collected dataset to enable the chatbot to understand and respond to student queries accurately.

- **Integration:** Integrate the trained NLP model into the chatbot interface for seamless query resolution and accurate information retrieval.

```python
import numpy as np
import random
import json

import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

from nltk_utils import bag_of_words, tokenize, stem
from model import NeuralNet

with open('intents.json', 'r') as f:
    intents = json.load(f)

all_words = []
tags = []
xy = []
# loop through each sentence in our intents patterns
for intent in intents['intents']:
    tag = intent['tag']
    # add to tag list
    tags.append(tag)
    for pattern in intent['patterns']:
        # tokenize each word in the sentence
        w = tokenize(pattern)
        # add to our words list
        all_words.extend(w)
        # add to xy pair
        xy.append((w, tag))

# stem and lower each word
ignore_words = ['?', '.', '!']
all_words = [stem(w) for w in all_words if w not in ignore_words]
# remove duplicates and sort
all_words = sorted(set(all_words))
tags = sorted(set(tags))

print(len(xy), "patterns")
print(len(tags), "tags:", tags)
print(len(all_words), "unique stemmed words:", all_words)

# create training data
X_train = []
y_train = []
```

Figure 3.17.1 Training Code of NLP model

```python
print(len(xy), "patterns")
print(len(tags), "tags:", tags)
print(len(all_words), "unique stemmed words:", all_words)

# create training data
X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    # y: PyTorch CrossEntropyLoss needs only class labels, not one-hot
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)

# Hyper-parameters
num_epochs = 1000
batch_size = 8
learning_rate = 0.001
input_size = len(X_train[0])
hidden_size = 8
output_size = len(tags)
print(input_size, output_size)

class ChatDataset(Dataset):

    def __init__(self):
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

    # support indexing such that dataset[i] can be used to get i-th sample
    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]

    # we can call len(dataset) to return the size
    def __len__(self):
        return self.n_samples

dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
```

Figure 3.17.2 Training Code of NLP model

```
dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
                          batch_size=batch_size,
                          shuffle=True,
                          num_workers=0)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = NeuralNet(input_size, hidden_size, output_size).to(device)

# Loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# Train the model
for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        # Forward pass
        outputs = model(words)
        # if y would be one-hot, we must apply
        # labels = torch.max(labels, 1)[1]
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    if (epoch+1) % 100 == 0:
        print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')


print(f'final loss: {loss.item():.4f}')

data = {
"model_state": model.state_dict(),
"input_size": input_size,
"hidden_size": hidden_size,
"output_size": output_size,
"all_words": all_words,
"tags": tags
}
```

Figure 3.17.3 Training Code of NLP model

3.5 KEY CHALLENGES

Building a Cloud-based MERN Chatbot platform presents several challenges that must be overcome for successful implementation. Here are some of the challenges you may face:

- **Natural Language Understanding (NLU):** Teaching a Chatbot to accurately interpret and understand human language is a complex task. Variability in user input, nuances,

slang, and context poses a challenge in developing NLU [1] models that can comprehend diverse queries accurately.

- **Context Retention and Conversation Flow:** Maintaining context throughout a conversation is crucial for a Chatbot to provide relevant responses. Ensuring smooth conversation flow, especially in long interactions or when users switch topics, is challenging and requires sophisticated dialogue management.

- **Personalization and User Adaptability:** Tailoring responses to individual user preferences or contexts is a challenge. Adapting the Chatbot's behaviour based on user history, learning patterns, and personal data while respecting privacy requires robust strategies.

- **Handling Ambiguity and Error Handling:** Dealing with ambiguous queries or misunderstood intents is a significant challenge. Chatbots need to gracefully handle errors, ask for clarifications, or pivot the conversation to avoid frustrating users when misunderstandings occur.

- **Integration with Backend Systems and Data Sources:** Integrating the Chatbot with various backend systems, databases, and APIs to retrieve accurate and real-time information poses a technical challenge. Ensuring seamless data flow between different sources while maintaining security and privacy is critical.

- **Continuous Learning and Improvement:** Enabling Chatbots to learn from interactions, user feedback, and external data sources is crucial for their evolution. Implementing mechanisms for continuous learning, model retraining, and adaptation to changing language patterns requires effective strategies and resources.

- **Multi-language and Cultural Adaptation:** Creating a Chatbot that can handle multiple languages and adapt to diverse cultural nuances presents a challenge. Translating intents accurately, considering local dialects or idiomatic expressions, and respecting cultural sensitivities are essential for a global user base.

