



Ghana

# CoinSavvy

A WhatsApp quiz and educational chatbot using advanced language modelling for accessible financial tutoring.

[Darkye, Ahmed \(38557207\)](#)

LANCASTER UNIVERSITY GHANA  
COMPUTER SCIENCE DEPARTMENT

## **DECLARATION**

I, Ahmed Darkye, hereby attest that the content presented in this dissertation, entitled "CoinSavvy: A WhatsApp quiz and educational chatbot using advanced language modelling for accessible financial tutoring", is an authentic piece of work authored by myself, and was carried out under the supervision of Dr. Pius Kwao Gadosey from the Department of Computer Science at Lancaster University Ghana. With the exception of explicit disclosures, all content pertaining to the CoinSavvy initiative in this dissertation is the result of my individual effort and has not been previously presented for any academic qualification at this or any other academic institution. This investigation has been executed in compliance with the ethical standards of Lancaster University Ghana and all relevant professional bodies. I consent to the digital retention and duplication of this manuscript for assessment objectives, encompassing the verification of originality through the institution's anti-plagiarism mechanisms. Nevertheless, I retain the authority to manage the dissemination of the content articulated in this study along with any supplementary materials, disclosing information regarding the execution of the system. Approval in written form must be obtained prior to the public disclosure of this data.

### ABSTRACT

In the pursuit of revolutionizing financial literacy, the "CoinSavvy" initiative presents a groundbreaking approach through an AI-enhanced WhatsApp integrated application. The research question driving this endeavor is framed as, "How can cutting-edge AI language models within a WhatsApp-based finance quiz and educational app bridge the gap in financial literacy and empower users to make informed financial decisions?". A key component is the integration of an AI language model, enhancing the learning experience of the finance app. According to preliminary results, the project has the potential to greatly increase financial literacy by giving users a dynamic and easily accessible platform to interact with financial education materials. Notably, the WhatsApp AI-powered chatbot enhances user personalisation and engagement. "CoinSavvy" stands out as a trailblazing initiative in today's intricate financial environment, democratising financial literacy via cutting-edge technology. Through the utilisation of advanced artificial intelligence (AI) capabilities and the extensive use of WhatsApp, the initiative seeks to equip people all over the world with the necessary knowledge and abilities to make wise financial decisions, ultimately promoting economic prosperity. This abstract effectively summarises the main ideas and step-by-step development of the "CoinSavvy" project.

**KEYWORDS:** *Financial Literacy, NLP, Chatbots, Education Technology, WhatsApp Integration, Machine Learning*

# COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

## Table of Contents

<b>DECLARATION.....</b>	<b>1</b>
<b>ABSTRACT .....</b>	<b>2</b>
<b>CHAPTER 1 - INTRODUCTION.....</b>	<b>7</b>
<b>CHAPTER 2 - LITERATURE REVIEW .....</b>	<b>10</b>
<b>    2.1 Integration of Natural Language Processing in financial education .....</b>	<b>10</b>
2.1.1 Introduction:.....	10
2.1.2 Advancements in NLP:.....	10
2.1.3 Applications in Finance Tutoring:.....	10
2.1.4 Ethical and Regulatory Considerations:.....	10
2.1.5 Balanced Perspective:.....	11
<b>    2.2 Chatbots in Education.....</b>	<b>11</b>
2.2.1 Introduction:.....	11
2.2.2 Strengths for Education:.....	11
2.2.3 Risks and Downsides: .....	11
2.2.4 Mitigation Strategies:.....	11
2.2.5 Considerations for a Finance Tutoring Chatbot: .....	11
2.2.6 Conclusion: .....	11
<b>    2.3 Using Artificial Intelligence in Education.....</b>	<b>12</b>
2.3.1 Introduction:.....	12
2.3.2 Relevance to Finance Tutoring Chatbot: .....	12
2.3.3 Ethical Concerns and Challenges:.....	12
2.3.4 Implications for a Finance Tutoring Chatbot: .....	12
2.3.5 Conclusion: .....	12
<b>    2.4 Financial Literacy and Education .....</b>	<b>12</b>
2.4.1 Introduction:.....	12
2.4.2 Relevance to Proposed Project: .....	13
2.4.3 Empirical Evidence and Disparities:.....	13
2.4.4 Implications for the AI-based Chatbot: .....	13
2.4.5 Conclusion: .....	13
<b>CHAPTER 3 - SYSTEM ANALYSIS AND DESIGN.....</b>	<b>14</b>
<b>    3.1 System functional requirements .....</b>	<b>14</b>
<b>    3.2 System non-functional requirements .....</b>	<b>16</b>
<b>    3.3 System constraints .....</b>	<b>17</b>
<b>    3.4 Types of system users.....</b>	<b>18</b>
<b>    3.5 Architectural pattern .....</b>	<b>19</b>
3.5.1 Visual representation of CoinSavvy's architectural pattern .....	19
3.5.2 Introduction to CoinSavvy's MVT Architecture .....	19
3.5.3 Conclusion - WhatsApp User Interface: .....	20
<b>    3.6 Software development life cycle (SDLC) model: Agile development .....</b>	<b>20</b>
<b>    3.7 Class representation .....</b>	<b>22</b>
3.7.1 Class diagram for entire project .....	22
3.7.2 Class description .....	22
<b>    3.8 Use cases .....</b>	<b>24</b>
3.8.1 Use case diagram.....	24
3.8.2 Use case descriptions .....	25

# COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

<b>3.9 Data flow .....</b>	<b>31</b>
3.9.1 Context diagram .....	31
<b>3.10 Sequence diagram .....</b>	<b>32</b>
3.10.1 Quiz sequence .....	32
3.10.2 Language model sequence .....	33
<b>CHAPTER 4 – IMPLEMENTATION .....</b>	<b>34</b>
<b>4.1 PROGRAMMING LANGUAGE.....</b>	<b>34</b>
<b>4.2 FRAMEWORKS FOR API DEVELOPMENT .....</b>	<b>34</b>
4.2.1 Django Framework .....	34
4.2.1.1 Project Creation and App Setup .....	34
4.2.1.2 File Structure in Django .....	35
4.2.1.3 Integration with External APIs .....	36
4.2.1.4 Database Management .....	36
4.2.2 Quiz Functionality Description .....	36
4.2.2.1 Django App Structure .....	36
4.2.2.2 Quiz Logic Implementation.....	36
4.2.3 Flask framework .....	37
4.2.3.1 Why Flask? .....	37
4.2.3.2 Flask API Implementation.....	37
<b>4.3 DATABASE.....</b>	<b>39</b>
<b>4.4 TUNNELING .....</b>	<b>39</b>
4.4.2 Ngrok Setup for Django Project.....	40
<b>4.5 EXTERNAL APIs AND LIBRARIES.....</b>	<b>40</b>
4.5.1 Twilio WhatsApp API .....	40
4.5.1.1 Configuration and Setup for Twilio[20]: .....	41
4.5.1.2 Utilization in the Project .....	41
4.5.1.3 Ngrok Integration.....	42
4.5.2 Audio Processing and Transcription for Speech Recognition (Within Django Project) .....	42
4.5.2.1 Retrieval of Audio from WhatsApp .....	42
4.5.2.2 Audio Conversion from OGG to WAV.....	43
4.5.2.3 Transcribing Audio to Text[23] .....	43
4.5.2.4 Integration with Application Logic.....	45
4.5.3 LangChain – Finance AI Development.....	46
4.5.3.1 Tools and Library Integration: .....	46
4.5.3.2 Chains in LangChain: .....	48
4.5.3.4 OpenAI API and Embeddings: .....	48
4.5.3.5 Agents in LangChain.....	49
4.5.3.6 Function within the CoinSavvy AI Development: .....	49
<b>4.6 LLM DATASET .....</b>	<b>51</b>
4.6.1 Integration of GPT-3 and GPT- 4: .....	51
4.6.2 Incorporation of Custom Dataset:.....	51
4.6.3 Hybrid Approach for Enhanced Performance: .....	51
4.6.4 Data Extraction: .....	51
4.6.5 Data Processing: .....	52
<b>CHAPTER 5 – THE SYSTEM IN OPERATION.....</b>	<b>54</b>
<b>5.1 CoinSavvy QuizMaster .....</b>	<b>54</b>
5.1.2 New User Registration.....	54
5.1.3 Choosing a Topic.....	55
5.1.4 Quiz Participation .....	56
5.1.5 Question and Answering .....	57

