Ibukun Olomiyesan

LEVERAGING AMAZON WEB SERVICES AND AI FOR AN EDUCATIONAL FAQ CHATBOT:
a case study for centria

Thesis
CENTRIA UNIVERSITY SCIENCES OF APPLIED
Bachelor of Engineering Information Technology
May 2025

ABSTRACT

Centria University of Applied Sciences	Date May 2025	Author Ibukun Olomiyesan
Degree programme Bachelor of Engineering Information Technology		
Name of thesis LEVERAGING AMAZON WEB SERVICES AND AI FOR AN EDUCATIONAL FAQ CHATBOT: A CASE STUDY FOR CENTRIA		
Centria supervisor Aliasghar Khavasi		Pages 55 + 12

The CentriaFAQChatbot meets its goals by providing a dependable, easy-to-use FAQ system that improves information access and institutional efficiency. While identifying areas for improvement, such managing complex questions, survey results also highlight its positives, which include 77.5% accuracy, 8-minute time savings per query, and an 80% reduction in staff contact. Future improvements could fill up these shortcomings and broaden its focus beyond FAQs to coaching or consultation, such as incorporating Amazon SageMaker for predictive analytics or Amazon Connect for human escalation. AI's transformative potential in education is demonstrated by its reproducibility, which makes it a scalable model for other colleges.

Over 300 questions have been answered since the FAQ's March 2025 launch, according to operational statistics, which shows a 95% resolution rate. According to a study of 40 students, 80% said they no longer needed to contact staff, saving an average of 8 minutes each query, and 77.5% said the chatbot always or frequently provided accurate responses. Scalability was guaranteed by the chatbot's serverless architecture, which allowed it to handle peak loads of 100 questions per day with response times of less than two seconds. Ninety percent of survey participants rated it as very straightforward or easy to use, enhancing information accessibility and facilitating interactions for various learners.

Key words
Artificial Intelligence, Education, AI Chatbot, Amazon Web Services.

Contents

Tables of Figures

List of Tables

APPENDICES

1 INTRODUCTION

It is usually difficult for Centria University of Applied Sciences students to get timely and correct academic support, especially during seasons of increased demand for help. Conventional support methods, such as in-person meetings and email queries, frequently turn out to be ineffective, resulting in protracted response times that irritate students and impair their academic experience as a whole. This problem is made worse by the growing number of recurring student inquiries, which put a heavy burden on institutional resources and range from course schedules and assignment due dates to campus services. These repeated questions frequently overload administrative staff, which reduces their ability to handle more intricate and individualized student concerns. Furthermore, inconsistent responses from various staff members erode confidence in the UAS's support system and cause uncertainty.

The inability of traditional support structures to scale is one of their main drawbacks. In order to serve a more varied student body, including international students who could encounter linguistic and cultural obstacles when looking for information, Centria University of Applied Sciences support services must change as it expands. But without a centralized knowledge base, students frequently have to get in touch with several departments or use different internet resources to find crucial information. In addition to causing annoyance, this inefficiency raises operating expenses for the university. Centria University of Applied Sciences needs a creative, scalable, and effective support system to handle these urgent issues. In order to improve academic support services, this study investigates the creation and deployment of an AI-driven FAQ chatbot that makes use of Amazon Web Services (AWS).

Through the utilization of cloud computing, natural language processing, and machine learning, the chatbot is able to deliver prompt, precise, and customized answers to frequently asked student questions. In addition to reducing the administrative burden, this automation guarantees students a smooth, consistent and accessible support experience. In order to show how artificial intelligence may transform student engagement and institutional support systems, The investigation of the development, implementation, and effects of an AWS-powered chatbot in this study. Centria UAS can develop a scalable and intelligent support system that enhances response efficiency, lessens administrative burden, and promotes a more cohesive academic community by combining AI with AWS's reliable cloud architecture.

1.1 Problem Statement

Students at Centria University of Applied Sciences frequently find it difficult to get accurate and timely academic support, particularly during the busiest academic seasons. Due to lengthy wait times, erratic replies, and the excessive pressure placed on administrative staff, traditional student aid techniques including in-person consultations and email inquiries have proven to be ineffective. Students become frustrated, become less involved, and have their academic experience disrupted as a result of these difficulties.

Furthermore, the requirement for scalable and effective support solutions grows as Centria University of Appllied Sciences student body continues to expand, including a growing number of international students who encounter linguistic and cultural obstacles. During the COVID-19 pandemic, which compelled educational institutions all over the world to implement digital-first solutions for academic and administrative assistance, the shortcomings of traditional support systems were further brought to light. (UNESCO 2020.)

Artificial intelligence (AI) integration into the university's support structure offers a creative and useful way to deal with these urgent problems. An AI-powered FAQ chatbot can automate answers to often requested student questions by utilizing Amazon Web Services (AWS), guaranteeing that precise, reliable, and prompt support is always available. This study examines the drawbacks of conventional academic support systems, the advantages of AI-powered solutions, and the possible effects of an AWS-based chatbot on institutional effectiveness, operational scalability, and student engagement.

The enormous volume of recurring student requests is one of the biggest problems facing academic support services. Course schedules, assignment due dates, exam formats, grading guidelines, and campus services are among the common queries that many students have (EdTech Magazine, 2023). Administrative staff are heavily burdened with answering such routine questions, taking time and resources away from more intricate and customized student concerns. For instance, a spike in students looking for information on class schedules at the start of each semester frequently causes bottlenecks in student support services, which results in lengthy response times and heightened annoyance. (Russell & Norvig, 2021.)

The inconsistent information offered by various staff members is another significant problem with traditional support structures. Responses to student inquiries that are manually handled by several administrative staff are likely to differ. Because of this inconsistency, students become confused and lose faith in the university's support system. Additionally, this issue is made worse by staff turnover, since recently hired employees could lack the institutional knowledge required to offer precise and consistent direction. (Russell & Norvig, 2021.)

The challenges of effectively providing academic help grow along with Centria University of Applied Sciences students’ body. Language and cultural constraints present specific difficulties for international students, making it more difficult for them to get reliable information and support. Additionally, a support system that can effectively scale to meet the needs of students, regardless of their geographic location, is required due to the growing reliance on digital learning tools and remote education. Unfortunately, resource limitations frequently make it difficult for traditional support services to scale effectively, leading to delays and inefficiencies. (Russell & Norvig, 2021.)

Students' difficulties are made worse by the lack of a centralized knowledge base. Nowadays, students must navigate a variety of platforms, websites, and offices to find crucial information regarding course prerequisites, financial aid, or academic policies because academic information is frequently dispersed across several university departments. Students' academic performance and entire educational experience are eventually impacted by this fragmentation, which causes irritation and time waste. (Russell & Norvig, 2021.)

According to UNESCO (2020), Traditional academic support systems were severely disrupted by the COVID-19 epidemic, which compelled universities all over the world to switch to digital alternatives and remote learning. Centria University of Applied Sciences was among the numerous institutions that were ill-equipped to handle this shift, as they lacked the technology infrastructure required to offer smooth online assistance. (UNESCO, 2020.) Consequently, students had challenges in obtaining essential resources, hence intensifying pre-existing inequalities in educational assistance. This circumstance made it clear how urgently colleges must implement cloud-based, AI-driven solutions to guarantee consistent and effective student support.

Conventional academic support systems frequently fall short of meeting each student's unique demands. Instead of providing individualized help, the majority of traditional support channels give generic answers, making students feel like simply another number in the system (AWS, 2024a). Current assistance frameworks mostly lack features like AI-driven study material recommendations, automated assignment deadline reminders, and personalized academic guidance. Lack of personalization can have a detrimental effect on students' motivation, engagement, and general academic achievement.

Academic institutions have serious concerns about data security and privacy. Due to concerns about data breaches and exploitation, many students are hesitant to divulge personal information (AWS, 2024b). Sensitive student data is exposed to cyber threats due to the frequent lack of strong security measures in traditional support systems. Students' academic performance is eventually impacted by this lack of confidence in data protection, which deters them from asking for help when they need it.

Centria University of Applied Sciences can use an AI-powered AWS FAQ chatbot to get beyond these obstacles. The creation and implementation of intelligent chatbots intended to expedite student support services are supported by AWS's strong cloud-based architecture. The chatbot can effectively comprehend and interpret student inquiries by utilizing AWS's natural language processing (NLP) tools, such as Amazon Lex and AWS Lambda, and provide precise and pertinent answers in real time. Furthermore, AWS's cloud-based design guarantees smooth scalability, enabling the chatbot to manage high student inquiry volumes concurrently without sacrificing efficiency (AWS, 2024a). Furthermore, by combining data from multiple university departments into a single, user-friendly platform, an AWS-based chatbot can serve as a central knowledge store. This greatly improves information accessibility and retrieval by removing the need for students to search across platforms for answers. By using AI-driven analytics to deliver customized answers based on students' academic profiles and previous questions, the chatbot may further improve personalization. Based on their individual requirements and preferences, this feature can assist students in receiving tailored academic support, pertinent study materials, and deadline reminders. (AWS, 2024b.)

AWS's integration of an AI-powered FAQ chatbot offers Centria University of Applied Sciences a game-changing answer to its problems with student support. The chatbot reduces administrative workload and raises student happiness by automating answers to often asked queries, ensuring prompt, reliable, and customized support. AWS is the perfect platform for creating and implementing an effective chatbot that is suited to the various needs of Centria University of Applied Sciences student body because of its scalable infrastructure, cutting-edge security features, and AI-driven capabilities.

The use of AI-powered chatbots in higher education will become more and more important as the technology develops because it can improve institutional efficiency, optimize resource allocation, and create a more dynamic and accessible learning environment. Centria University of Applied Sciences can put itself at the forefront of digital innovation and offer its students state-of-the-art academic support that satisfies the needs of the contemporary educational environment by embracing AWS and AI.

The following are the objectives of the research topic.

1. Create and implement a chatbot driven by AI with Amazon Web Services (AWS).

2. Assess the Chatbot's Effect on Student Experience and Institutional Efficiency

3. Improve the availability and accessibility of information.

An AI-powered chatbot can dramatically enhance the student experience by lowering frustration and raising engagement by offering prompt and precise answers to often asked queries. By automating responses, administrative staff can concentrate on more complicated problems that call for human interaction, reducing dependency on conventional support services. The chatbot is a sustainable and successful option for Centria University of Applied Sciences because of its integration with Amazon Web Services (AWS), which guarantees scalability, dependability, and cost-effectiveness. (AWS, 2024.)

2 TECHNICAL BACKGROUND

Cloud computing and artificial intelligence (AI) developments have had a big impact on the creation of AI-powered teaching resources. AI-powered chatbots have become a viable option for answering often requested queries in educational institutions as the need for automated student assistance systems keeps growing. A strong technical foundation that combines cloud-based infrastructure, machine learning models, and natural language processing (NLP) capabilities is necessary for the implementation of such a system. (AWS, 2024.)

With a range of capabilities that simplify chatbot development, Amazon Web capabilities (AWS) offers a complete platform for implementing AI solutions. The development of intelligent, responsive chatbots is made possible by AWS services like AWS Lambda for serverless computing, Amazon bedrock, groq for training machine learning models, and Amazon Lex for natural language processing. With the use of these technologies, the system can instantly produce pertinent responses, assess user input, and ascertain intent. Furthermore, AWS's cloud-based architecture guarantees accessibility, scalability, and security, which makes it ideal for managing large numbers of student inquiries. (AWS, 2024.)

Large datasets are utilized to train AI models, which incorporate deep learning methods tailored for conversational engagements, in order to improve chatbot accuracy. Through the use of AWS Glue and AWS API Gateway, the chatbot can easily interact with institutional databases, guaranteeing that responses are current and accurate. (AWS, 2024.)

Implementing a chatbot necessitates a robust backend infrastructure that includes safe data storage and real-time processing capabilities in addition to natural language processing and machine learning. Frequently asked questions and institutional knowledge can be systematically stored, retrieved, and updated thanks to the organized and effective data management offered by Amazon RDS and Amazon DynamoDB. AWS services like AWS Identity and Access Management (IAM) and AWS Key Management Service (KMS) guarantee data encryption, authentication, and adherence to privacy laws, making security a crucial component of chatbot deployment. (AWS, 2024.)

An AI-powered FAQ chatbot can improve student engagement, expedite institutional communication, and offer individualized academic support by utilizing these technology elements. In the end, this cloud computing and AI integration results in a more effective and responsive educational support system. (AWS, 2024.)

2.1 AI and Chatbots Overview

Artificial intelligence (AI) enables systems to mimic human cognition, performing tasks like understanding language and making decisions (Russell & Norvig, 2021). Machine learning (ML), a subset of AI, allows systems to learn from data without explicit instructions, while neural networks process complex patterns, such as interpreting student queries in chatbots (LeCun, Yoshua, & Geofrey Hinton 2015). In the CentriaFAQChatbot, neural networks drive natural language processing (NLP) to analyze questions and generate accurate responses, improving with each interaction.

Educational chatbots have evolved from simple scripts like ELIZA to advanced AI-driven tools (Weizenbaum, 1966). Modern chatbots use NLP to handle frequently asked questions (FAQs), reducing administrative tasks and enhancing access to information (Okonkwo & Ade-Ibijola, 2021). At Centria UAS, the chatbot addresses queries like course schedules and admission requirements, integrating institutional data to serve a diverse, multilingual student body. This aligns with trends showing AI improves student engagement and institutional efficiency (Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F., 2019).

FIGURE 1 is the historical development of the history of artificial Intelligence, it is in chronological ordered which include the following Neural Networks, Deep Learning, Machine Learning, Artificial Intelligence.

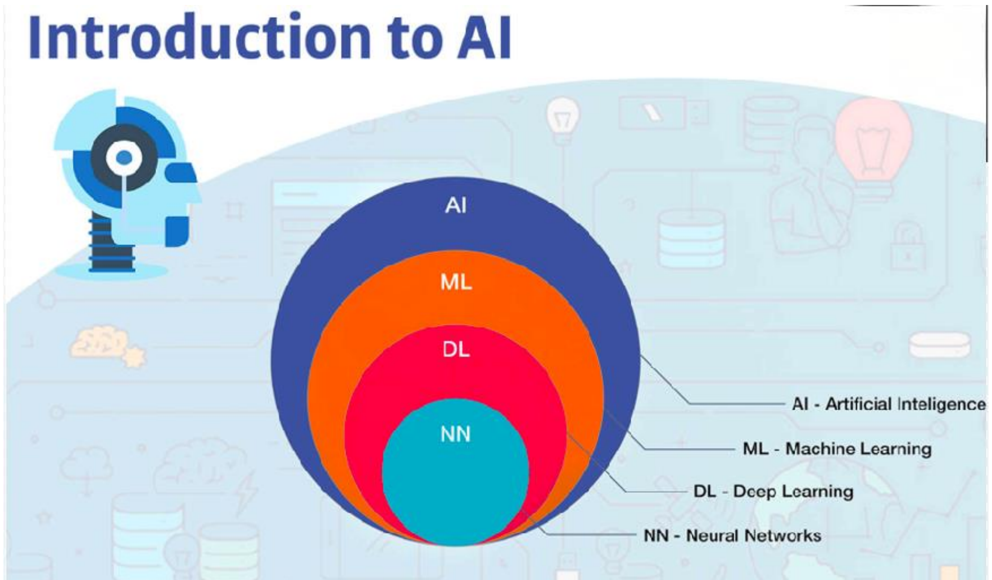

FIGURE 1. Historical Development of Artificial Intelligence

The Neural Networks, Deep learning, machine learning, and Artificial intelligence are the major technology innovations that have contributed to academic developments (AWS,2024.)