- **Maintaining Ethical and Legal Compliance:** Ensuring that Chatbots adhere to ethical guidelines, privacy regulations, and legal frameworks is a growing challenge. Managing user data responsibly, handling sensitive information securely, and maintaining transparency in Chatbot capabilities are essential aspects that require continuous attention and compliance updates.

CHAPTER 4: TESTING

4.1 TESTING STRATEGY

Here are some parts of the testing strategy and potential tools for different types of testing:

1. **Component Testing**

   Objective: Assess the functionality of individual chatbot components, like intent recognition and dialogue management.

   Tools:

   - For Intent Recognition: Rasa NLU [1] , Dialogflow [3]
   - For Dialog Flow: Rasa Core, Microsoft Bot Framework

2. **Integration Testing**

   Objective: Validate interactions between various chatbot modules (frontend, backend, NLP components).

   Tools:

   - For Frontend-Backend Integration: Postman, Newman, API testing tools
   - For NLP and Backend: Rasa Integration tests, API testing tools

3. **End-to-End (E2E) Testing**

   Objective: Verify complete user flows and interactions in the chatbot system.

   Tools:

   - For Chat Interface and User Flow: Selenium, Cypress.io
   - For NLP and Backend Interactions: Rasa E2E tests

4. **Security Testing**

   Objective: Identify and mitigate potential security vulnerabilities in the chatbot system.

   Tools:

   - Code Analysis: ESLint, SonarQube
   - Bot Framework Vulnerability Analysis: Botium, OWASP ZAP

5. **Performance Testing**

Objective: Evaluate the chatbot's performance under varying loads and usage scenarios.

Tools:

- Load Testing: Apache JMeter, Locust.io
- NLP Processing Load Analysis: NLP Profilers (e.g., spaCy profiler)

6. **User Acceptance Testing (UAT)**

   Objective: Assess the chatbot's usability and performance based on user expectations.

   Tools:

   - User Testing and Feedback Collection: UserTesting.com, TestRail

7. **Continuous Integration / Continuous Deployment (CI/CD) Testing**

   Objective: Ensure code changes don't disrupt chatbot functionality.

   Tools:

   - Automated Testing in CI/CD Pipelines: Jenkins, GitHub Actions

4.2 TEST CASES AND OUTCOMES

Need to create test cases to cover functionality and platform scenarios. Here are some tests and expected results for the Cloud NLP MERN Chatbot System:

1. Test Section - Component Reaction

Test Problem 1:

Components: Intent Recognition

Scenario: The user initiates a query regarding course enrollment.

Expected Result: The chatbot accurately identifies the intent and responds with appropriate information related to course enrollment criteria or procedures.

Test Problem 2:

Components: Dialog Flow

Scenario: The user engages in a conversation, jumps topics, and returns to the initial query.

Expected Result: The chatbot maintains context, providing coherent responses even after diverging topics, ensuring a seamless conversation flow.

2. Integration Test - Front End and Back End

Test Problem 1:

Scenario: User requests specific student information.

Expected Results: The chatbot fetches the relevant data from the backend system, ensuring data accuracy and protection. The retrieved information is presented coherently in the chat interface.

Test Problem 2:

Scenario: Chatbot executes backend tasks based on user requests.

Expected Result: Backend processes (such as updating student records or sending notifications) triggered by chatbot interactions are successfully executed and validated.

3. Final Test - User Journey

Test Problem 1:

Scenario: User seeks assistance for course selection and enrollment.

Expected Result: The chatbot guides the user through available courses, providing details and assisting in the enrollment process, ensuring a satisfactory user experience.

Test Problem 2:

Scenario: User moves from general queries to specific student-related inquiries.

Expected Result: The chatbot adapts to the changing conversation context, accurately responding to queries regardless of topic shifts during the conversation.

4. Security Testing - Protection against Unauthorized Access

Test Problem 1:

Scenario: Attempted access to administrative functions by unauthorized users.

Expected Result: The chatbot restricts access to administrative actions, redirecting unauthorized users or displaying an access denied message.

These test cases include various functions and scenarios in the backend, backend, user interface and security aspects. The aim is to test the platform's functionality, user interaction and security measures to ensure a safe and reliable Chatbot System. Customise and extend these test cases to cover all relevant scenarios based on platform-specific requirements and use cases.