# **COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP**

5.1.6 Display of Score .....	58
5.1.7 Topic Survey.....	59
<b>5.2 CoinSavvy AI .....</b>	<b>60</b>
5.2.1 Conversing with the AI .....	60
5.2.2 Asking about Financial Topics.....	61
5.2.3 Summarize Financial Topics.....	62
5.2.4 Obtaining Real-Time Finance Information .....	63
5.2.5 Audio Interaction and Response .....	64
<b>CHAPTER 6 – PROCESS DESCRIPTION .....</b>	<b>66</b>
<b>6.1 Quiz Process Flow.....</b>	<b>66</b>
<b>6.2 AI Process Flow .....</b>	<b>66</b>
<b>CHAPTER 7 – TESTING AND EVALUATION .....</b>	<b>68</b>
<b>7.1 TESTING .....</b>	<b>68</b>
7.1.1 Methodology .....	68
7.1.1.1 Testing Approach.....	68
7.1.1.2 Tools and Frameworks.....	68
7.1.2. Test Cases and Scenarios .....	68
7.1.2.1 Quiz Functionality .....	68
7.1.2.2 AI Chatbot .....	69
7.1.2.3 Audio Processing.....	69
7.1.2.4 System Testing .....	69
7.1.3 Results and Discussion.....	69
7.1.3.1 Key Insights .....	69
7.1.3.2 Metrics .....	69
7.1.4 Summary of Findings .....	70
<b>7.2 EVALUATION.....</b>	<b>70</b>
7.2.1 Evaluation Methodology .....	70
7.2.1.1 Approach.....	70
7.2.1.2 Tools and Methods .....	70
7.2.2 Evaluation Metrics.....	70
7.2.2.1 Usability Metrics .....	70
7.2.2.2 Content Evaluation Metrics .....	70
7.2.2.3 Performance Metrics .....	71
7.2.2.4 Effectiveness Metrics.....	71
7.2.3. Results and Analysis .....	71
7.2.3.1 Usability Evaluation Results.....	71
7.2.3.2 Content Evaluation Results.....	71
7.2.3.3 Performance Evaluation Results.....	71
7.2.4 Effectiveness Evaluation Results .....	71
7.2.5 Critical Review and Limitations .....	72
<b>CHAPTER 8 – CONCLUSION.....</b>	<b>73</b>
<b>8.1 Review of Aims.....</b>	<b>73</b>
<b>8.2 Suggested Revisions to Design/Implementation .....</b>	<b>73</b>
<b>8.3 Future Work.....</b>	<b>73</b>
<b>8.4 Lessons Learned .....</b>	<b>74</b>
<b>8.5 Concluding Remarks .....</b>	<b>75</b>
<b>REFERENCES .....</b>	<b>76</b>