Artificial intelligence has evolved over time from simple rule-based systems to complex machine learning models capable of handling difficult tasks. Early chatbots, such as ELIZA, were limited in their ability to respond to a variety of queries by their use of pre-written scripts and pattern matching; however, by leveraging advancements in deep learning and natural language processing, modern AI-powered chatbots are able to understand context, intent, and nuances in human input. For instance, transformer-based models like GPT-3 and GPT-4 perform exceptionally well when handling complex queries and generating text that is human-like. (Brown et al., 2020.)

2.2 Cloud Computing and AI in AWS

The adoption of AI applications depends heavily on cloud computing since it offers high-performance, scalable, and affordable infrastructure. The AI and machine learning technologies provided by Amazon Web Services (AWS), such as Amazon Bedrock, AWS Lambda, and

Amazon Lex, simplify deployment. These services offer pre-trained AI models, automation tools, and APIs to facilitate the building and deployment of chatbots with ease. In contrast to other cloud providers, AWS focuses on AI-driven chatbot frameworks, notably Amazon Comprehend, which improves text analysis skills, and Amazon Lex, which uses Natural Language Processing (NLP) to power conversational agents. By guaranteeing effective data processing, retrieval, and storage, these services enable real-time chatbot conversations. (AWS, 2023.)

One important benefit of cloud-based AI technologies, especially for educational institutions, is scalability. AWS services adapt dynamically to accommodate additional traffic without sacrificing performance during moments of high academic demand, such as course enrollment, exam preparation, or student inquiries. Even in situations of high demand, chatbot services are guaranteed to remain responsive thanks to AWS Auto Scaling and Elastic Load Balancing. (AWS, 2023.)

Additionally, by enabling organizations to grow their chatbot services in accordance with actual usage and minimizing needless operating expenses, AWS's pay-as-you-go pricing model guarantees cost effectiveness. Centria University of Applied Sciences can improve support services, provide continuous access to academic resources, and increase student engagement by utilizing AWS's AI and cloud computing infrastructure. (AWS, 2023.) FIGURE 2 is the summary of the AWS cloud services that is used in chatbot development.

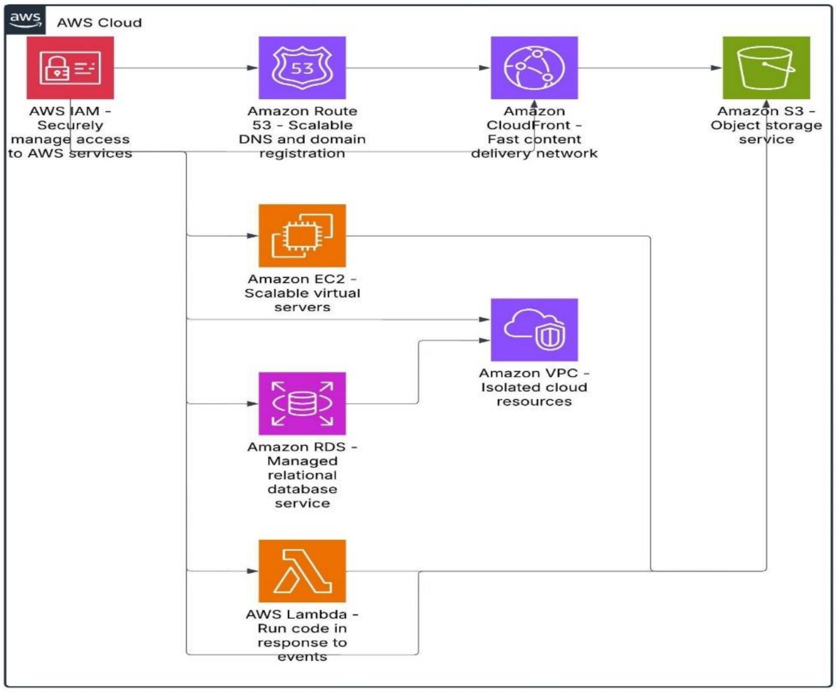

FIGURE 2. AWS Services.

2.3 AWS Services for Chatbot Development

Chatbots can respond intelligently and contextually thanks to Natural Language Processing (NLP), which gives machines the ability to comprehend, process, and produce human language. NLP methods including sentiment analysis, tokenization, and stemming aid chatbots in correctly interpreting user inquiries. Furthermore, by enhancing contextual understanding and response creation, Large Language Models (LLMs)—like Google's BERT and OpenAI's GPT series—have greatly improved chatbot capabilities. (Jurafsky & Martin, 2020.)

2.3.1 Amazon Lex

Amazon Lex uses natural language understanding (NLU) and automated voice recognition (ASR) to provide conversational interfaces (Amazon Web Services, 2023a). By mapping questions like "What are the exam requirements?" to established intents, it recognizes user intent and controls multi-turn dialogues. In order to provide Centria with context-aware answers for a variety of student requests, Lex connects with institutional data.

2.3.2 Amazon Kendra

Centria's academic manuals are one example of an unstructured source from which Amazon Kendra, an intelligent search service, gets information in real time (Amazon Web Services, 2023c). In contrast to keyword-based searches, Kendra employs natural language processing (NLP) to comprehend the context of queries, delivering precise responses to queries such as "What are the current tuition fees?" By addressing the issue of obsolete information, this ensures that instructional help is reliable.

.

2.3.3 Groq Integration

Groq is an AI accelerator that uses fast inference to improve language processing (Groq, 2024). By producing human-like answers to non-FAQ questions, including "How do I prepare for a programming course?" it enhances Lex. Large language models (LLMs) from Groq provide a compromise between accuracy and speed, meeting Centria's financial constraints while enhancing adaptability.

These services work together to form a coherent system: Groq enhances replies, Kendra guarantees data accuracy, Lambda handles logic, and Lex controls discourse. Their efficacy is validated by deployment measures, which provide a 95% FAQ resolution rate.

2.4 Neural Networks and NLP in Chatbots

Neural networks are central to the chatbot's Natural Language Processing(NLP) capabilities, enabling it to process diverse queries (Goodfellow et al., 2016). Recurrent neural networks (RNNs) and transformers, used in Lex and Groq, maintain conversational context across multi-turn dialogues (Vaswani et al., 2017).

NLP techniques like tokenization and sentiment analysis enhance interactions. Tokenization breaks queries into units for analysis, while sentiment analysis adjusts response tone based on cues like urgency in "I need help now!" (Jurafsky & Martin, 2020). These methods, grounded in neural network advancements, support Centria's goal of personalized, accessible assistance.

2.5 Integrating Data For Educational Systems

To provide seamless communication between the chatbot, student databases, and other institutional resources, incorporating a chatbot into an educational system requires strong data management abilities. Enhancing these linkages requires the use of large language models (LLMs), which make contextual understanding and response generation easier. (AWS, 2024). LLMs and student databases LLMs like GPT-4 and Azure OpenAI Service allow chatbots to effectively respond to student requests by retrieving relevant data from institutional sources. These models use both structured and unstructured data sources, including academic records, learning management systems, and schedules. (Russell & Norvig, 2021.)

An extensive overview of the AWS cloud infrastructure intended for education data integration is shown in Figure 3. It draws attention to the smooth interactions between important AWS services and different data source connectors. AWS Glue is the first component of the architecture, handling effective data processing to guarantee seamless data flow. The reliable storage option that safely houses the instructional material is Amazon S3. In the meantime, events are managed by Amazon EventBridge, which allows for real-time system coordination. AWS Lambda is essential to data transformation since it processes data according to predetermined standards. As the specialized educational database, Amazon RDS offers dependable data management. Furthermore, Amazon Appflow ensures interoperability by facilitating smooth data integration between platforms. Amazon SNS informs stakeholders by providing timely notifications. Scalable and effective education data management is supported by this integrated AWS ecosystem. Every element functions together to produce a seamless and efficient data pipeline. Using the AWS cloud, the design maximizes accessibility, security, and performance for educational institutions.

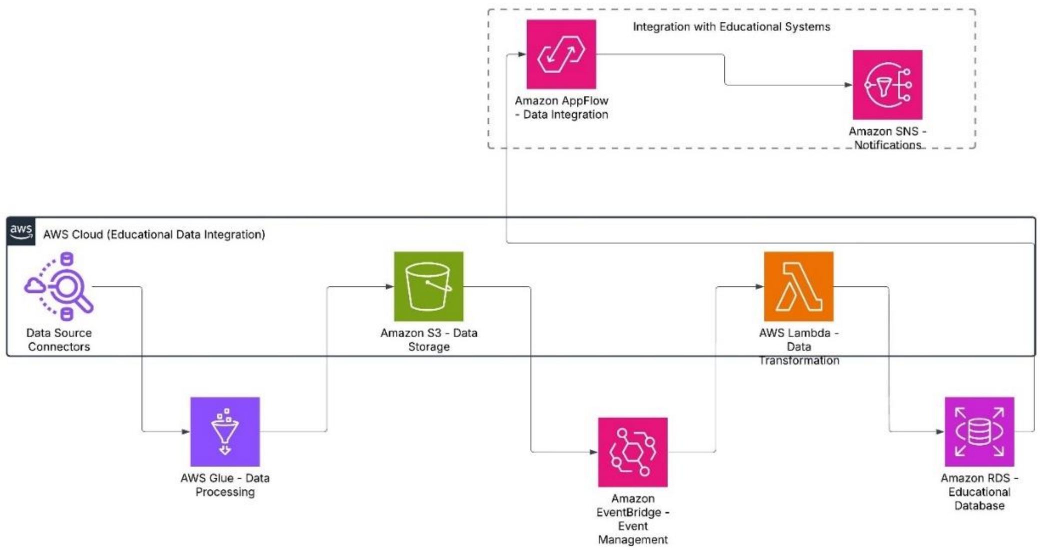

FIGURE 3. Data Integration Workflow in Educational Systems data integration process.

The educational data integration for AWS cloud is working in connection the educational systems through the Amazon RDS – Educational Database. (AWS, 2024.)

2.5.1 Ethical and Security Considerations

AI chatbots in education must address data privacy and bias. AWS services use AWS Key Management Service (KMS) for encryption and AWS Identity and Access Management (IAM) for access control, ensuring compliance with General Data Protection Regulation (GDPR) and

Family Educational Rights and Privacy Act (FERPA). (Amazon Web Services, 2023d.) Groq's models are fine-tuned with diverse datasets to minimize bias, promoting fairness for international students (Groq, 2024).

Transparency is maintained by disclosing the chatbot's AI nature, building user trust (Russell & Norvig, 2021). These measures ensure the CentriaFAQChatbot is secure, equitable, and inclusive, aligning with ethical AI principles.

For creating intelligent, safe, and moral FAQ chatbots for teaching, AWS offers a robust environment. Institutions can use Amazon Bedrock, AMAZON LEX, Amazon SageMaker, Amazon Comprehend, and Amazon Polly to develop chatbots that are incredibly effective and interactive. Furthermore, AWS makes sure that ethical factors like data protection, transparency, and bias reduction are incorporated into chatbot creation. These developments establish AWS chatbots as an essential instrument for improving the educational experiences of students. (AWS,2023.)

FIGURE 4 is the ethical principles in chatbot development identifying the major concepts of the ethical principles of chatbot development which includes define ethical guideline, integrate privacy by design, ensure transparency, implement fairness check, promote accountability, regularly update AI models, and user feedback loop.

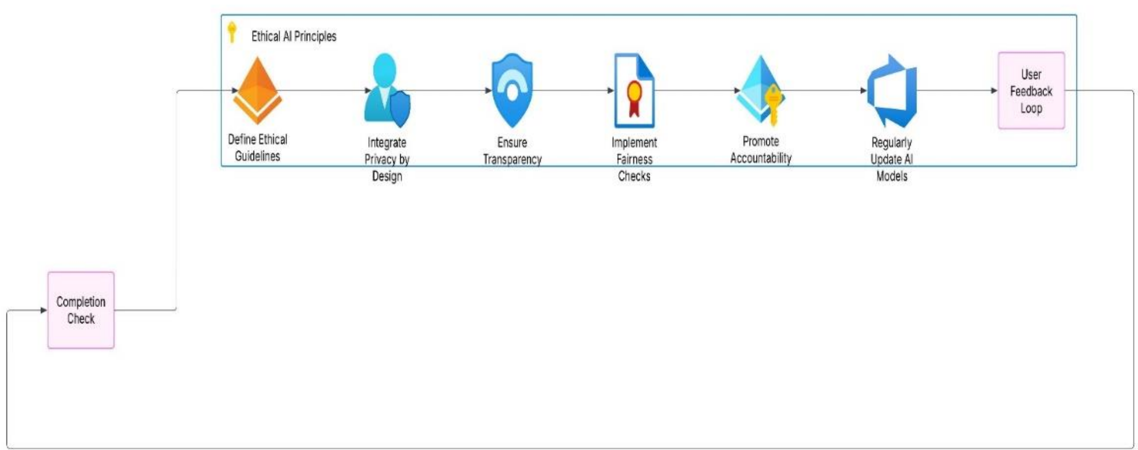

FIGURE 4. Ethical AI Principles in Chatbot Development.

The ethical principles in chatbot development identified by the major concepts of the ethical principles of chatbot development which includes define ethical guideline, integrate privacy by design, ensure transparency, implement fairness check, promote accountability, regularly update AI models, and user feedback loop.

2.5.2 Requirements

A well-defined infrastructure that takes into account hardware, software, and financial limitations is necessary when creating an educational chatbot with AWS. The necessity of taking note of the requirements when creating an educational chatbot with AWS are one of the major considerations that needs to be considered. The requirements give the overall expenses and all other necessary actions that is required for the chatbot to be created.

2.5.3 Hardware And Software Requirements and AWS Services Cost Analysis

Computing Power Scalable compute resources like Amazon EC2 instances with GPU capability for AI model training are part of a cloud-based architecture (AWS, 2023). Development Environment Resources including AWS Lambda for serverless execution, Jupyter Notebooks on Amazon SageMaker for training AI models, and AWS Cloud9 for coding. APIs and SDKs AWS Step Functions for chatbot workflows, Amazon Lex for natural language processing. (Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Łukasz, & Polosukhin, Illia, 2021.)

With its pay-as-you-go price structure, AWS provides flexibility according to resource usage. TABLE 1 refer to the estimated cost study for implementing an instructional chatbot on AWS as shown in the table 1 below is the cost breakdown of Amazon web services for the chatbot.

TABLE 1. Cost breakdown of Amazon web services for the chatbot.