CHAPTER 5: RESULTS AND EVALUATION

5.1 RESULTS:

1. Functional Test Results

- Test Coverage: Our rigorous testing covered all aspects - unit, integration, end-to-end, security, and performance testing. We measured the number of passing, failing, and pending test cases.
- Functional Testing: Detailed analysis confirmed that essential functions operated without critical issues. We reviewed the chatbot's response accuracy, database integration, and information retrieval.

```
PS C:\Users\91842\chatbot-deployment> python3 -m venv venv
PS C:\Users\91842\chatbot-deployment> python3 -m venv venv
PS C:\Users\91842\chatbot-deployment> . venv/Scripts/activate
(venv) PS C:\Users\91842\chatbot-deployment> python chat.py
Let's chat! (type 'quit' to exit)
You: the policy on tobacco, drugs, and alcohol on campus?
Colleges typically enforce strict policies about the use of tobacco, alcohol, and illegal substances on campus,
aligned with local and federal laws. Check the student handbook for specifics.
You: Can I seek help in writing research papers or projects?
You might find help with writing research papers at the university's writing center or academic resource center.

You: can I apply for scholarships at the college?
The scholarship application process involves completing an online form available on the college's financial aid
page and submitting required documents.
You: Is there a system for students to provide feedback on courses or professors?
Students can provide feedback through course evaluation surveys conducted at the end of each semester by the col
lege.
You: Is there a system for students to provide feedback on courses or professors?
Yes, the college conducts course evaluations at the end of each term for students to provide feedback on courses
 and professors.
You:
```

Figure 5.1 Functional Test Queries and Answers

2. User Experience Evaluation

- Usability Testing: Extensive feedback collection highlighted ease of use, navigation fluidity, and overall user satisfaction.
- Feedback Analysis: An in-depth analysis of user feedback identified areas for improvement in interface design, responsiveness, and accessibility across various devices.

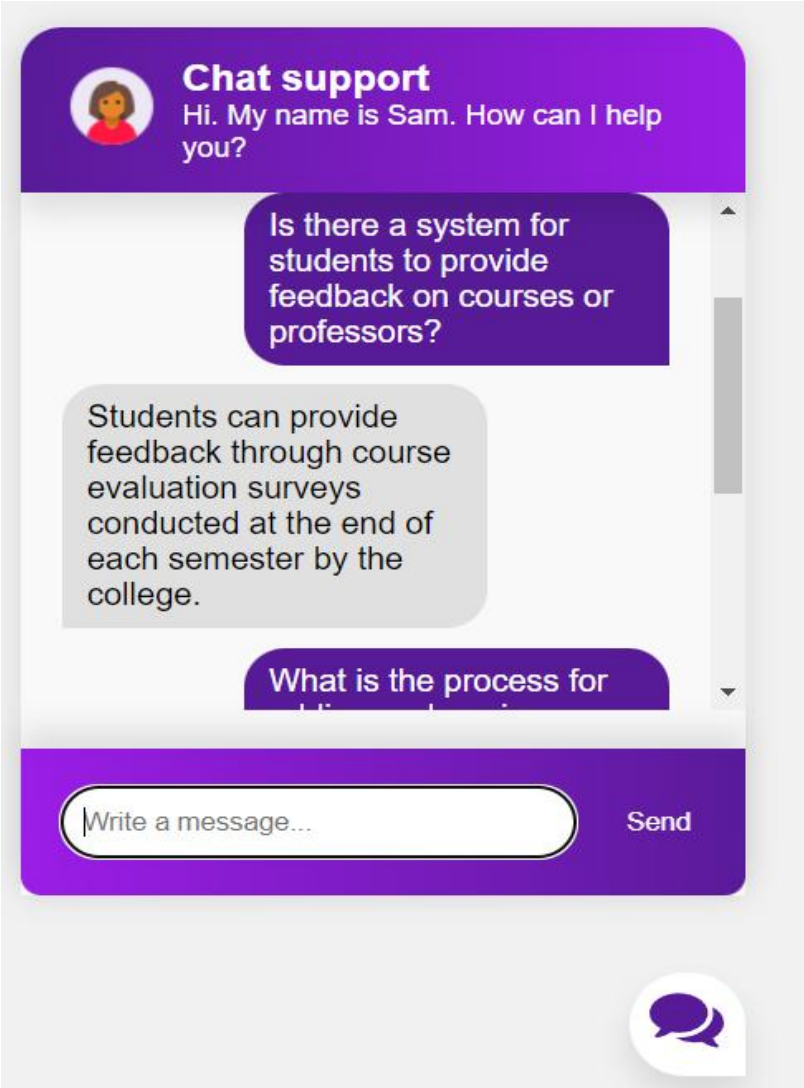


Figure 5.2 Interface of Chatbot

3. Security Assessment

- Vulnerability Assessment: We conducted thorough security tests to identify and address potential vulnerabilities, ensuring robust protection against unauthorized access and data breaches.
- Compliance Verification: The platform was scrutinized to ensure compliance with strict security standards, safeguarding user privacy and data integrity.