## Table of Figures

Figure 1: CoinSavvy's Architecture .....	19
Figure 2: Illustration of the Agile Methodology phases[8].....	21
Figure 3: CoinSavvy's classes in diagram representation.....	22
Figure 4: CoinSavvy's use cases illustration .....	24
Figure 5: Data Flow illustrating how the system interacts with external entities .....	31
Figure 6: Sequence diagram illustrating the sequence of actions in quiz engagement.....	32
Figure 7: Sequence diagram illustrating interaction with language model.....	33
Figure 8: Django project creation .....	35
Figure 9: Django app creation.....	35
Figure 10: Making migrations in Django .....	35
Figure 11: Executing the Django server .....	36
Figure 12: JSON file containing questions.....	37
Figure 13: Flask End-Point in Language Model Code.....	38
Figure 14: Flask End-Point in Django Views.py.....	38
Figure 15: Snippet of sample data from quiz engagement.....	39
Figure 16: Starting Ngrok .....	40
Figure 17: Ngrok Interface .....	40
Figure 18: Twilio WhatsApp URL Webhook dashboard .....	41
Figure 19: Successfully fetching audio from WhatsApp.....	43
Figure 20: Conversion of audio from OGG to WAV.....	43
Figure 21: Transcription of WAV file .....	44
Figure 22: Simplified Illustration of how speech to text conversion works[24] .....	44
Figure 23: A detailed illustration of speech-to-text conversion using Google Cloud Platform API[25] .....	45
Figure 24: Flask End-Point in Django Views.py .....	45
Figure 25: Language Model Libraries.....	46
Figure 26: Illustration of Language Model using Web Search .....	50
Figure 27: Illustration of Language Model summarizing a topic .....	50
Figure 28: Vector representation of finance dataset .....	52
Figure 29: Simplified illustration of using LangChain for LLM dataset training[39] .....	53
Figure 30: Detailed Illustration of suing LangChain to handle PDFs for LLM training[40] .....	53
Figure 31: New User Registration Snippet .....	54
Figure 32: Using choosing a topic snippet .....	55
Figure 33: User participating in quiz snippet .....	56
Figure 34: Questioning and answering mechanism snippet .....	57
Figure 35: Display of score snippet .....	58
Figure 36: Topic survey snippet.....	59
Figure 37: Conversing with AI snippet.....	60
Figure 38: Finance topics interaction snippet .....	61
Figure 39: Topic summary snippet .....	62
Figure 40: Real-time information snippet .....	63
Figure 41: Audio interaction snippet.....	64

### CHAPTER 1 - INTRODUCTION

The "CoinSavvy" initiative is revolutionizing financial literacy and decision-making through the use of cutting-edge artificial intelligence (AI) within the modern financial landscape. By launching a finance app integrated into WhatsApp, it addresses a significant gap in financial education. The initiative employs AI language models to equip individuals with the necessary knowledge and skills for making informed financial decisions, and fostering a more secure financial future.

Financial literacy is crucial for safeguarding personal funds, planning for the future, and achieving financial goals in today's economic climate[1]. Despite its importance, many people lack the basic skills needed for effective financial management due to a pervasive knowledge gap. "CoinSavvy" aims to bridge this gap with its finance application that utilizes AI and quizzes to enhance financial decision-making abilities for a wider audience.

Understanding and applying financial concepts such as debt management, saving, investing, budgeting, and navigating financial products are fundamental for sound financial control[2]. However, the challenge of financial illiteracy leads to poor financial choices, increasing debt, and missed opportunities. "CoinSavvy" seeks to leverage WhatsApp's extensive user base to improve financial literacy with a personalized and engaging experience through its AI-powered chatbot.

The app's integration as a WhatsApp chatbot, combined with advanced AI, ensures accessibility for users worldwide, regardless of location or demographic differences[3]. Its primary objective is to provide users with the knowledge and skills for prudent financial management. The use of AI language models creates a dynamic and adaptable learning environment.

"CoinSavvy" aims to close the gap in financial education availability by positioning itself as a finance quiz app that leverages modern technology to make financial education engaging, empowering, and accessible. The initiative's AI-powered chatbot on WhatsApp offers a mix of interactive tests and individualized financial guidance through advanced AI language models, addressing the widespread issue of financial literacy and the consequences of poor financial decisions.

The ultimate goal is to enhance users' overall financial literacy by furnishing them with the knowledge and skills required for informed financial decisions, with the AI aspect ensuring a cutting-edge and adaptive learning experience. Through active learning, critical thinking, and personalized financial tutoring facilitated by AI, the chatbot promotes financial responsibility and confidence. The democratization of financial literacy, facilitated by the accessibility of financial education via WhatsApp and enriched with advanced AI capabilities, seeks to improve individual financial decision-making and overall economic well-being. The aim and objectives of this project are mentioned below:

**Research Question:** How can cutting-edge AI language models within a WhatsApp-based finance quiz and educational app bridge the gap in financial literacy and empower users to make informed financial decisions?