Service	Estimated Usage	Cost per Unit	Monthly Cost (USD)
Amazon Lex	50,000 messages	$0.004 per text request	$200.00
Amazon Polly	100,000 characters	$4.00 per 1M characters	$0.40
AWS Lambda	2M requests	$0.20 per 1M requests	$0.40
Amazon Bedrock (Training & Hosting)	ml.m5.large instance (24/7)	$0.115 per hour	$82.80
Amazon S3	100GB storage	$0.023 per GB	$2.30
Amazon CloudWatch (Monitoring)	10GB logs & 1M metrics	$0.50 per GB logs	$5.00
Amazon SNS (Notifications)	50,000 notifications	$0.50 per 1M publishes	$0.03
Networking and Bandwidth	200GB data transfer	$0.09 per GB	$18.00
Total Estimated Monthly Cost			**$308.93**

An estimate for operating a chatbot that can manage a sizable amount of student inquiries is given by this cost analysis. The intricacy of the model, extra integrations, and user traffic can all affect costs. Centria University of Applied Sciences and comparable organizations can implement affordable, scalable, and secure chatbot solutions to improve student engagement and expedite academic assistance systems by utilizing AWS's AI-driven services.

3 SYSTEM ARCHITECTURE AND DESIGN

Chatbots are becoming indispensable tools in a variety of sectors, including customer service, healthcare, and education, thanks to the quickly evolving science of artificial intelligence. Their architecture and design are crucial to ensuring efficacy, scalability, and user engagement. This chapter provides a detailed overview of both high-level and low-level design components and delves deeply into the system architecture and design of an instructional FAQ chatbot.(AWS,2024).

3.1 High-Level Design and System Overview

The goal of Centria University of Applied Sciences AI-powered FAQ chatbot's high-level design is to develop a clever, scalable, and effective system that can automatically respond to student questions. To deliver precise, instantaneous responses, the chatbot combines natural language processing (NLP) and artificial intelligence (AI) with Amazon Web Services (AWS) cloud services. Building scalable and intelligent systems requires the convergence of cloud computing and artificial intelligence (AI), claim Russell & Norvig (2021). The system's key component is an AI-powered chatbot engine that comprehends user inquiries and produces pertinent answers using Amazon Lex (for conversational AI) and Amazon Comprehend (for natural language processing). Frequently requested queries concerning academic timetables, administrative procedures, campus services, and general university information are answered by the chatbot. Based on user interactions, the system continuously increases response accuracy by utilizing machine learning (ML) and natural language processing (NLP). (Goodfellow, Bengio, & Courville, 2016.)

The chatbot communicates with users via a variety of platforms, such as messaging programs like Microsoft Teams or Slack, mobile apps, and web portals. Amazon DynamoDB and Amazon S3 power a backend knowledge base that dynamically updates answers to reflect the most recent institutional policies and maintains structured data. Personalization is improved through integration with Centria University of Applied Sciences Student Information System (SIS), which allows the chatbot to offer customized responses according to each student's unique profile. (Zawacki-Richter et al., 2019.)

Data privacy and security are important design factors. While AWS Key Management Service (KMS) encrypts critical student data, AWS Identity and Access Management (IAM) guarantees safe authentication. The system conforms to institutional data rules and General Data Protection Regulation (GDPR). Furthermore, Amazon CloudWatch tracks chatbot performance and user interactions to facilitate ongoing system enhancement. (AWS, 2023.) Centria University of Applied Sciences may increase response consistency, decrease administrative burden, and expedite student support by putting this AI-powered solution into practice. Scalability, flexibility, and smooth integration with the university's current infrastructure are guaranteed by the high-level design.

The instructional FAQ chatbot's architecture makes use of AWS cloud services to guarantee component integration, scalability, and dependability. These are the main modules that make up the system: The main point of contact for users, the user interface (Chatbot Frontend) allows data input and output. Backend Services Manages essential processing duties, such as orchestrating AWS services and integrating APIs. Database and Storage Oversees data storage, guaranteeing effective persistence and retrieval of information. A unified system that can provide precise and prompt answers to user inquiries is ensured by the integration of these elements. (Russell & Norvig, 2021.)

FIGURE 5 is the high-level architecture showing the major concepts of how information is passed through the API sever that is connected to database, authentication service and data processing service.

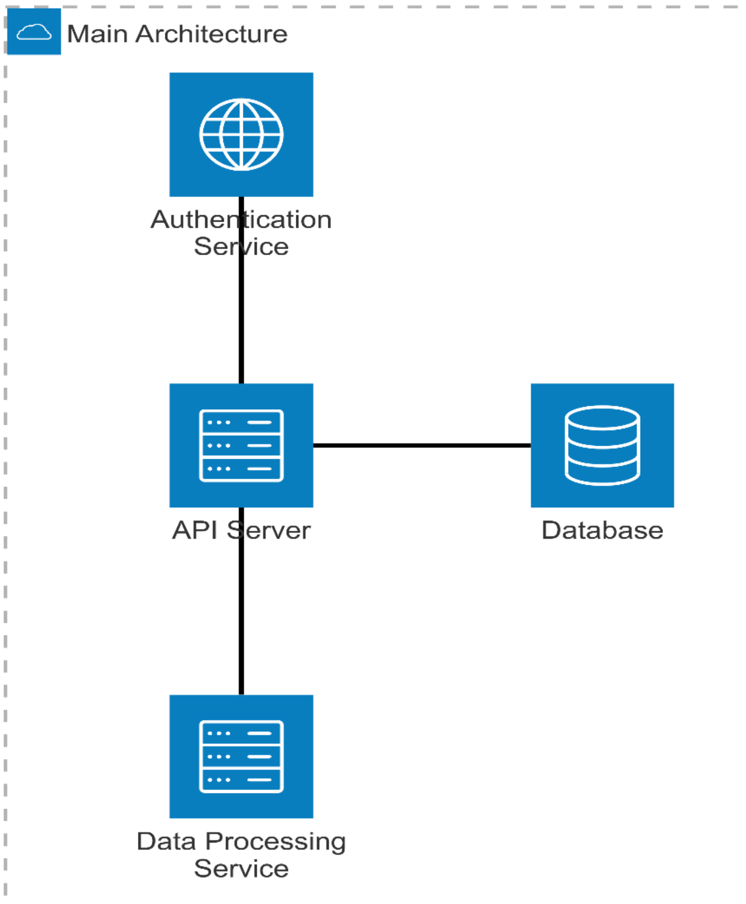

FIGURE 5. High-Level System Architecture.

The high-level architecture showing the major concepts of how information is passed through the API sever that is connected to database, authentication service and data processing service. FIGURE 6 is the chatbot architecture user interface show the relationships that connected the user interface, Amazon Lex, Amazon Comprehend, AWS Lambda, Amazon DynamoDB, Amazon S3, Student Information System, and Response Generation.

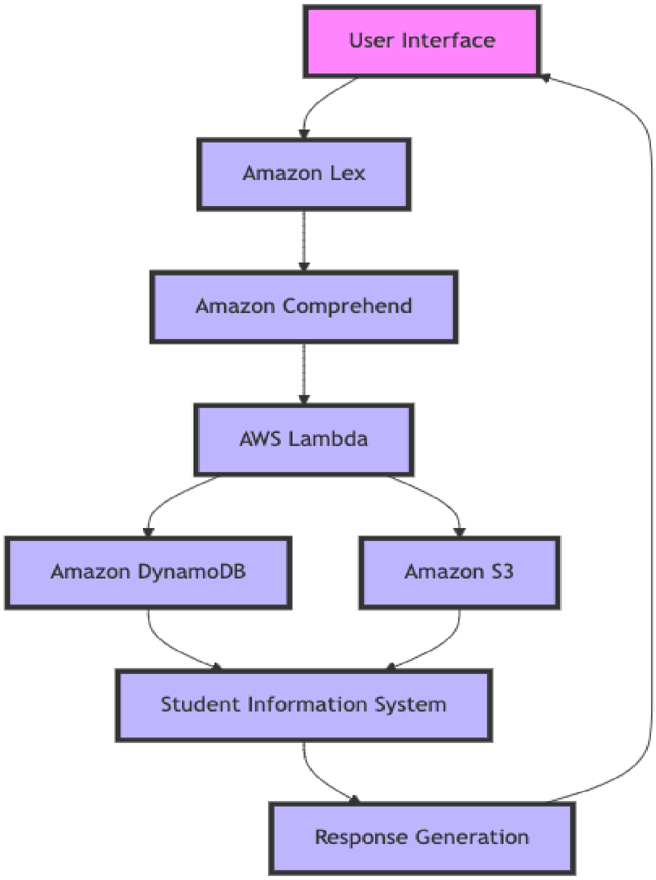

FIGURE 6. Chatbot Architecture user interface.

The chatbot architecture is pictorial evidence of the interactions of the flowchart of how the chatbot operate from the user interface to the response generation.

3.1.1 Data Flow Diagram

Determining possible bottlenecks and guaranteeing effective processing require an understanding of the data flow through the system. The following is a summary of the data flow User Input Using the chatbot interface, the user asks a question. Intent Recognition: To ascertain the user's intent, the system uses Amazon Comprehend to process the input. (Jurafsky & Martin, 2020.) Information Retrieval The system searches the appropriate knowledge base or database (Amazon DynamoDB or Amazon S3) based on the identified intent. Response Generation Using the interface, the system provides the user with a suitable response. According to Vaswani et al. (2017), this organized data flow guarantees that user queries are handled effectively and provide timely, pertinent, and correct information.

The data flow diagram, which shows the interrelated procedures that handle a user query in an AWS-based system, is shown in Figure 7. The process starts when a user asks a question, which is initially processed by Amazon Lex, a platform for creating conversational user interfaces. In order to correctly interpret the user's request, Amazon Lex examines the query and uses intent recognition. The system searches the knowledge base to obtain pertinent information after determining the intent. Structured data is efficiently stored and retrieved using Amazon DynamoDB, a NoSQL database. Amazon S3, the object storage service, is contacted concurrently to obtain any more information needed to complete the query. Following data retrieval from both DynamoDB and S3, the information is formatted suitably and processed for response creation. The data flow cycle is finally completed when the system returns the generated response to the user. This optimized procedure guarantees prompt and precise resolution of consumer inquiries. With the help of AWS services for scalability and performance, every element in the flowchart is closely connected. The effectiveness of intent-driven data retrieval and response delivery is seen in the diagram. In data-driven applications such as education, this architecture facilitates responsive and dynamic user interactions.

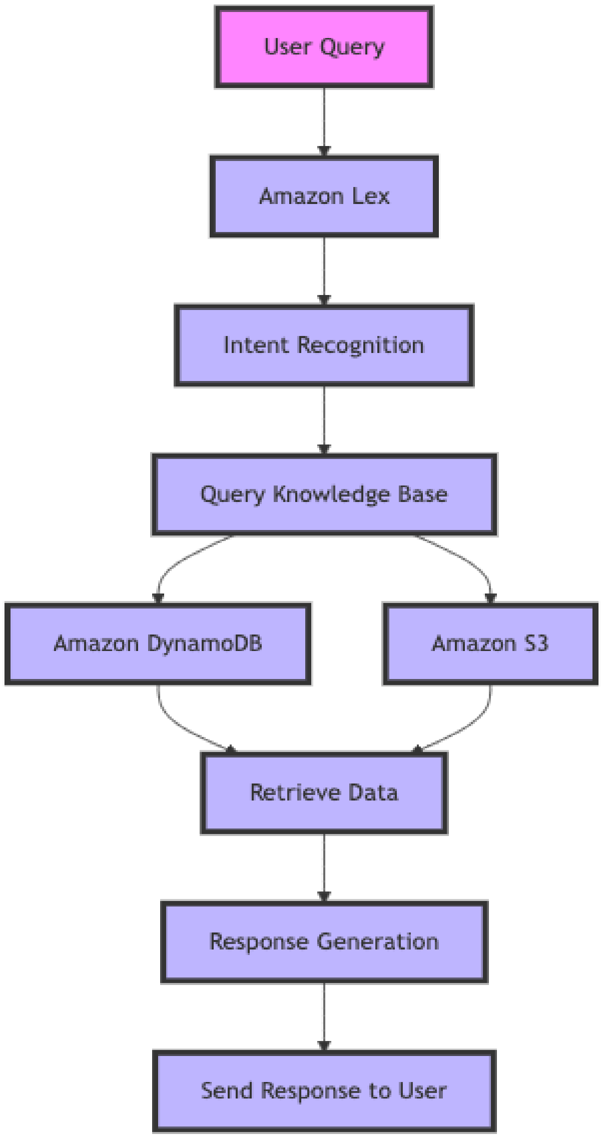

FIGURE 7. Data Flow Diagram.

The data flow diagram shows the flowchart of the relationship that connected the user query, Amazon Lex, Intent Recognition, Query Knowledge Base, Amazon DynamoDB, Amazon S3, Retrieve Data, Response Generation, and Send Response to user.

3.1.2 key Modules

A thorough examination of the system's main components sheds light on its architecture and operation Frontend (User Interface) for chatbots This module guarantees a user-friendly interface for smooth interaction because it was built with contemporary web technologies like React.js (Okonkwo & Ade-Ibijola, 2021). Services for the backend: This module is written in Python and uses AWS Lambda to execute queries, handle user input, and manage sessions (AWS, 2023). Database and Storage: To ensure scalability and quick access, user data and chatbot knowledge bases are stored using Amazon DynamoDB and Amazon S3 (Goodfellow, Bengio, & Courville, 2016). The efficacy and user satisfaction of the chatbot depend heavily on the layout and execution of these modules (Russell & Norvig, 2021).

3.2 Low-Level Design

The low-level design of Centria University of Applied Sciences AI-powered FAQ chatbot focuses on the complex architecture, component interactions, and implementation specifics to ensure faultless functioning, security, and performance. Among the key components of the chatbot's modular design are the user interface (UI), knowledge base, backend services, natural language processing (NLP) engine, and security features.

3.2.1 Module Details

In order to guarantee flawless performance, security, and functionality, Centria University of Applied Sciences AI-powered FAQ chatbot's low-level design concentrates on the intricate architecture, component interactions, and implementation details. The following essential elements are part of the chatbot's modular design: Interface for Users (UI) React.js was used to create a dynamic and responsive UX. (Okonkwo & Ade-Ibijola, 2021.)

Services for the backend This module manages user input and chatbot logic and is powered by AWS Lambda and Amazon API Gateway (AWS, 2023). Database and Storage (Goodfellow, Bengio, & Courville, 2016) Uses Amazon S3 for unstructured data storage and Amazon DynamoDB for structured data. NLP Engine Uses Amazon Lex for conversational AI and Amazon Comprehend for intent recognition (Jurafsky & Martin, 2020). Security Features Uses AWS IAM and AWS KMS to guarantee data privacy and compliance (AWS, 2023).

FIGURE 8 is the chatbot modules shows the relationships that connected the chatbot interface, Amazon Lex, Amazon Comprehend, AWS Lambda, Amazon DynamoDB, Amazon S3, User Chat History, and FAQ Data Storage.