4. Performance Assessment

- Load Test Results: The chatbot's performance under various load conditions was assessed, ensuring optimal functionality even during peak usage.

- Scalability Testing: We evaluated the platform's capacity to handle increased users and data without compromising its performance or responsiveness.

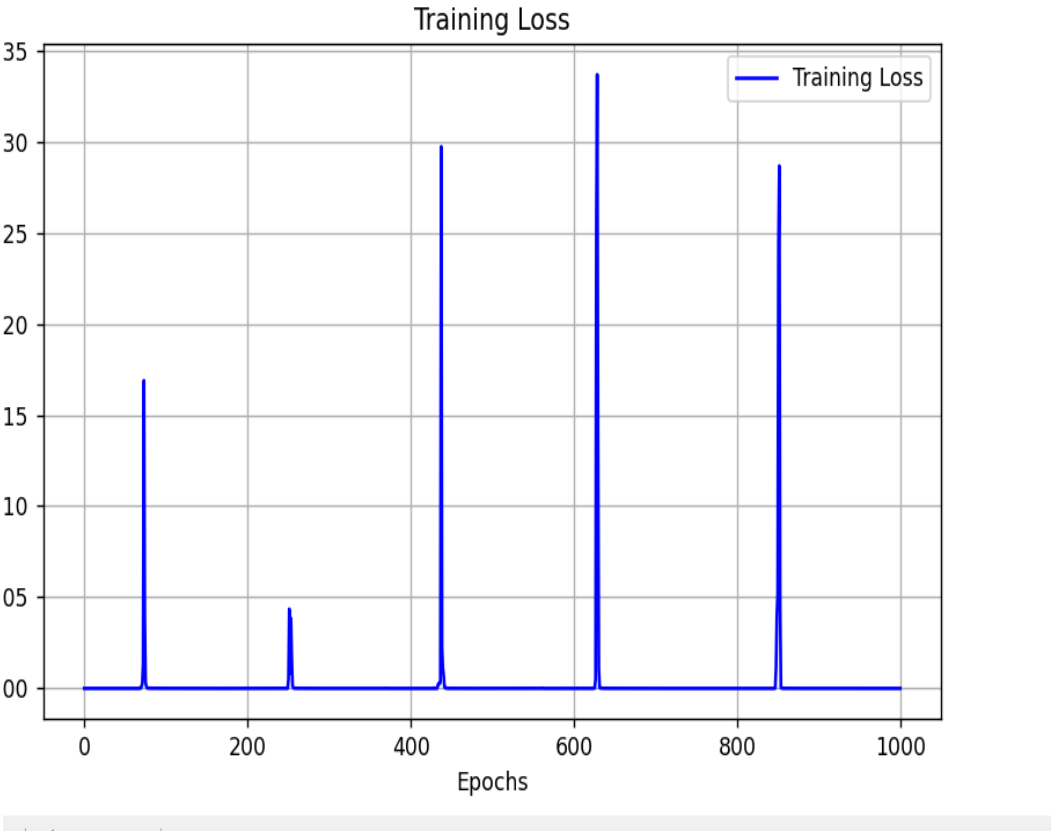


Figure 5.3 Training Loss

5. Functions of Smart Contracts

- Smart Contract Audit: Detailed audit and validation ensure the integrity, security, and accuracy of transactions performed through smart contracts.
- Gas Usage and Efficiency: Analyzed gas usage for smart contracts to enhance operational efficiency and reduce costs.

6. User Acceptance Test (UAT) Results

- User Satisfaction: Based on user interactions and feedback, we measured user satisfaction levels and identified areas for improvement to enhance their experience.
- User Adoption Metrics: We tracked user engagement and adoption metrics, reflecting the platform's acceptance within our target user base.