**Aim:** Develop an application embedded on WhatsApp Business that provides accessible financial tutoring and quizzes on thoroughly researched financial topics.

### Objectives:

- **Increasing Financial Literacy:** Enhance financial literacy by offering a variety of interactive quizzes and educational content on financial topics.
- **Reaching a Wide Audience:** Leverage the popularity of WhatsApp to reach a diverse audience globally, ensuring accessibility to financial education and quizzes for users from different regions, backgrounds, and age groups.
- **Continuous Improvement Based on User Feedback:** Actively seek and consider user feedback to identify areas for enhancement, optimize quizzes, and improve the overall user experience.
- **Accessible Learning:** Provide accessible learning by integrating financial quizzes and educational content directly within WhatsApp, ensuring a user-friendly learning experience.
- **Measuring Impact on Users:** Measure the impact on users by tracking engagement, quiz performance, and progress over time to evaluate the effectiveness of educational efforts in enhancing users' financial literacy and decision-making abilities.
- **Collaborating with Experts in Finance:** Collaborate with finance industry experts to ensure the accuracy and reliability of content, delivering high-quality quizzes and educational materials aligned with the latest financial trends and best practices.
- **Promoting Informed Financial Decision-Making:** Promote informed financial decision-making through thought-provoking quizzes and informative content, encouraging users to critically assess financial scenarios.
- **Innovation in Educational Technology:** Advance the field of educational technology by developing a novel learning tool, committing to improving financial education through technology as a catalyst for change.

As we embark on an in-depth journey through the "CoinSavvy" initiative, understanding the intricacies of each chapter is essential. This chapter-by-chapter overview offers a roadmap for readers, providing a glimpse into the key aspects and insights they can expect to discover in the subsequent sections.

### Chapter 2: Background (Literature Review)

As we delve into the "CoinSavvy" initiative, Chapter 2 will serve as the foundation, exploring the existing body of knowledge in financial literacy initiatives, AI applications in education, and the integration of technology in messaging platforms. This comprehensive literature review will set the stage for understanding the context, challenges, and opportunities that inform the rationale behind "CoinSavvy."

### Chapter 3: Design (System Design and Analysis)

Building upon the insights gained from the literature review, Chapter 3 will shift focus to the intricacies of the "CoinSavvy" system design. Here, we will discuss the architecture, functionalities, and underlying logic that govern the finance app. System analysis will provide

## **COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP**

readers with a clear understanding of how the application aligns with the project's objectives, ensuring a seamless transition from conceptualization to execution.

### **Chapter 4: Implementation**

With the design principles in place, Chapter 4 will take a deep dive into the practical implementation of the "CoinSavvy" initiative. From coding and software development to the integration of AI language models, this chapter will unveil the technical aspects that bring the finance quiz and artificial intelligence app to life, ensuring a robust and effective solution.

### **Chapter 5: The System in Operation**

Chapter 5 will provide a firsthand look at the day-to-day functioning of the "CoinSavvy" system. Readers will gain insights into user interactions, the seamless integration of the finance quiz and artificial intelligence app within WhatsApp, and the overall operational processes that make the initiative user-friendly and accessible.

### **Chapter 6: Process Description**

This chapter delves deeper into a high-level overview, offering a more detailed understanding of key functionalities and processes.

### **Chapter 7: Testing and Evaluation**

In Chapter 7, we will shift our focus to the rigorous testing processes employed to ensure the reliability, security, and effectiveness of the "CoinSavvy" application. Evaluation metrics and user feedback will be analyzed, providing valuable insights into the success of the initiative in achieving its goals and paving the way for potential improvements.

### **Chapter 8: Conclusion**

The final chapter will synthesize the findings from the preceding chapters, offering a comprehensive overview of the "CoinSavvy" initiative. This concluding section will encapsulate the achievements and impact of the project, highlighting its contributions to financial literacy and decision-making. Moreover, it will explore potential avenues for future research and development, cementing "CoinSavvy" as a pioneering effort in the realm of AI-driven financial education.