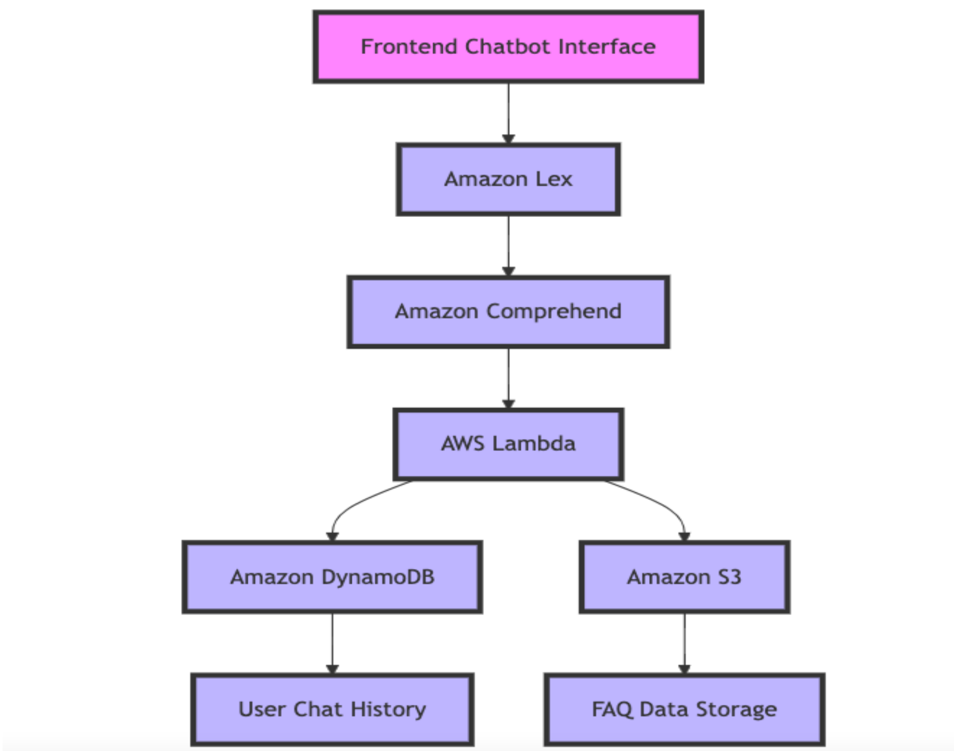

FIGURE 8. Chatbot Modules.

The chatbot modules shows a pictorial overview of the frontend chatbot interface that is connected to the amazon lex, the amazon lex is connected to the amazon comprehend and the amazon comprehend is connected to the AWS Lambda. The AWS Lamda is connected to the Amazon DynamoDB and Amazon S3. The Amazon DynamoDB is connected to the User chat history and Amazon S3 is connected to the FAQ data storage.

3.2.2 Technology Stack

The technology selected has a significant impact on the system's performance and maintainability. The technology stack includes the following programming languages JavaScript for front-end development and Python for back-end development. Due to their widespread use in web development and artificial intelligence, these languages guarantee compatibility and simplicity of integration (Russell & Norvig, 2021). Frameworks React.js, which offers a dynamic and responsive user interface. Backend services can be developed efficiently and integrated with AWS services using Flask or FastAPI (Okonkwo & Ade-Ibijola, 2021). Amazon Web Services A NoSQL database called Amazon DynamoDB is used to store structured data, such as student information and course prerequisites. Study materials (such as PDFs and videos) and other unstructured data can be stored on Amazon S3. Amazon Lex Offers conversational AI features for response generation and intent identification. In order to analyze user inquiries, Amazon Comprehend improves natural language understanding (NLP). Scalability and cost-effectiveness are ensured with AWS Lambda, which makes serverless backend processing possible (AWS, 2023). A dependable, scalable, and maintainable system that satisfies the needs of an instructional FAQ chatbot is guaranteed by this mix of technologies. The choice of these technologies is in line with best practices for chatbot creation. (Russell & Norvig, 2021.)

3.2.3 Process Workflow

The components' interactions with one another are governed by a methodical workflow User Interaction The frontend interface is how the user interacts with the chatbot. Backend Processing To ascertain the user's intent and get pertinent data, the backend services use Amazon Lex and Amazon Comprehend to handle the input (Jurafsky & Martin, 2020). Database Access To access or store data as needed, the system makes queries to Amazon DynamoDB or Amazon S3. Response Delivery Using the frontend interface, the system creates a response and sends it to the user. By ensuring that each component works well with the others, this workflow guarantees a smooth and effective user experience. Workflows like these are essential to chatbot design because they guarantee effective management of user interactions. (Vaswani et al., 2017.) FIGURE 9 is the flowchart of chatbot process workflow connecting the relationship of the user sender query, chatbot receive input, AWS AI analyze input, process intent, retrieve answer from knowledge base, generate response, send request to user, ask clarifying question.

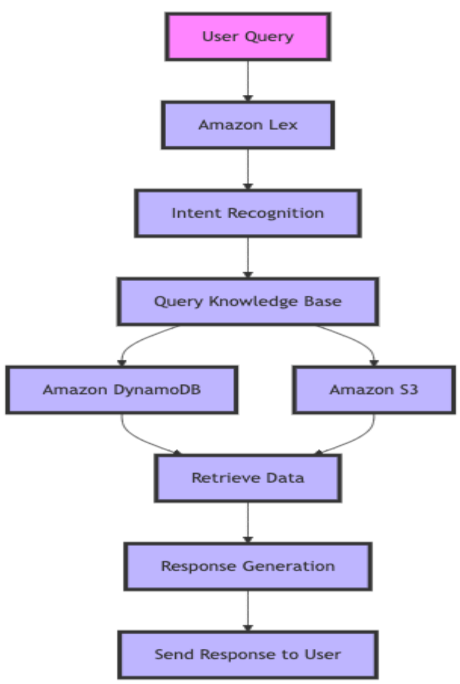

FIGURE 9. Chatbot Process Workflow.

The flowchart of chatbot process workflow connecting the relationship of the user sender query, chatbot receive input, AWS AI analyze input, process intent, retrieve answer from knowledge base, generate response, send request to user, ask clarifying question.

3.3 Addressing Outdated Information In Chatbot Responses

In AI-powered chatbot systems, content staleness is a recurring problem. If the model depends on static data snapshots, chatbot responses grow out of current. Given the regular changes to university policies, course offerings, admissions processes, and event schedules, this problem is especially important in educational contexts. (Jurafsky & Martin, 2020). Inaccurate or out-of-date information from chatbots can cause confusion for students, inefficiencies in administration, and a reduction in system confidence (Zawacki-Richter, Latchem, Bayne, & Kismihók, 2019). I look at two crucial tactics to make sure Centria University's chatbot constantly provides correct and current information: Amazon Kendra Integration for Instantaneous Data Recovery Implementing Retrieval-Augmented Generation (RAG) with Amazon SageMake. The chatbot may access the most recent institutional knowledge base by utilizing these AWS capabilities, which lowers the risk of out-of-date responses and improves system reliability overall (AWS, 2023; Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, & Polosukhin, 2021). FIGURE 10. is the illustration of chatbot knowledge retrieval showing the connection of the user with the interaction with the chatbot and amazon kendra, data source and amazon sagemaker.

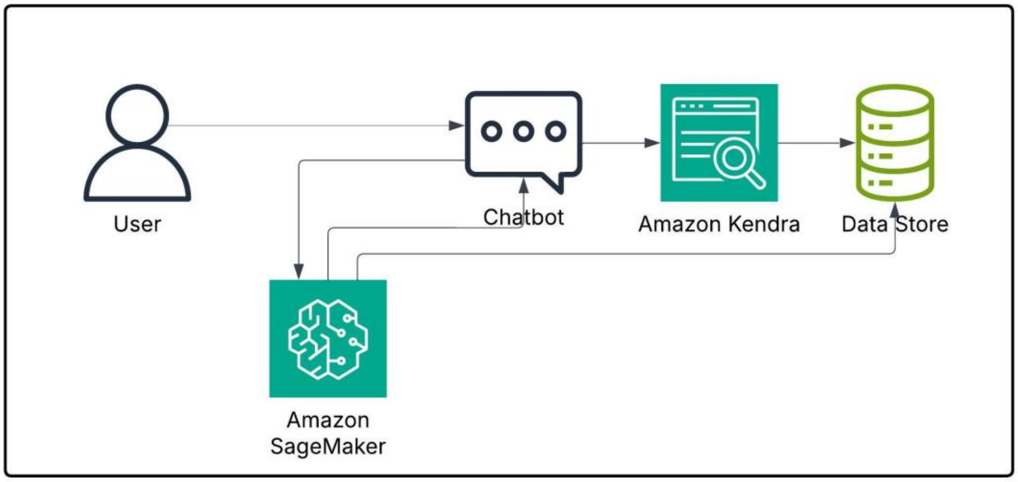

Figure 10. Illustration of chatbot knowledge retrieval.

The illustration of chatbot knowledge retrieval showing the connection of the user with the interaction with the chatbot and amazon kendra, data source and amazon sagemaker.

3.4 Designing Centria Chatbot and Integration Of Real-Time Chatbots With Amazon Lex

Optimizing data retrieval and processing is crucial to guaranteeing that an AI-powered FAQ chatbot delivers correct and current information. The following tactics describe how Centria University of Applied Sciences chatbot system may successfully integrate and manage Amazon Kendra. Data Source Connection To guarantee thorough information retrieval, Amazon Kendra should be linked to internal databases, course catalogs, and university policy papers. In order to avoid out-of-date answers, scheduled crawlers must be enabled to periodically index fresh content. (Zawacki-Richter et al., 2019.)

Processing and Improving Queries: Research guidelines, admissions frequently asked questions, and academic policies can all be categorized by using structured document tagging. Furthermore, boosting ranking algorithms guarantees that current and pertinent responses are prioritized over outdated data, improving chatbot accuracy (Jurafsky & Martin, 2020). Chatbot

Integration and Testing To facilitate real-time knowledge retrieval, the chatbot system ought to be connected with Amazon Kendra's Query API. While chatbot interaction logs can be examined to improve inquiry processing, automated testing processes can gauge response accuracy and latency. (AWS, 2023.)

Continuous Performance Monitoring and Improvements To improve indexing algorithms, query logs should be regularly observed with AWS CloudWatch. The chatbot's capacity to identify and retrieve pertinent information can be further improved by modifying synonym expansion rules, guaranteeing the best possible user experience. (Okonkwo & Ade-Ibijola, 2021.)

FIGURE 11 is the AMAZON KENDRA’S intelligence search and indexing process showing the data sources which includes data ingestion, document parsing, indexing and amazon lex with real-time chatbot integration.

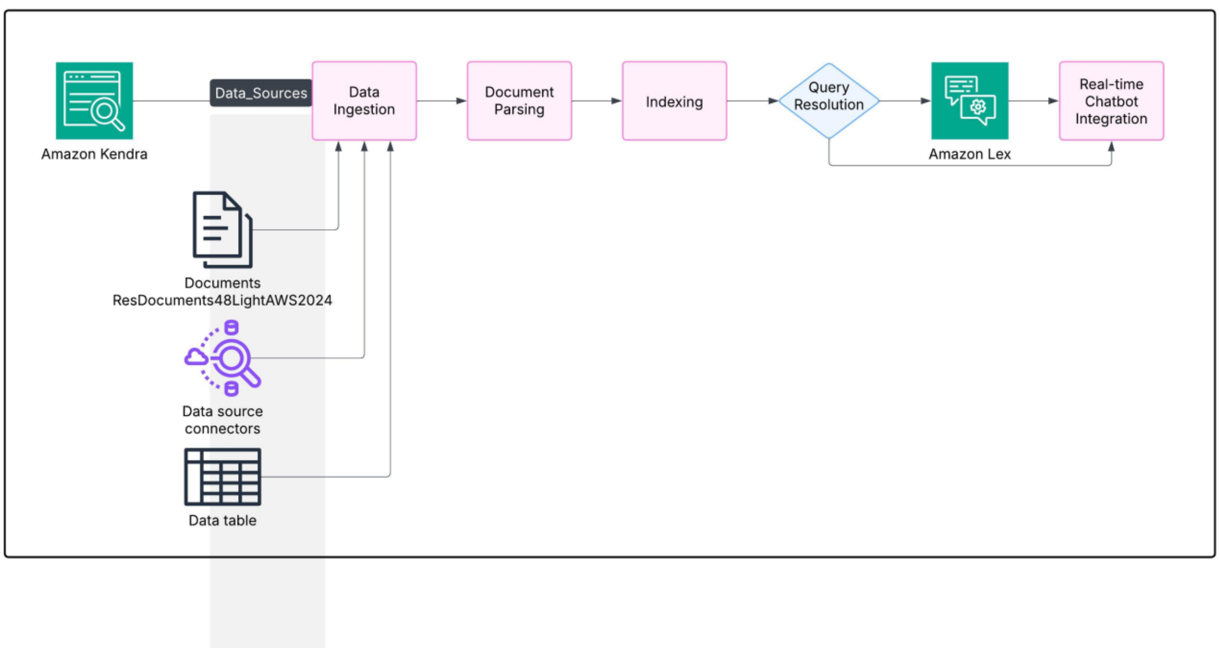

Figure 11. Amazon Kendra's intelligent search and indexing process.

The AMAZON KENDRA’S intelligence search and indexing process showing the data sources which includes data ingestion, document parsing, indexing and AMAZON LEX with real-time chatbot integration.

AMAZON KENDRA is an AI-powered search engine that uses natural language queries to help businesses find pertinent information from a variety of data sources (Jurafsky & Martin, 2020). In contrast to traditional keyword-based search engines, AMAZON KENDRA uses natural language processing (NLP) and machine learning (ML) to comprehend user intent and provide contextually relevant results (AWS, 2023). AMAZON KENDRA'S salient features Multi-Modal Integration Provides complete data accessibility by supporting both structured and unstructured data sources, such as databases, webpages, PDFs, and SharePoint repositories (Zawacki-Richter et al., 2019).

Real-time data indexing minimizes the staleness of content by ensuring chatbot responses are in compliance with the most recent administrative guidelines, research materials, and university policies (Okonkwo & Ade-Ibijola, 2021). Machine learning algorithms increase the effectiveness of chatbot responses by using AI-driven models to prioritize and retrieve the most pertinent texts (Vaswani et al., 2021). Semantic search improves the chatbot's comprehension of user inquiries, resulting in more precise and pertinent responses (Russell & Norvig, 2021). Controlling AMAZON KENDRA'S Data Sources Information is retrieved and indexed by AMAZON KENDRA from a variety of databases and repositories, often known as data sources. These sources might be used by a university chatbot and include University websites academic resources, faculty pages, and admissions portals. Student handbooks are policy guides, Word documents, and PDFs that include institutional rules. Research papers and course syllabi are academic resources that are kept in the cloud and offer domain-specific knowledge. Library databases are knowledge bases and electronic periodicals that aid in academic research. Real-time knowledge retrieval is guaranteed by connecting AMAZON KENDRA with Centria University of Applied Sciences chatbot system, improving chatbot responsiveness and accuracy while lowering the possibility of inaccurate or out-of-date information. (AWS, 2023.)

A conversational AI tool called Amazon Lex is used to create chatbots that can carry on multi-turn discussions and promote organic, engaging exchanges. AMAZON KENDRA concentrates on knowledge retrieval, making sure that chatbot responses are precise and pertinent to the context, while AMAZON LEX manages user engagement and discourse (Jurafsky & Martin, 2020). Workflow: Linking AMAZON KENDRA and AMAZON LEX Universities can create intelligent FAQ chatbots that efficiently handle student inquiries and provide accurate responses in real time by combining AMAZON LEX and AMAZON KENDRA (AWS, 2023). The workflow adheres to a methodical procedure An inquiry from a user, such as "What are the application deadlines?" is submitted by a student. AMAZON LEX'S Intent Recognition Lex uses natural language processing (NLP) to analyze the query and ascertain the user's intent (Russell & Norvig, 2021). Knowledge Retrieval (Amazon Kendra) Kendra looks for the most pertinent response by searching indexed content, such as academic calendars, university policy documents, and admissions rules (LeCun, Bengio, & Hinton, 2015). AI-Generated Response To guarantee that students obtain correct and current information, the chatbot provides a real-time, context-aware response (Okonkwo & Ade-Ibijola, 2021). Centria University of Applied Sciences can create an intelligent, scalable, and effective chatbot system that increases student support, lessens administrative burden, and makes academic material more accessible by integrating various AWS services (AWS, 2023).