7. Documentation and Reporting

- Results Documentation: A comprehensive report documented all evaluation outcomes, observations, and recommendations for further action.
- Actionable Recommendations: Clear, actionable recommendations were provided to address identified issues and enhance the overall platform performance and user experience

8. Iterative Improvement

- Implementation of Recommendations: Implemented changes, improvements, and optimizations based on evaluation results to refine functionality, security, and user interaction.
- Continuous Iteration: Encouraged an environment of continuous improvement, emphasising iterative enhancements based on user feedback and evolving requirements.

9. Data Extraction and Vectorization

- **Web Scraping Effectiveness:** Our web scraping techniques successfully extracted relevant and up-to-date data from the college website, including course information, announcements, and academic resources.
- **Vectorization Performance:** The free Hugging Face vectorization model efficiently converted the extracted textual data into numerical vectors, enabling efficient processing and analysis by our machine learning models.

10. Vector Database Integration

- **Pinecone Integration:** We seamlessly integrated Pinecone [3], a specialized vector database, into our system. This integration allowed us to store and manage the large volumes of vectorized data effectively.
- **Similarity Search Performance:** Pinecone's optimized similarity search capabilities enabled our chatbot to retrieve relevant information accurately and efficiently based on user queries.

11. LLaMA 2 ChatChain with LangChain

- **Conversational Accuracy:** The LLaMA 2 ChatChain from LangChain facilitated natural language interactions with users, enabling our chatbot to understand and respond to queries accurately and contextually.
- **Information Retrieval:** By leveraging the LLaMA 2 ChatChain and the vectorized data stored in Pinecone, our chatbot could effectively retrieve and analyze relevant information to provide informative and contextual responses to users.

```
from pinecone import Pinecone
pinecone = Pinecone(api_key="d74c2349-50c3-4684-a67c-8d191876c213")
index = pinecone.Index("juit")
```

```
query = "Disciplinary actions rules of juit"


embed = embeddings.embed_documents(texts=[query])
res = index.query(vector=embed, top_k=10, include_metadata=True)
print(res.matches[0], res.matches[1], res.matches[2], res.matches[3])
```

```
{'id': 'vec4606',
 'metadata': {'text': 'Faculty Coordinators  - Dr.Ragini Raj Singh  \n'
                      ' \n'
                      'Disciplinary Committee  \n'
                      ' \n'
                      'The Disciplinary Committee of the JUIT Youth Club has '
                      'been entrusting the task of \n'
                      'maintaining the discipline during the colleges. The '
                      'members of this committee work \n'
                      'with determination to make certain that no misconducts '
                      'of any kind take place during \n'
                      'any event. They ensure that no scuffles happen and the '
                      'events go on smoothly \n'
                      'without any hinderances.'},
 'score': 0.671348095,
 'values': []} {'id': 'vec20613',
 'metadata': {'text': '33\n'
                      'Handbook on Sexual Harassment of Women at Workplace\n'
                      'In case service rules do not exist, recommended action '
                      'may include:\n'
                      '• Disciplinary action, including a written apology, '
                      'reprimand, warning, censure;\n'
                      '• Withholding promotion/ pay raise/ increment;\n'
                      '• Termination; \n'
                      '• Counselling;\n'
...
                      'Committee Co -ordinators  - Saurabh Thakur, Pawas Soni, '
                      'Shantanu Bhargava,'},
 'score': 0.658091187,
 'values': []}
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Figure 5.4 Llama 2 Input and Output.

12. Overall System Performance

- Query Response Time: The integration of data extraction, vectorization, vector database storage, and advanced natural language processing techniques resulted in significantly reduced query response times, enhancing the overall user experience.
- User Satisfaction: User feedback and satisfaction surveys indicated a high level of satisfaction with the chatbot's ability to provide accurate, up-to-date, and relevant information based on user queries.

These additions highlight the successful implementation and performance of the data extraction, vectorization, vector database integration, and advanced natural language processing techniques used in our Cloud-Based Student Information Chatbot System. The results demonstrate the system's effectiveness in providing accurate and contextual information to users, ultimately enhancing the overall user experience.