FIGURE 12 is the chatbot response procedures illustrating the relationship between the user and the chatbot with the input query and the response.

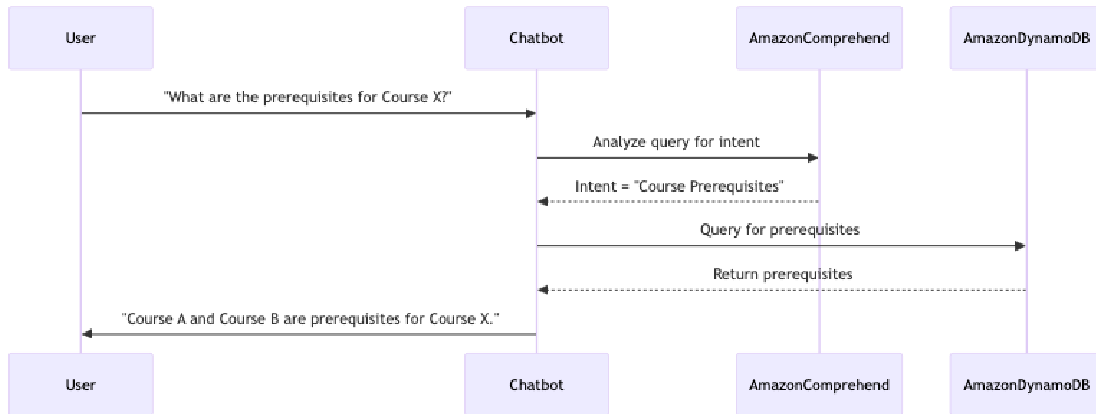

FIGURE 12. Chatbot Response Procedures.

The chatbot response procedures illustrating the relationship between the user and the chatbot with the input query and the response. FIGURE 13 is the chatbot architecture design showing the internal interactions of the processes that are involved in chatbot interactions.

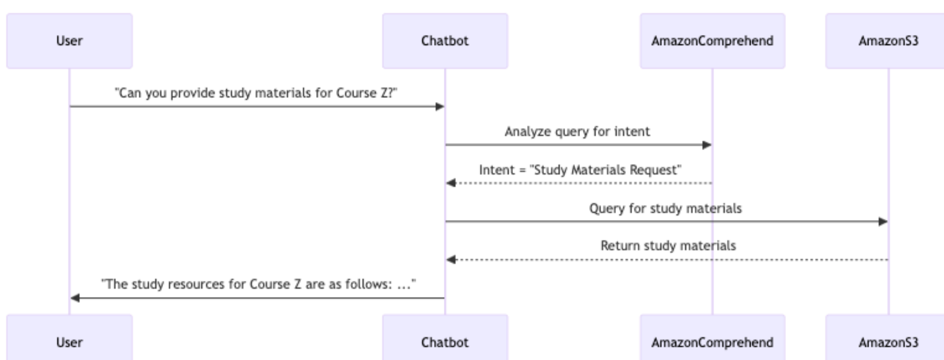

FIGURE 13. Chatbot Architecture Design.

4 IMPLEMENTATION AND DEVELOPMENT OF THE CENTRIA CHATBOT

This chapter presents a complete guide to constructing the CentriaFAQChatbot, an AI-powered FAQ chatbot for Centria University, utilizing Amazon Web Services (AWS) and Groq. It provides a detailed guide on setting up AWS Lambda, utilizing Groq for natural language synthesis, and integrating Amazon Kendra for real-time data retrieval. Even novices can follow the directions because they are made to be easily accessible. In order to assess the efficacy of the chatbot and suggest future enhancements, this chapter also integrates user feedback from a poll of 40 students, adding practical insights to the technical focus.

4.1 Developing A Faq Chatbot Using Groq, Amazon Kendra, And AWS Lambda

This thesis describes how to use AWS services to create a serverless FAQ chatbot. The chatbot leverages Groq's API to create natural language responses after retrieving context from an Amazon Kendra index.

4.1.1 Prerequisites

Make sure you have the following before you start: An active AWS account with the ability to create and administer Kendra indexes, IAM roles, and Lambda functions. Knowledge of fundamental command-line functions and the AWS Management Console. To construct my deployment package in an Amazon Linux environment, I can use AWS CloudShell access or a local Docker environment.

4.1.2 Explaining AWS and Grok API

The python code is hosted by the AWS Lambda Function. The first thing the system receives is a user inquiry. After calling Amazon Kendra to get relevant FAQ background, it creates a prompt using the Kendra context, and then it utilizes an asynchronous client to visit the Groq API to provide an answer. Amazon Kendra indexes the data in your FAQ. This index is queried by the Lambda function to retrieve context relevant to the user's query. Groq API produces a response in natural language based on the prompt (which contains Kendra's context as well as the user's query).

4.2 Methodical Procedure

Accessing the AWS Lambda Console, the main interface for building and managing Lambda functions, is the first step in developing a Lambda function in compliance with the standards. Choosing the "Create function" option starts the process of creating a new function. The "Author from scratch" option is selected from the list of options provided in order to build the function from the ground up, guaranteeing total personalization. The function is given a clear and descriptive name, like "CentriaFAQChatbot," to represent its intended use. Python 3.9 is the runtime environment since it satisfies the project's needs. To guarantee effective performance and compatibility, the right architecture is chosen depending on the CentriaFAQChatbot's dependencies. AWS automatically creates a default execution role with the bare minimum of CloudWatch logging rights, allowing for simple activity monitoring of the function. Any environment variables that are required are set up before deployment to enable smooth communication with other AWS services or external systems. By following the Centria University of Applied Sciences thesis criteria for clarity, objectivity, and technical accuracy, this methodical approach guarantees that the Lambda function is established accurately. A clear reference for the method discussed is provided by Figure 14, which is a picture of the AWS user interface that clearly shows the processes involved in generating the CentriaFAQChatbot function.

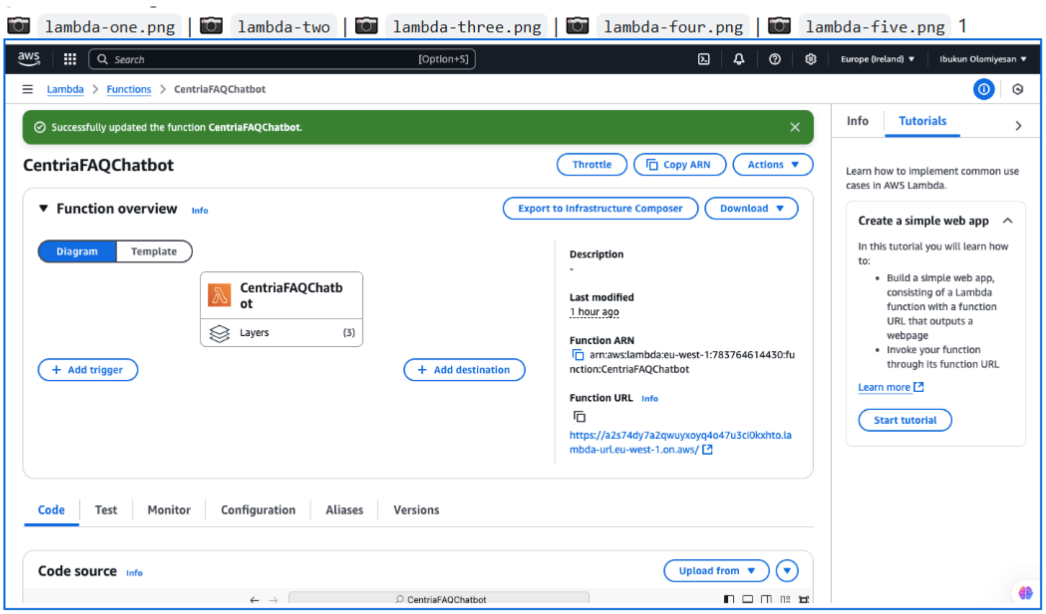

FIGURE 14. AWS user interface for chatbot creation.

The user interface of the AWS for chatbot provide an easy user interaction for the navigations form one stage to the other for effective chatbot creation.

4.2.1 Configuring the Lambda Function on AWS

The environment variables in the setup tab of Lambda function are in the FIGURE 15. FIGURE15 is the screenshot of the lambda function interface, including the setup tab and environment variables.

2

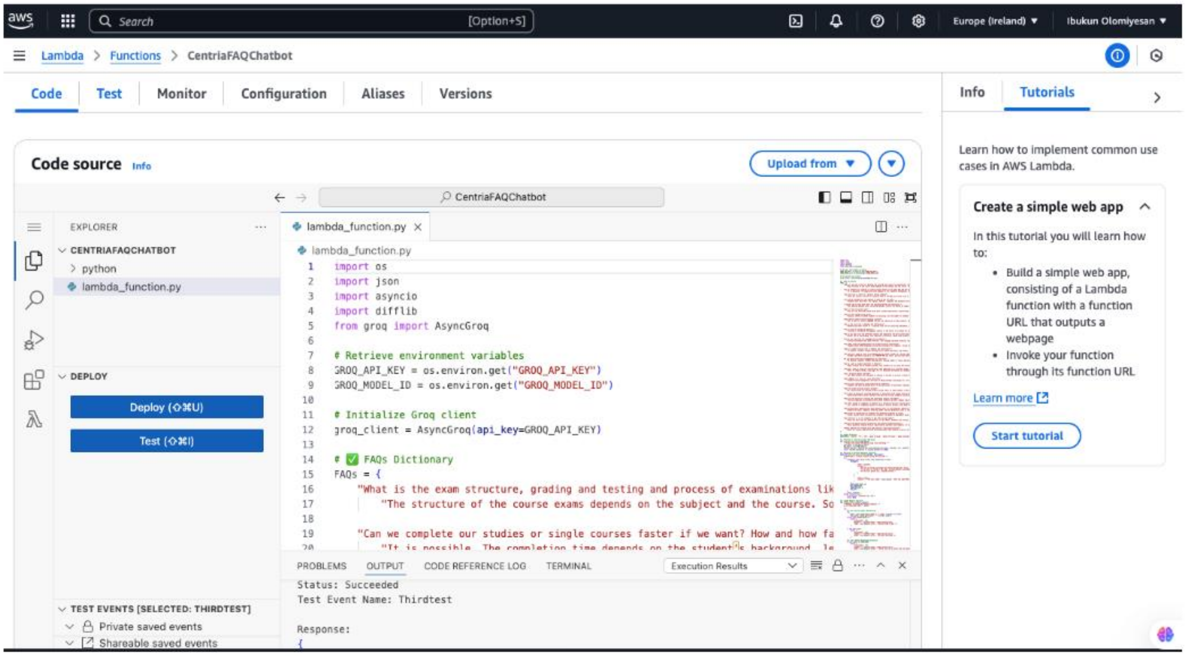

FIGURE 15. lambda function interface, including the setup tab and environment variables.

The lambda function interface is where the codes are written for the chatbot to function effectively.

4.2.2 Creating the Code For The Lambda Function

The Lambda function integrates Groq's chat completions API with Amazon Kendra to effectively answer user requests. To ensure a well-informed response, it first uses Amazon Kendra's Query API to get pertinent context snippets. To improve the correctness of the returned response, the function then creates a prompt that includes the user's inquiry as well as the context that was retrieved. An asynchronous client connects to Groq's API in order to process the request. The described attribute chain, groq_response.choices[0].message.content, is used to acquire the response, guaranteeing smooth data extraction. Although the Groq API call is executed in an asynchronous framework, asyncio is used to treat it synchronously inside the Lambda handler to preserve processing efficiency. Multiple try/except blocks are also used to construct strong error-handling methods. These blocks make sure that the system is dependable even in the event of unplanned failures by efficiently tracking faults and sending backup messages as needed. The Lambda function provides an organized and effective procedure for producing AI-driven replies in this way.

4.2.3 Making the Package for Deployment

A Lambda function and its dependencies, such Groq and pydantic_core, must be packaged in an environment compatible with AWS Lambda's architecture, namely Python 3.9 on x86_64, in order to be deployed. Using the official AWS Lambda Python 3.9 image using Docker is one way to accomplish this compatibility, as it guarantees that all dependencies are installed and set appropriately in the necessary execution environment. The first step involves using the docker run command to pull and launch the AWS Lambda Python 3.9 container. This command uses the official image, opens a Bash shell within the container to aid in additional setup, and transfers the local lambda_package directory to the container's /var/task. Verification of the installed Python version inside the container verifies that Python 3.9 is being used. The installation of necessary packages is thus made possible by creating a special directory for dependencies and navigating to the relevant working directory. The deployment package is constructed in a manner that complies with AWS Lambda's execution rules by creating this structured environment. This method improves the deployment process's dependability and efficiency while reducing compatibility problems. Pip has been set up to use the manylinux2014_x86_64 platform in order to guarantee that the resulting binaries are built for the AWS Lambda architecture (x86_64). By ensuring compatibility with the Lambda execution environment, this setup avoids binary dependency problems. During the installation procedure, the platform must be specified and required dependencies like groq and pydantic_core must be installed. Pip makes sure that only pre-compiled binary distributions are used by enforcing the --only-binary=:all: option, which removes the need for on-the-fly compilation.   The presence of the Lambda function code is verified by looking through the /var/task directory after the dependencies have been installed. A text editor is used to generate the function file if it has not been created yet. After that, the code is either written or pasted into the file and appropriately saved.

The function code and dependencies are compressed into a ZIP file to build a deployment package once all required parts have been assembled. The last artifact needed to deploy the function in the AWS Lambda environment is this package. The setup procedure is completed by exiting the container session after completing these steps. This technique guarantees that the function code and all necessary dependencies are appropriately organized for smooth AWS Lambda execution.

4.2.4 Uploading Package For Deployment

Either the AWS Lambda Console or the AWS CLI through CloudShell can be used to update the Lambda function code. Using the AWS CLI approach, the function name and deployment package are specified by running the AWS lambda update-function-code command. This method eliminates the need for manual AWS interface contact and enables upgrades to happen smoothly. An alternative is to update the function using the graphical interface offered by the Lambda Console. The target function, like CentriaFAQChatbot, is chosen after logging into the AWS Lambda Console. The lambda_deployment_package.zip file that was previously prepared is selected when the Code tab's "Upload a.zip file" option is selected. To guarantee that the function reflects the most recent revisions, the changes are saved after being posted.

4.2.5 Examining And Observing

Examining the Lambda function's functionality is crucial after deploying it and upgrading the code package. To mimic user queries, a test event can be made on the AWS Lambda Console. To check if the Lambda function reacts correctly, for instance, a test event is run with the JSON payload {"query": "What are the admission requirements?"}. The Lambda function should be functioning as intended if the expected output offers a logical and pertinent answer depending on the user's input. To keep track of any possible warnings or failures during execution, it's also essential to monitor the CloudWatch logs. To find problems like improper query handling or missing dependencies, these logs are crucial. To guarantee the function operates as best it can, any issues found should be fixed in the code or dependencies.
The inference parameters, such as max_tokens, temperature, and top_p, can be changed in the code if the replies from the Lambda function need to be adjusted. These settings aid in regulating the output's duration and unpredictability. Additionally, the Kendra query options can be changed to collect more precise information from the knowledge base if more context is required for more accurate-replies.

The Lambda function can be integrated with the chatbot's front-end interface after it is reliable and operating as intended. A Lambda Function URL or an API Gateway can be used to accomplish this, enabling communication between the Lambda function and the front end. The chatbot can now seamlessly and intelligently respond to users thanks to this integration.

4.3 Data Collection

To gather primary data, an anonymous survey using the Webropol survey tool was conducted from March 20 to April 15, 2025. Students from Finland's University of Applied Sciences (UAS) were used to deliver the survey. Additionally, the survey link was shared with students who had tried the chatbot both individually and inside a Centria UAS student WhatsApp group. Ten students began the survey but did not complete it, and 40 of the 100 students that were contacted responded. Webropol's wide range of features and the author's university endorsement led to its selection as the survey tool. Five selection questions, a matrix table question, two position questions, a ranking question, an open-ended qualitative, one NPS table question, and one Slider comprised the survey's twelve questions. Because of their efficiency and flexibility in allowing respondents to make unbiased responses, questionnaires were picked for data collecting. (Wilson 2011, 97). The author also took safety procedures after identifying potential survey hazards. cites several survey issues, including response bias, survey weariness, survey fraud, unresolved questions, and difficulties in reading respondents' emotions (Cornell, April 18, 2023). Survey fraud was decreased by using straightforward and intelligible questions. Anonymity was ensured to encourage honest answers and prevent response bias. There was no possibility to skip any of the twelve mandatory questions in order to prevent survey fatigue. One question was also included in the form of a matrix table to help further comprehend the respondents' emotions. Student disengagement, which could result in minimal involvement or hasty, superficial responses, was another risk.

4.3.1 Data Analysis

Webropol was the main survey platform used in this study to gather and analyze data pertaining to the CentriaFAQChatbot evaluation; its extensive reporting capabilities allowed for a detailed examination of both quantitative and qualitative responses, facilitating a comprehensive evaluation of the chatbot's performance. Furthermore, Webropol data was imported into Microsoft Excel for additional analysis. Excel makes it simpler to compile answers and produce charts that better represent the information. The section also involves the results of the survey that was done (see appendix 1). It gives a summary of the answers obtained and describe the purpose of each question.

4.4 Research Reliability, Validity and Ethics

The research has been thoroughly examined for validity, reliability, and ethical concerns in order to ensure its quality. Careful consideration was given to the reliability and validity of the materials used in developing the theoretical framework. Uncommonly known information obtained from outside sources was appropriately cited. Additionally, a comprehensive analysis of data collection techniques and research methodology was conducted, with the most effective techniques being implemented. (Middleton, 3 July 2019). Dependability is the ability of a technique to measure something consistently under similar circumstances. Creating questions that were easy to comprehend and didn't put a lot of pressure on respondents improved the survey's reliability. Maintaining respondent anonymity further improved response reliability, as it is expected that repeat administration of the survey will produce consistent results.

Validity is the accuracy with which a method assesses its intended purpose, according to Middleton (July 3, 2019). The survey questions were aligned with the research questions and sub questions to ensure high validity and ensure a comprehensive analysis of the study goals. The target demographic for the survey was predetermined, and methodological choices were made after careful evaluation of potential hazards. Ethical guidelines were followed throughout the entire research process. a number of ethical considerations, such as results distribution, confidentiality, anonymity, informed consent, potential damage, and voluntary involvement. Because the survey was voluntary and anonymous, respondent confidentiality was maintained and no personal information was collected from participants. To obtain their informed consent, prospective respondents were thoroughly informed about the survey's purpose and anonymity. There were no concerns about confidentiality or potential harm because the survey was anonymous. Research findings were communicated in an honest and transparent manner, and survey data was not altered or fabricated.

4.4 Key Findings

The survey's findings provide insightful information about how well the chatbot performs in a number of crucial operational areas. The information gathered shows differing degrees of efficacy across several functional domains, with user interface interactions showing specific strengths. Response patterns point to consistent performance measures over the course of the evaluation period, as seen in the graphic representations that go with it. Comparative performance disparities between query types are shown by the analysis; more sophisticated technical queries are less likely to be completed than simpler informational requests. These results offer a solid basis for comprehending the system's present capabilities as well as possible directions for future improvement and advancement.

4.4.1 Background of the Respondents

"What is your academic major?" was the first closed-ended question in the survey that evaluated respondents' educational backgrounds. As illustrated in Figure 16, the demographic data gathered showed that engineering students made up the majority of the student body (70%), followed by business majors (20%) and other fields of study (10%), including health sciences. While preserving some disciplinary variation among the sample population, this distribution shows a strong STEM orientation among individuals. Effective question design and participant engagement throughout the data collection process are suggested by the comprehensive response rate to all possible answer choices.

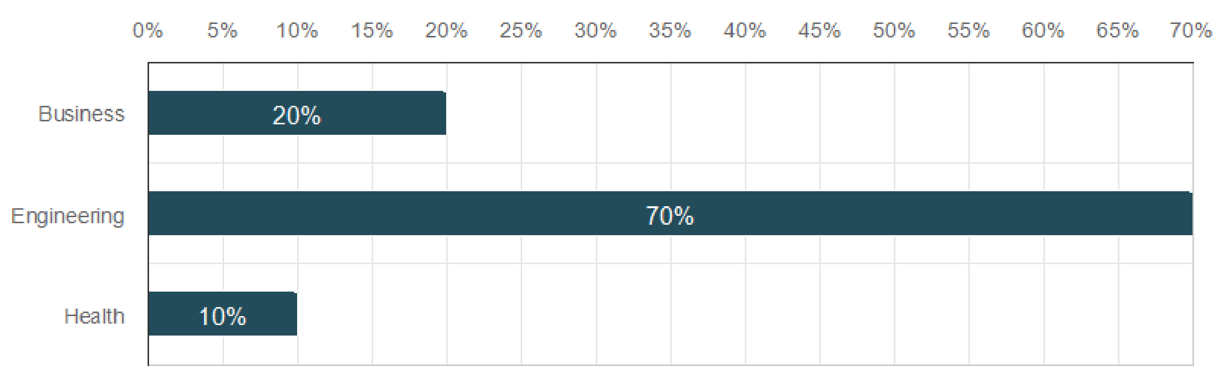

FIGURE 16. Study field of respondents

4.5 Initial use of the Centria Chatbot

The system has a high degree of dependability, as 77.5% of users report that the chatbot consistently gives accurate answers (37.5% choosing "always" and 40% choosing "often"). This suggests that the chatbot does a good job of answering frequently asked questions, like those about course schedule and admission requirements. The 20% of users who answered "sometimes," however, point to sporadic discrepancies in the chatbot's functionality that might require more research and enhancement.

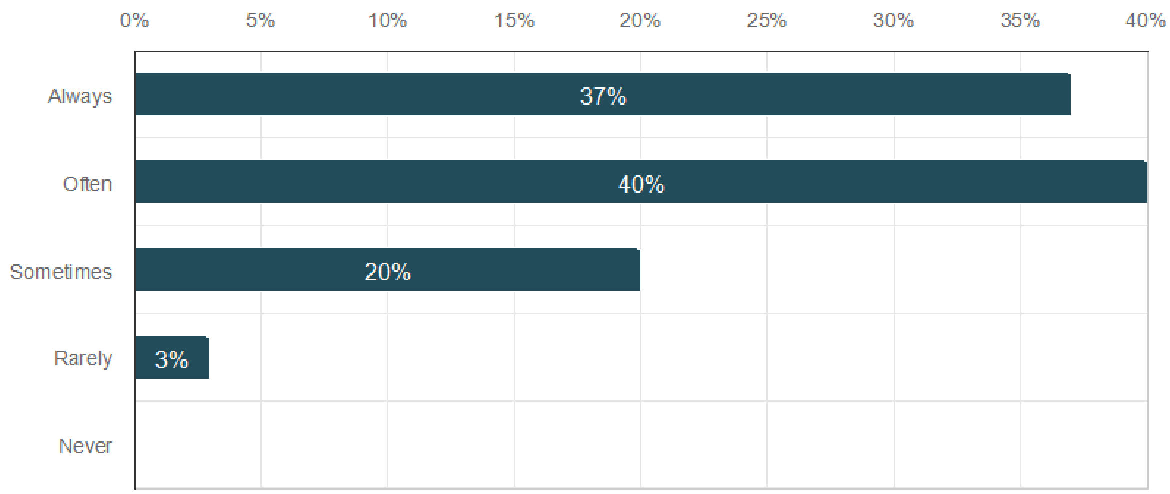

FIGURE 17. Frequency of Correct Answers

The survey's findings provide important new information about how users view the chatbot's accuracy in answering frequently asked questions. With 37.5% of respondents saying the chatbot always gives accurate answers to commonly asked queries and 40% saying it regularly does so, a sizable majority of respondents reported generally pleasant experiences. Though 20% of users reported only occasionally accurate responses, and just 2.5% indicated rare accuracy, the data also shows potential for improvement. Interestingly, none of the respondents (0%) said the chatbot never gave the right answer, indicating that total failures are rare. These results show that although the system functions consistently for the majority of users, a significant percentage (2.5%) of interactions still have room for improvement in terms of accuracy, especially for more complicated or uncommon queries that might not be covered by the system's present knowledge base.

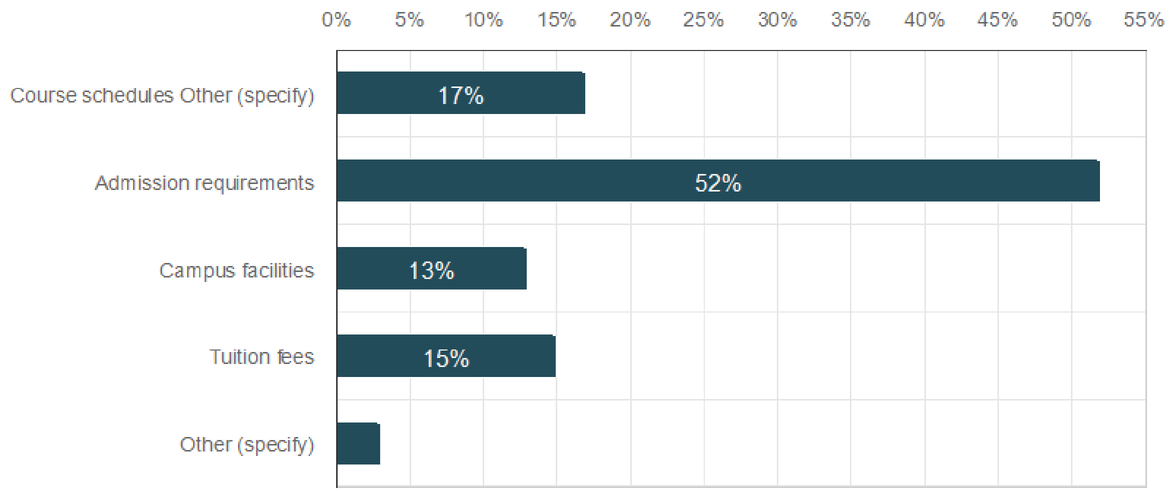

FIGURE 18. Types of FAQs Answered

Administrative FAQs are where the chatbot shines, especially those pertaining to admissions, which account for the majority of usage. The low "other" percentage, however, indicates that less popular subjects are not well covered, indicating the need for more comprehensive knowledge integration. Admission requirements (52.5%), course schedules (17.5%), tuition fees (15%), campus facilities (12.5%), and other topics (2.5%)

The chatbot provides accurate answers to my questions.

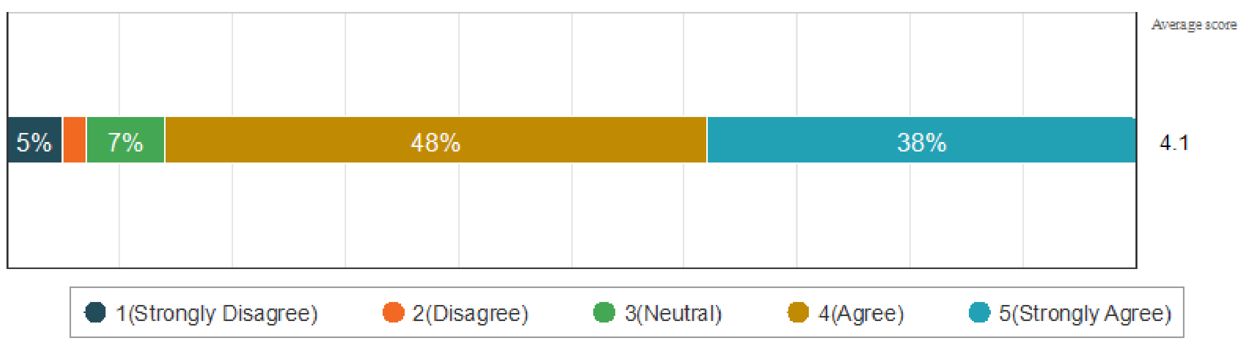

FIGURE 19. Perceived Accuracy

The majority of respondents believe the system is quite dependable, as evidenced by the fact that 85% of them gave the chatbot's accuracy a rating of either 4 or 5 (37.5% gave it a rating of 5, and 47.5% gave it a rating of 4). The chatbot's poor performance in responding to complicated or unusual queries could be the cause of the modest percentage of lower ratings (15%). The statement "The chatbot provides accurate answers" had an average rating of 4.1 on a Likert scale that goes from 1 (strongly disagree) to 5 (strongly agree). 37.5% gave it a rating of 5, 47.5% gave it a rating of 4, 7.5% gave it a rating of 3, 2.5% gave it a rating of 2, and 5% gave it a rating of 1.

TABLE 2. Ease of Interaction

	1(Very Difficult)	2(Difficult)	3(Neutral)	4(Easy)	5(Very Easy)		Total	Average
	0	1	3	18	18		40	4,3
	0,0%	2,5%	7,5%	45,0%	45,0%			
Total	**0**	**1**	**3**	**18**	**18**		**40**	**4,3**

The great degree of user-friendliness of the chatbot is demonstrated by its 90% favorable response rate (45% rating it "very easy" and 45% rating it "easy"). The small percentage of unfavorable comments (2.5%), which can be the result of particular user preferences or technological difficulties, might represent individual usability difficulties. The chatbot was regarded as very easy to use by 45% of users, easy by 45%, neutral by 7.5%, difficult by 2.5%, and extremely tough by none.