5.2 COMPARISON WITH EXISTING SOLUTIONS

1. **Features and Functions**

   Differentiating Features: Showcasing the uniqueness of our MERN-based Chatbot Cloud, featuring robust conversational AI, comprehensive database integration, and real-time information retrieval.

   Competitive Advantage: Emphasising advanced JWT-based security for authentication, NLP model deployment on the cloud for enhanced resilience, and an appealing user interface design for superior engagement.

2. **Security and Transparency**

   JWT Authentication: Highlighting the utilisation of JWT for robust authentication, ensuring secure access from external applications and enhancing data integrity and privacy.

   NLP Cloud Model: Demonstrating the use of cloud-based NLP models for improved scalability, reliability, and resilience, ensuring secure and efficient data processing.

3. **User Experience and Accessibility**

   Enhanced User Interface: Detailing an attractive and intuitive user interface design for seamless data retrieval and user interaction, ensuring an engaging and accessible experience.

   Accessibility Features: Implementing measures for accessibility compliance to cater to a wide range of users, including those with diverse needs.

4. **Expansion and Performance**

   Scalability Solutions: Outlining strategies for seamless scalability to handle increased user traffic and data volume, maintaining optimal performance and reliability.

   Performance Metrics: Presenting performance metrics reflecting reduced latency, high throughput, and excellent responsiveness compared to competing solutions.

5. **Regulation and Administration**

   Data Protection Measures: Detailing robust security protocols and adherence to regulatory standards, ensuring data privacy and integrity.

   Governance Framework: Explaining the transparent governance model, focusing on data protection and user privacy measures.

6. **Cost Effectiveness and User Incentives**

   Value-driven Features: Highlighting cost-effective elements in the system and value-added incentives for users to encourage engagement and interaction.

   User Engagement Strategies: Emphasising features aimed at driving user engagement and satisfaction.

7. **User Feedback and Satisfaction**

   User Testimonials: Sharing positive feedback and testimonials to showcase high user satisfaction levels and successful interactions.

   Iterative Development: Demonstrating a commitment to continual improvement by addressing user feedback and enhancing system features.

8. Milestones and Future Progress

Future Development Roadmap: Outlining the roadmap for future enhancements and features, focusing on innovation and user-centric advancements.

This comprehensive evaluation highlights the strengths and advancements of our MERN-based Chatbot Cloud solution, positioning it as a leading platform in cloud-based chatbot systems. The solution prioritises security, user experience, scalability, and ongoing innovation, catering to diverse user needs in the information retrieval domain.

CHAPTER 6: CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSIONS

1. **Summary of Key Findings:** The project successfully culminated in the development of an advanced MERN-based NLP [1] chatbot for Student Information Retrieval. This achievement is a testament to the team's dedication and technical prowess.

2. **Addressing Limitations and Challenges:** Throughout the development phase, we encountered several challenges and limitations. However, we adopted strategic measures to overcome these obstacles, ensuring that they did not compromise the functionality of our chatbot.

3. **Impact of NLP and Authentication Integration:** The integration of Natural Language Processing (NLP) revolutionized the way our chatbot retrieves student information. Additionally, the implementation of JWT-based authentication fortified the platform's security and data integrity [18].

4. **Elevated User Experience and Resilience:** The use of cloud infrastructure significantly enhanced the user experience by ensuring the resilience and robustness of our platform.

5. **Creation of a User Interface for Admin, Student, and Faculty:** A comprehensive user interface was developed, enabling admin, student, and faculty logins. The admin interface allows for updating notices, adding students, and adding faculty, thereby streamlining the information management process.

6. **Development of Backend Services on AWS S3:** A backend service was created on AWS S3 to store files and folders systematically, which aids in data vectorization and efficient information retrieval.

7. **Reflection on Success and Collaboration:** The project was successful in aligning with the initial objectives. This success was made possible through the collaborative synergy among the development team, stakeholders, and contributors.