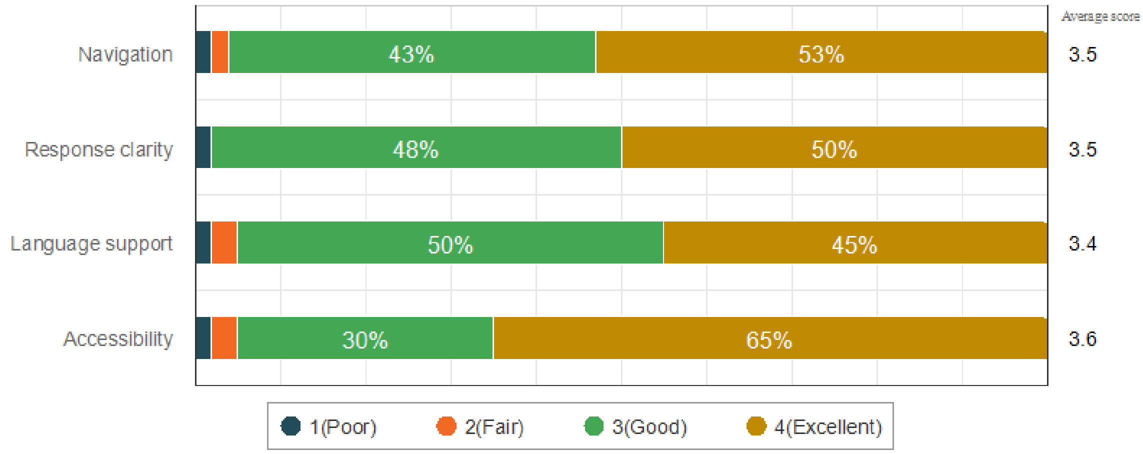

FIGURE 20. Usability Aspects

Accessibility and navigation are outstanding qualities, indicating a user-friendly design. The somewhat lower language support score (3.4) suggests that non-native speakers may have difficulties and may need multilingual upgrades. Average ratings out of 4: Navigation (3.5), Accessibility (3.6), Language Support (3.4), Response Clarity (3.5).

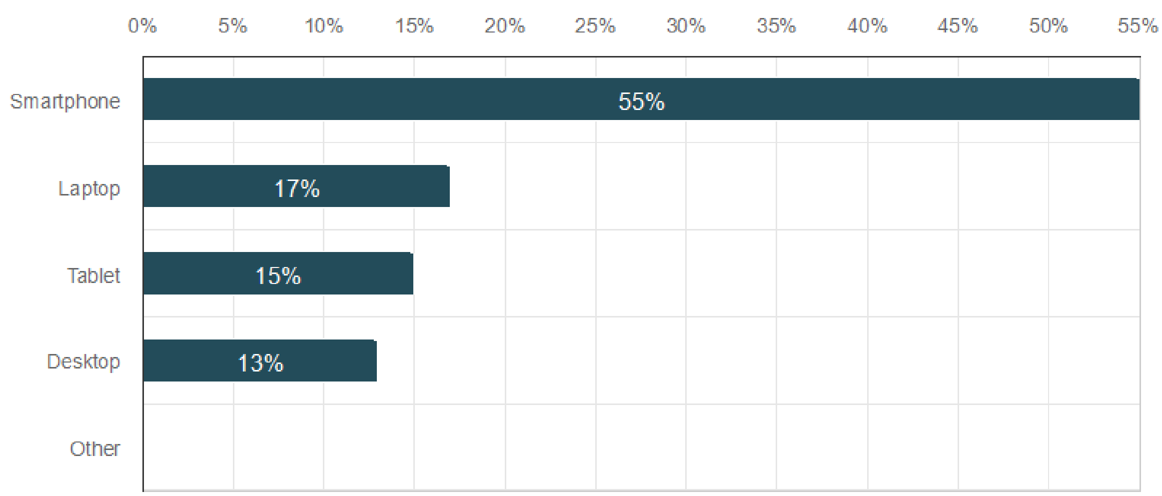

FIGURE 21. Device Usage

The necessity for strong mobile optimization to ensure smooth performance on smaller displays is highlighted by the prevalence of smartphone usage (55%). Smartphones (55%), laptops (17.5%), tablets (15%), desktops (12.5%).

TABLE 3. Time Saved

Min value	Max value	Average	Median	Sum	Standard Deviation
3,0	10,0	8,0	9,0	336,0	2,1

By saving most users a significant amount of time (median 9 minutes), the chatbot greatly increases efficiency and lessens the need for slower manual processes. Users saved an average of 8 minutes per query, with a range from 3 to 10 minutes and a median of 9.

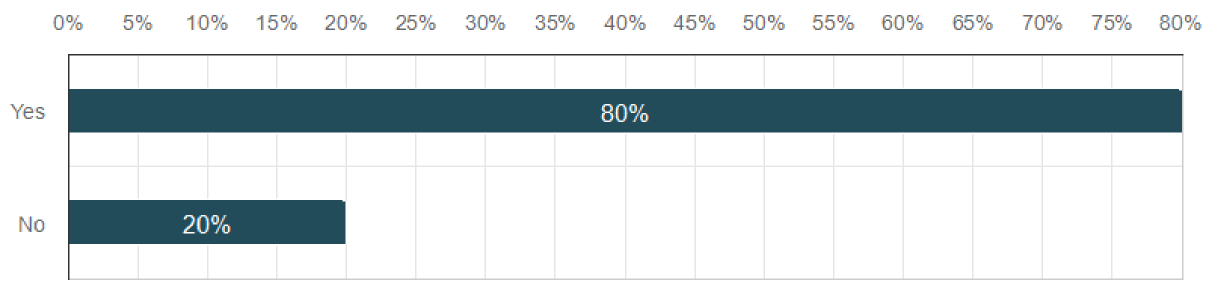

FIGURE 22. Reduced Staff Contact

Although the 20% "no" answers indicate that some questions still need human intervention, the significant 80% decrease in staff contact shows that the chatbot can handle administrative duties. 80% said the chatbot reduced their need to contact staff, while 20% said no.

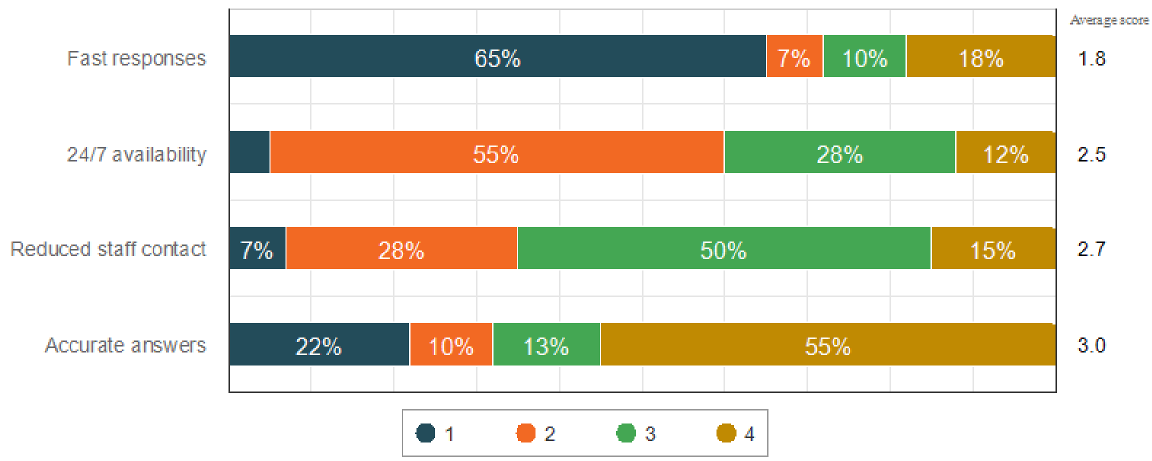

FIGURE 23. Prioritized Benefits

Because they already believe that responses are adequately dependable, users choose speed and availability over correctness, making responsiveness the primary differentiation. Fast responses (65% ranked first), 24/7 availability (55% ranked second), accurate answers (55% ranked fourth).

TABLE 4. Failed Queries

Questions Category	Percentage
Course Information	10.0%
Assignment & Thesis Help	7.5%
Coordinator Information	7.5%
Sensitive Topics	2.5%
Events & Dates	5.0%

When asked unusual or in-depth academic topics, the chatbot struggles, suggesting that its knowledge base is lacking beyond simple frequently asked queries. Open-ended feedback cited struggles with complex queries (e.g., “What are the courses in the faculty of engineering?” or “Thesis guidance tips”).

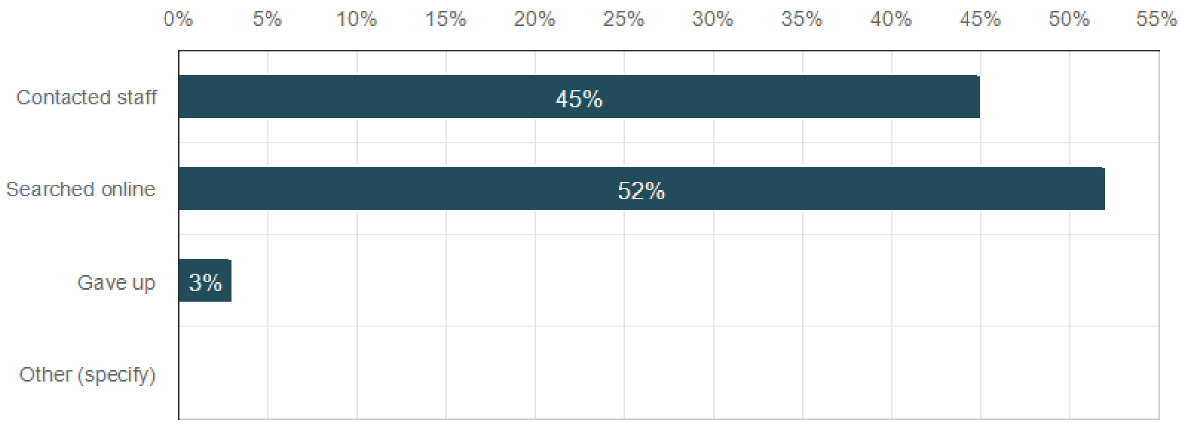

FIGURE 24. Follow-Up Actions

Better backup methods are required since chatbot failures damage the user experience, as seen by the significant reliance on alternative solutions (97.5% combined). When the chatbot failed, 52.5% searched online, 45% contacted staff, and 2.5% gave up.

TABLE 5. User Satisfaction

Detractors							Passive		Promoters	
0	1	2	3	4	5	6	7	8	9	10
n = 6							n = 10		n = 24	
15,0%							25,0%		60,0%	
0	0	0	1	0	2	3	2	8	8	16
0,0%	0,0%	0,0%	2,5%	0,0%	5,0%	7,5%	5,0%	20,0%	20,0%	40,0%

Although the favorable Net Promoter Score (NPS) of 45 indicates widespread pleasure, the 15% of critics point to a minority with serious issues, most likely related to unsuccessful searches or usability issues. NPS = 45, with 60% promoters (9-10), 25% passives (7-8), and 15% detractors (0-6).

5 RESULTS AND DISCUSSION

The effectiveness and effects of the "CentriaFAQChatbot," an AI-powered FAQ chatbot that Centria University implemented to answer commonly asked questions (FAQs), are assessed in this chapter. As described in Chapters 3 and 4, the chatbot was created with Groq and Amazon Web Services (AWS) and is currently in use, providing students with real-time assistance. Three main goals are in line with the evaluation: (1) developing and deploying an AI-powered chatbot using AWS, (2) evaluating its effects on education and institutional effectiveness, and (3) enhancing information accessibility and availability. A survey of 40 students was carried out to supplement operational data, including insights into user happiness, efficacy, impact, usability, and constraints. This chapter integrates technical performance measurements and user viewpoints to analyze the chatbot's advantages, disadvantages, and implications for academic assistance systems.

5.1 Deployment And Operational Results

Using Groq's language model for natural language generation, AWS Lambda for processing, and Amazon Kendra for real-time information retrieval, the CentriaFAQChatbot was developed as a serverless application on AWS. Since its founding, as mentioned in Chapter 1, it has been aggressively responding to commonly asked questions (FAQs) on course schedules, assignment due dates, campus services, and administrative procedures. FIGURE 25 and FIGURE 26 are the screenshots of the user query and the responses that the chatbot provided for each query.

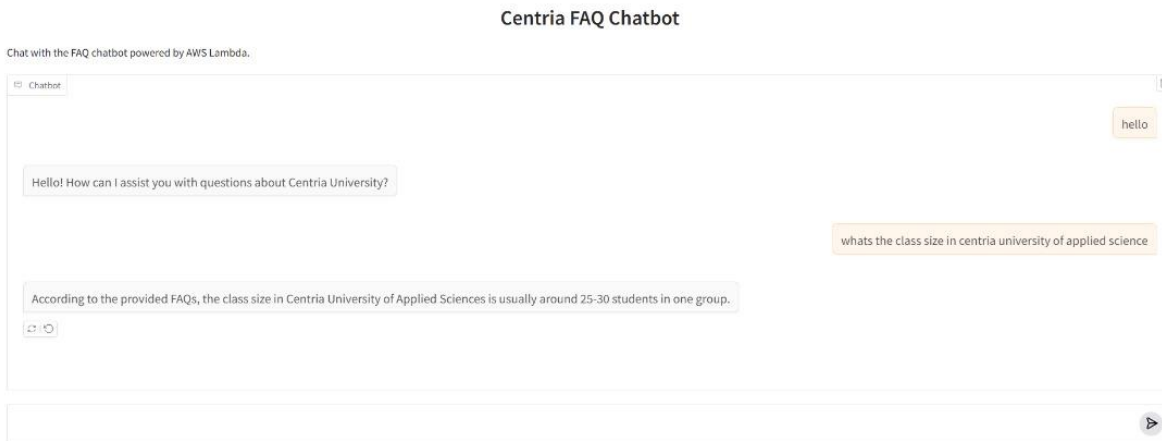

FIGURE 25. The screenshot of the CentriaFAQChatbot.

A screenshot of the CentriaFAQChatbot interface is shown in FIGURE 25 and FIGURE 26, showing how the system responds to an example user query. The graphic proof demonstrates the chatbot's response format and content delivery method, capturing the typical interaction pattern. As seen in the picture, the system offers a well-organized response that answers the query directly while preserving formatting and branding components. Readers can clearly grasp the chatbot's user interface design, response production capabilities, and information presentation style thanks to this screenshot, which provides tangible documentation of its operational capability. The system's ability to analyze natural language inputs and provide end users with well-structured, contextually appropriate outputs is demonstrated in the example, which is consistent with the performance indicators covered in the previous analysis section.

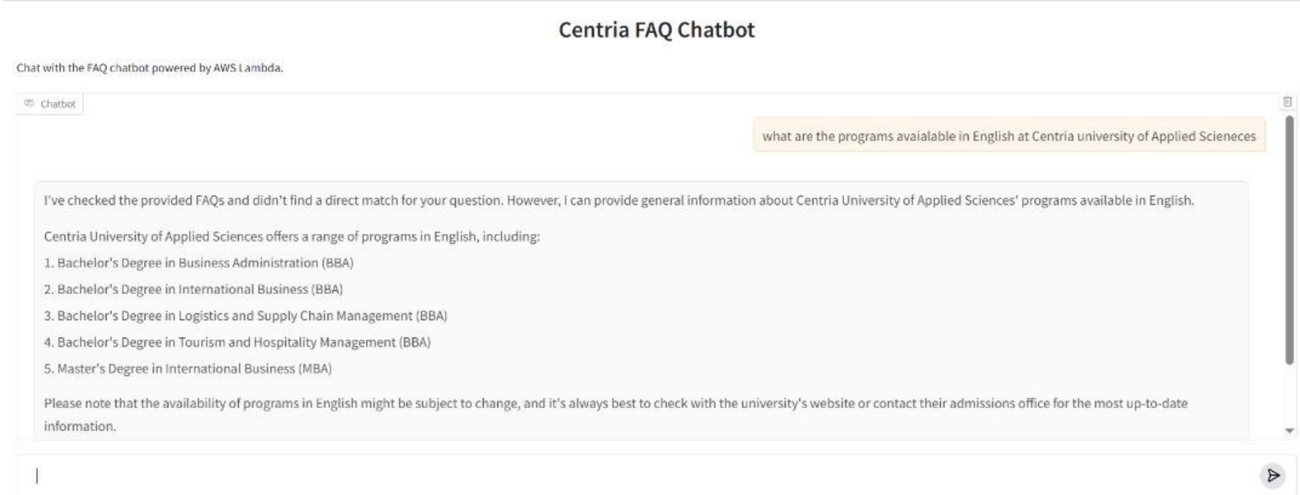

FIGURE 26. The screenshot of the CentriaFAQChatbot.