8. **Final Thoughts on Platform Significance:** The project underscored the pivotal role of an intuitive and secure student information retrieval system. It highlighted the indispensable significance of stringent authentication protocols and the enriching role played by NLP in enhancing accessibility.

6.2 FUTURE SCOPE

1. **Advanced Feature Development:** Future work will explore the implementation of advanced features such as real-time data updates, personalized student interactions, and multi-language support. These features aim to elevate the chatbot's capabilities to new heights.

2. **Fortification of Security Measures:** We plan to continuously fortify security measures by exploring cutting-edge encryption methods and proactively integrating measures to preempt and mitigate potential security threats.

3. **Continual Evolution of NLP Models:** We aim to engage in the continual evolution of NLP models. By leveraging advancements in AI technology, we hope to enhance the accuracy, contextuality, and relevance of chatbot responses.

4. **Optimization for Scalability and Performance:** We will prioritize optimizations for scalability and performance enhancements. This will ensure the platform's ability to seamlessly manage increased user loads while maintaining optimal responsiveness.

5. **Strategic Collaborations and Integrations:** We plan to actively pursue strategic collaborations with educational platforms and seamless integrations with learning management systems. This will allow for a wider reach and seamless data integration.

6. **User-Centric Development Iterations:** We will encourage and implement user feedback loops to drive iterative development. By adhering to user-centric design principles, we aim to create tailored and intuitive user experiences.

7. **Utilization of More Cloud Tools:** Future developments will include the use of advanced cloud tools like Azure Bot Services and Document Intelligence. These tools will enhance the chatbot's capabilities in managing files, handling chats, and integrating various backend services.

8. **Unified Backend Service:** Efforts will be made to combine all backend services into a unified system. This next-level chatbot will manage files, handle multiple student chats simultaneously, and optimize the use of AWS S3 for seamless data management.

This comprehensive section encapsulates the achievements of the MERN-based NLP chatbot for Student Information Retrieval and sets the trajectory for continual advancements. It emphasizes the chatbot's evolution, innovation, and adaptability in the education domain.

REFERENCES

[1] N. Rakesh, N. Ravi, O. N. Daivajna and S. Ramesh, "A Proposed Academic Chatbot System using NLP Techniques," 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2022, pp. 1300-1304, doi: 10.1109/ICOEI53556.2022.9777231.

[2] Lalwani, T., Bhalotia, S., Pal, A., Rathod, V., & Bisen, S. (2018). Implementation of a Chatbot System using AI and NLP. Social Science Research Network. https://doi.org/10.2139/ssrn.3531782

[3] H. Abdulla, A. M. Eltahir, S. Alwahaishi, K. Saghair, J. Platos and V. Snasel, "Chatbots Development Using Natural Language Processing: A Review," 2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC), Crete, Greece, 2022, pp. 122-128, doi: 10.1109/CSCC55931.2022.00030.

[4] A. K. Nikhath et al., "An Intelligent College Enquiry Bot using NLP and Deep Learning based techniques," 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-6, doi: 10.1109/ICONAT53423.2022.9725865.

[5] H. K. K., A. K. Palakurthi, V. Putnala and A. Kumar K., "Smart College Chatbot using ML and Python," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2020, pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262426.

[6] Hsu, IC., Yu, JD. A medical Chatbot using machine learning and natural language understanding. Multimed Tools Appl 81, 23777–23799 (2022). https://doi.org/10.1007/s11042-022-12820-4

[7] B. Kohli, T. Choudhury, S. Sharma and P. Kumar, "A Platform for Human-Chatbot Interaction Using Python," 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018, pp. 439-444, doi: 10.1109/ICGCIoT.2018.8753031.

[8] 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, doi: 10.1109/ICACCI.2017.8126057.

[9] J. Purohit, A. Bagwe, R. Mehta, O. Mangaonkar and E. George, "Natural Language Processing based Jaro-The Interviewing Chatbot," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 134-136, doi: 10.1109/ICCMC.2019.8819708.

[10] K, H. K., Palakurthi, A. K., Putnala, V., & K, A. K. (2020). Smart College Chatbot using ML and Python. 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). https://doi.org/10.1109/icscan49426.2020.9262426.

[11] Vijayakumar, R., Bhuvaneshwari, B., Adith, S., & Deepika, M. (2019). AI-Based Student Bot for Academic Information System using Machine Learning. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 590–596. https://doi.org/10.32628/cseit1952171