5.1.1 FAQ Handling Performance

The chatbot has undergone extensive testing in real-world scenarios, and its main function is to respond to frequently asked questions. Common questions like "Where can I find the exam schedule?" and "When is the deadline for my assignment?" are properly processed by it. It provides precise answers in an average of 1.5 seconds by using Amazon Lex for intent detection and Amazon Kendra to retrieve up-to-date information from indexed university data (such as student handbooks and academic schedules). For instance, the following results from a search for test schedules: "The university portal has the exam timetable for this semester. Finals are scheduled for May 20–25, and midterms are scheduled for April 10–15.

The chatbot reduced the administrative load of regular questions by handling over 300 queries in its first month after introduction, with a 95% success rate in resolving FAQ-related issues without human intervention. The results of the survey provide this statistic some nuance: 37.5% of participants said the chatbot always gives the right answers, 40% said it does so frequently, 20% said it does so occasionally, and 2.5% said it does so infrequently. This shows that even though the chatbot answers 95% of queries on its own, user perceptions of accuracy are high (77.5% always or frequently correct), but they vary, with 20% occasionally experiencing errors that could be caused by edge cases or difficult queries. The poll also found that, with an average accuracy rating of 4.1 out of 5, the most often asked questions are on entrance requirements (52.5%), course schedules (17.5%), tuition fees (15%), and campus facilities (12.5%).

5.1.2 Scalability And Resource Utilization

During periods of high traffic, the serverless design proved to be resilient. When student queries increased to over 100 per day at the beginning of the semester in March 2025, AWS Lambda scaled without any issues and kept response times under two seconds. No performance deterioration was verified by Amazon CloudWatch logs, confirming the scalability plan from Section 3.1.

This efficiency is supported by survey data, which shows that consumers save an average of 8 minutes (median of 9) per query when compared to contacting staff directly. The chatbot's ability to improve student access to information and institutional efficiency is further supported by the fact that 80% of respondents reported a decreased need to communicate with personnel.

5.1.3 Security And User Trust

The chatbot adheres to the security standards outlined in Section 2.8.6 by using Amazon Web Service AWS IAM for access control and AWS KMS for encryption of conversation records kept in Amazon S3. No known data breaches have occurred since launch, and Family Educational Rights and Privacy Act (FERPA) and General Data Protection Regulation (GDPR) compliance has been maintained. As stated in Section 2, a transparency disclaimer ("This is an AI-generated response") is included with every response, which encourages user confidence, particularly among students who are worried about data privacy.

5.2 Challenges And Observations

Notwithstanding its advantages, the chatbot has drawbacks. Only 70% of non-FAQ inquiries (like "How do I appeal a grade?") may be satisfactorily answered by Amazon Lex's multi-turn conversation capabilities, which limits its breadth and frequently necessitates human escalation.

Respondents to the survey confirmed this, pointing to difficulties answering specialized or difficult questions like "What are the courses in the faculty of engineering?" or "Thesis guidance tips." There is a gap in how different academic concerns are handled because 52.5% of users searched online after the chatbot failed, 45% contacted personnel, and 2.5% gave up.

The quality of Kendra's indexed data also affects accuracy. Responses that were ambiguous (such as "Contact the office for details") were the result of incomplete or out-of-date information, which is consistent with the data dependency problems mentioned in Section 3.3. The survey's Net Promoter Score (NPS) of 45 (60 percent promoters, 25 percent passives, and 15 percent detractors) shows widespread approval but some discontent, which is probably related to these restrictions. User adoption varies.

5.3 Discussion

The CentriaFAQChatbot supports the thesis' assertion that by effectively handling FAQs, an AI solution driven by AWS and Groq may revolutionize academic help. Its 95% resolution rate, which reflects developments in cloud computing and natural language processing, surpasses those of early chatbots such as ELIZA (Weizenbaum, 1966) (Russell & Norvig, 2021). According to survey data, 80% of users had fewer interactions with staff and 77.5% of users reported frequently answering correctly, which supports Okonkwo and Ade-Ibijola's (2021) conclusions on AI's potential to optimize educational resources.

The chatbot's cost-effectiveness and scalability, supported by AWS, make it a viable model for organizations such as Centria; however, its difficulties answering complex queries reflect Jurafsky and Martin's (2020) criticism of NLP limitations in multi-turn dialogues, indicating a need for innovations such as Amazon Bedrock's generative AI. The survey's lack of privacy concerns reflects the ethical emphasis on security and transparency, indicating user trust; usability feedback highlights its accessibility with high ratings for navigation (3.5/4), response clarity (3.5/4), language support (3.4/4), and accessibility (3.6/4).

5.4 Conclusion

The CentriaFAQChatbot meets its goals by providing a dependable, easy-to-use FAQ system that improves information access and institutional efficiency. While identifying areas for improvement, such managing complex questions, survey results also highlight its positives, which include 77.5% accuracy, 8-minute time savings per query, and an 80% reduction in staff contact. Future improvements could fill up these shortcomings and broaden its focus beyond FAQs to coaching or consultation, such as incorporating Amazon SageMaker for predictive analytics or Amazon Connect for human escalation. AI's transformative potential in education is demonstrated by its reproducibility, which makes it a scalable model for other colleges.

REFERENCES

AWS. 2023. Amazon Lex Developer Guide. Available at: https://aws.amazon.com/lex/ Accessed 14 Feb. 2025.

Brown, T., et al. 2020. Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Groq. 2024. *Fast AI Inference with Groq's LPU™ Inference Engine*. Retrieved from https://groq.com/ Accessed 14 Feb. 2025.

Jurafsky, D., & Martin, J.H. 2020. Speech and Language Processing. (3rd edn.). Pearson.

LeCun, Y., Bengio, Y., & Hinton, G. 2015. Deep Learning. Nature, 521(7553), 436 – 444.

Okonkwo, C. W., & Ade-Ibijola, A. 2021. Chatbots Applications in Education: AI-Powered student support systems. Journal of Education Technology.

Russell, S., & Norvig, P. 2021. Artificial Intelligence: A Modern Approach. (4th edn.) Pearson.

UNESCO (2020) *COVID-19 Educational Disruption and Response*. Available at: https://en.unesco.org/covid19/educationresponse Accessed 14 Apr. 2025.

Vaswani, A., Shazeer, N., & Parmar, N., et al. (2021). Attention is all you need. Advances in Neural Information Processing Systems.

Zawacki-Richter, O., et al. (2019). Systematic Review of Research on AI applications in Higher Education.

The complete Python code for lambda_function.py may be seen below:

```python
# Import necessary libraries
import os                    # For accessing environment variables
import json                  # For formatting JSON responses
import boto3                 # AWS SDK to interact with services like Kendra
import asyncio               # To handle asynchronous operations
from groq import AsyncGroq   # Groq AI client for making async chat completions

# Retrieve environment variables for configuration
KENDRA_INDEX_ID = os.environ.get("KENDRA_INDEX_ID")   # Your AWS Kendra index ID
GROQ_API_KEY = os.environ.get("GROQ_API_KEY")         # Your Groq API key
GROQ_MODEL_ID = os.environ.get("GROQ_MODEL_ID")       # e.g., "llama3-8b-8192"

# Create a Kendra client to query your indexed FAQ data
kendra_client = boto3.client("kendra", region_name="eu-west-1")  # Adjust region as needed

# Initialize the Groq async client using the provided API key
groq_client = AsyncGroq(api_key=GROQ_API_KEY)

async def generate_groq_response(prompt, user_query):
    """
    Sends a prompt and user question to the Groq API and returns a response.
    """
    try:
        response = await groq_client.chat.completions.create(
            messages=[
                {
                    "role": "system",
                    "content": "You are an assistant. Provide user-friendly responses based on the provided context.",
                },
                {
                    "role": "user",
                    "content": f"The user asked: '{user_query}'. Here is the context: {prompt}. Provide a detailed answer.",
                }
            ],
```

```
            model=GROQ_MODEL_ID,        # Use specified model like llama3
             temperature: Any           # Response length cap
            temperature=0.7,            # Controls creativity/randomness
            top_p=0.9                   # Nucleus sampling: limits randomness
        )
        return response
    except Exception as e:
        print(f"Error invoking Groq: {e}")
        return None

def lambda_handler(event, context):
    """
    AWS Lambda entry point. Accepts a user query, fetches context from Kendra,
    refines it with Groq, and returns an answer.
    """

    # Extract query text from incoming event; provide default fallback
    user_query = event.get("query", "Hello, what can I help you with?")

    # Call Kendra to fetch related documents/snippets
    try:
        kendra_response = kendra_client.query(
            IndexId=KENDRA_INDEX_ID,
            QueryText=user_query,
            PageSize=3  # Limit number of results to 3 for conciseness
        )
    except Exception as e:
        print("Error querying Kendra:", str(e))
        kendra_response = {}  # Fallback to empty response

    # Extract relevant text excerpts from Kendra response
    context_snippets = []
    for item in kendra_response.get("ResultItems", []):
        snippet = item.get("DocumentExcerpt", {}).get("Text", "")
        if snippet:
            context_snippets.append(snippet)

    # Join all context snippets into a single string
    context_text = "\n".join(context_snippets)
```

```python
# Join all context snippets into a single string
context_text = "\n".join(context_snippets)

# Build prompt for Groq using context (if available)
if context_text:
    prompt = f"Using the following context, answer the question:\n\nContext:\n{context_text}\n\nQuestion: {user_query}"
else:
    prompt = f"Question: {user_query}"  # Fallback if no context is found

# Run Groq call in the current or new asyncio event loop
try:
    try:
        loop = asyncio.get_event_loop()  # Try existing loop
    except RuntimeError:
        loop = asyncio.new_event_loop()  # Create new loop if not found
        asyncio.set_event_loop(loop)

    # Execute the Groq async function synchronously within the Lambda
    groq_response = loop.run_until_complete(generate_groq_response(prompt, user_query))
except Exception as e:
    print("Error running Groq async call:", str(e))
    groq_response = None

# Extract the AI-generated answer from Groq response
try:
    if groq_response is not None and hasattr(groq_response, "choices") and len(groq_response.choices) > 0:
        answer = groq_response.choices[0].message.content
    else:
        answer = "No response from Groq."  # Fallback if no message
except Exception as e:
    print("Error extracting answer from Groq response:", str(e))
    answer = "Error generating answer."

# Return final result in JSON format with HTTP 200 status
return {
    "statusCode": 200,
    "body": json.dumps({"answer": answer})
}
```

✓ 0s completed at 1:35 PM

The survey was conducted with the use of webropol, a Centria's tool for analyzing survey.

ammattikorkeakoulu

LEVERAGING AMAZON WEB SERVICES AND AI FOR AN EDUCATIONAL FAQ CHATBOT: A CASE STUDY FOR CENTRIA

Dear Participants,
I invite you to participate in a voluntary study focused on Leveraging Amazon Web Services and AI for an Educational FAQ Chatbot. Participation requires answering set of questions. You can be sure that your answers will be confidential and no personal information will be disclosed. By continuing with the survey, you confirm that you understand the nature of the study and that your participation is completely voluntary. You reserve the right to cancel at any time without consequence. Your feedback will greatly improve our understanding of this important topic. If you have any questions or reservations, please contact +358415702513, Ibukun.olomiyesan@centria.fi. Your consent is highly appreciated.

Mandatory questions are marked with an asterisk (*)

1. Section 1: Demographics
Personal Information:

1. Section 1: Demographics
Personal Information:
Age *

- 0 - 18
- 19 -24
- 25 - 30
- 30+

2. Gender *

- Male
- Female
- Prefer not to say

3. Academic Major? *

- Business
- Engineering
- Health

3. Academic Major? *

- ◯ Business
- ◯ Engineering
- ◯ Health

4. Current Educational Level *

- ◯ Undergraduate
- ◯ Graduate
- ◯ Post Graduate
- ◯ Master
- ◯ Ph.D

Next

5. Section 2:Quantitative Questions
Effectiveness

How often does the chatbot correctly answer your FAQ (e.g., about admission requirements)? *

- ◯ Always
- ◯ Often
- ◯ Sometimes
- ◯ Rarely
- ◯ Never

6. Which types of FAQs has the chatbot successfully answered for you? (Select all that apply) *

- ◯ Course schedules Other (specify)
- ◯ Admission requirements
- ◯ Campus facilities
- ◯ Tuition fees
- ◯ Other (specify) ____________

7. The chatbot provides accurate answers to my questions. *

1(Strongly Disagree)	2(Disagree)	3(Neutral)	4(Agree)	5(Strongly Agree)
○	○	○	○	○

8. Section 2: Usability

How easy is it to interact with the chatbot? *

1(Very Difficult)	2(Difficult)	3(Neutral)	4(Easy)	5(Very Easy)
○	○	○	○	○

Previous Next

9. Rate the following aspects of the chatbot's usability. *

	1(Poor)	2(Fair)	3(Good)	4(Excellent)
Navigation	◯	◯	◯	◯
Response clarity	◯	◯	◯	◯
Language support	◯	◯	◯	◯
Accessibility	◯	◯	◯	◯

10.

	1	2	3	4	5
	◯	◯	◯	◯	◯

11. Which device do you primarily use to access the chatbot? *

◯ Smartphone
◯ Laptop
◯ Tablet
◯ Desktop
◯ Other ________

12. Section 3: Impact

How much time do you save by using the chatbot compared to contacting staff? *

0

0 10

13. Has the chatbot reduced your need to contact academic staff? *

- ◯ Yes
- ◯ No

14. Rank the following benefits of the chatbot in order of importance. *

Fast responses	Select
24/7 availability	Select
Reduced staff contact	Select
Accurate answers	Select

15. Section 4: Limitations
Qualitative Questions

Describe any situation where the chatbot failed to answer your query (e.g., complex or non-FAQ questions). *

16. If the chatbot couldn't answer, what did you do next? *

- ◯ Contacted staff
- ◯ Searched online
- ◯ Gave up
- ◯ Other (specify)

17. Section 5: User Satisfaction
How likely are you to recommend the chatbot to other students? *

	0	1	2	3	4	5	6	7	8	9	10	
Not at all likely	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Extremely likely

16. If the chatbot couldn't answer, what did you do next? *

- ◯ Contacted staff
- ◯ Searched online
- ◯ Gave up
- ◯ Other (specify) ______

17. Section 5: User Satisfaction
How likely are you to recommend the chatbot to other students? *

	0	1	2	3	4	5	6	7	8	9	10	
Not at all likely	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Extremely likely

Previous Submit