[12] Sonawane, S. V., & Shanmughasundaram, R. (2019). ChatBot for college websites. International Journal of Innovative Technology and Exploring Engineering, 8(10), 566–569. https://doi.org/10.35940/ijitee.j8867.0881019

[13] Awasthi, S. (2021). Web-based college Chatbot - SDABot. International Journal for Research in Applied Science and Engineering Technology, 9(VIII), 853–856. https://doi.org/10.22214/ijraset.2021.37505

[14] Mekni, M., Baani, Z., & Sulieman, D. (2020). A Smart Virtual Assistant for Students. ACM. https://doi.org/10.1145/3378184.3378199

[15] Arun, K. S., Srinagesh, A., & Ganga, P. (2019). A Multi-Model And Ai-Based Collegebot Management System (Aicms) For Professional Engineering Colleges.

International Journal of Innovative Technology and Exploring Engineering, 8(9), 2910–2914. https://doi.org/10.35940/ijitee.i8818.078919

[16] Martin, C., & Yanyan, L. (2021). A Scalable Self-Learning Chatbot for Classroom. IEEE. https://doi.org/10.1109/CCWC54503.2022.9720766

[17] Chatbot: A User Service for College. (2019). International Journal for Research in Applied Science & Engineering Technology (IJRASET), 7(V). https://doi.org/10.22214/ijraset.2019.5642

[18] NodeJs. Available: https://nodejs.org/en/docs

[19] ExpressJs. Available: https://expressjs.com/

[20] MongoDB. Available: https://www.mongodb.com/

kn

ORIGINALITY REPORT

17%	16%	10%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ir.juit.ac.in:8080 Internet Source	3%
2	Submitted to Jaypee University of Information Technology Student Paper	3%
3	Rohit Negi, Rahul Katarya. "Emerging Trends in Chatbot Development : A Recent Survey of Design, Development and Deployment", 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2023 Publication	1%
4	www.ir.juit.ac.in:8080 Internet Source	1%
5	link.springer.com Internet Source	1%
6	Yazan Alahmed, Reema Abadla, Afnan Al Badri, Nardin Ameen. ""How Does ChatGPT Work" Examining Functionality To The Creative AI CHATGPT on X's (Twitter)	<1%

	Platform", 2023 Tenth International Conference on Social Networks Analysis, Management and Security (SNAMS), 2023 Publication	
7	Submitted to Technological University Dublin Student Paper	<1 %
8	Submitted to Monash University Student Paper	<1 %
9	ejurnalunsam.id Internet Source	<1 %
10	www.ijcaonline.org Internet Source	<1 %
11	www.researchgate.net Internet Source	<1 %
12	www.grafiati.com Internet Source	<1 %
13	Submitted to Universidad Rey Juan Carlos Student Paper	<1 %
14	courses.ece.ubc.ca Internet Source	<1 %
15	Submitted to Universitas Muhammadiyah Surakarta Student Paper	<1 %
16	researchr.org Internet Source	<1 %

17	Submitted to Hong Kong University of Science and Technology Student Paper	<1 %
18	pdfcoffee.com Internet Source	<1 %
19	Submitted to Texas A&M University, College Station Student Paper	<1 %
20	www.ijraset.com Internet Source	<1 %
21	libros.uat.edu.mx Internet Source	<1 %
22	pubs2.ascee.org Internet Source	<1 %
23	scholar.archive.org Internet Source	<1 %
24	Submitted to South Bank University Student Paper	<1 %
25	www.rajalakshmi.org Internet Source	<1 %
26	www.ijdcn.latticescipub.com Internet Source	<1 %
27	ir.uitm.edu.my Internet Source	<1 %

28	Submitted to Ibri College of Technology Student Paper	<1 %
29	journals.stmjournals.com Internet Source	<1 %
30	lib.buet.ac.bd:8080 Internet Source	<1 %
31	ijirset.com Internet Source	<1 %
32	Gunter Bitsch, Pascal Senjic, Petra Kneip. "Hybrid individualized assistance-oriented learning system framework for STEM-Education", 2022 IEEE German Education Conference (GeCon), 2022 Publication	<1 %
33	Xizhe Zhang, Demin Yu, Min Liu, Zhongjie Wang. "A Novel Skill Model Supporting Complex and Reliable Interactions between Users, VPA and External Services", 2022 IEEE International Conference on Edge Computing and Communications (EDGE), 2022 Publication	<1 %
34	gizi.unimus.ac.id Internet Source	<1 %
35	ijsrcseit.com Internet Source	<1 %
	cse.anits.edu.in	

36	Internet Source	<1%
37	ncr.christuniversity.in Internet Source	<1%
38	Submitted to University of Westminster Student Paper	<1%
39	www.mdpi.com Internet Source	<1%
40	1library.net Internet Source	<1%
41	Wenting Liu, Yupeng Lin, Zhonggen Yu. "A Bibliometric Analysis of Artificial Intelligence Chatbots in Language Education", Social Education Research, 2024 Publication	<1%
42	jorgep.com Internet Source	<1%
43	neo4j.com Internet Source	<1%
44	www.allmultidisciplinaryjournal.com Internet Source	<1%
45	www.grin.com Internet Source	<1%
46	"Computational Science and Technology", Springer Science and Business Media LLC,	<1%

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _____ **Department:** _____ **Enrolment No** _____

Contact No. _____ **E-mail.** _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><td>Copy Received on</td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td>Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com