## CHAPTER 2 - LITERATURE REVIEW

The four selected papers contribute to setting the stage for the development of a finance tutoring chatbot. These papers collectively explore topics ranging from the future of natural language processing (NLP) in finance, the impact of AI-based chatbots in education, the role of artificial intelligence in education, to the importance of financial literacy and the need for financial education. These papers stress responsible implementation in education, explore the potential applications and ethical considerations of large language models, and underline the critical need to address global financial illiteracy. Collectively, they paved the way for the Finance Tutoring Chatbot, imagining a game-changing instrument that combines smart learning, sophisticated language processing, and creative approaches to close the global financial literacy gap.

### 2.1 Integration of Natural Language Processing in financial education

#### 2.1.1 Introduction:

This section critically examines the utilization of large language models, honing in on a specific paper that discusses the implications of employing such models in the finance sector, showcasing their potential to revolutionize natural language processing (NLP) within this domain[4]. Delving into the potential applications and ethical considerations within natural language processing (NLP) tasks in finance, the paper emphasizes the increasing importance of NLP in handling unstructured textual data[4]. It underscores the role of ChatGPT in tasks like sentiment analysis, text summarization, and language generation, harnessing its advanced transformer architecture and diverse training data. The insights from this paper serve as a cornerstone, guiding the examination of ethical implications, potential applications, and the transformative impact of large language models like ChatGPT in the realm of finance education and tutoring[4].

#### 2.1.2 Advancements in NLP:

ChatGPT's transformative impact on NLP is emphasized, attributed to its advanced transformer architecture and comprehensive training on diverse textual datasets[4]. The potential improvements it brings to existing financial applications and the possibility of enabling novel applications are discussed, with a specific focus on natural language understanding, generation, and explanation.

#### 2.1.3 Applications in Finance Tutoring:

Highlighting the relevance to finance education, the paper envisions ChatGPT's role in enhancing finance tutoring chatbots. Its capabilities could significantly improve the comprehension of student queries, provide clearer explanations of financial concepts, and generate tailored study materials such as practice questions and study guides[4].

#### 2.1.4 Ethical and Regulatory Considerations:

The literature underscores the ethical challenges and regulatory considerations associated with deploying ChatGPT in sensitive domains like finance. Risks related to bias, privacy, transparency, and compliance are articulated. For developers creating a finance tutoring chatbot, rigorous testing and safeguards are deemed essential to ensure fair and unbiased guidance, transparent explanations, and appropriate handling of personal data, aligning with ongoing research on interpretability and algorithm auditing.

## **COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP**

### **2.1.5 Balanced Perspective:**

In conclusion, the paper offers a nuanced perspective, weighing ChatGPT's remarkable capabilities against the open challenges when applied in the finance sector. While recognizing its exciting potential for enhancing explanations and content generation in a finance tutoring chatbot, the importance of responsible and ethical implementation is emphasized, acknowledging the need for ongoing research in interpretability and bias mitigation.

## **2.2 Chatbots in Education**

### **2.2.1 Introduction:**

This review critically evaluates the insights gained from a study examining the effects of employing Artificial Intelligence-based ChatGPT chatbots in education, particularly focusing on their impact on both teachers and students[5].The paper explores the influence of chatbots, especially ChatGPT, on education, emphasizing their quick responses, interactive learning, and content generation. Its relevance to a finance tutoring chatbot underscores the paper's significance in guiding the intersection of AI, education, and financial literacy[5].

### **2.2.2 Strengths for Education:**

The paper underscores the strengths of chatbots like ChatGPT, emphasizing their relevance to education, particularly in fields like finance tutoring[5]. The ability to comprehend student queries, deliver clear explanations, and generate tailored practice problems aligns with the requirements of an effective learning tool.

### **2.2.3 Risks and Downsides:**

Acknowledging the benefits, the authors caution against risks associated with inadequate implementation of chatbots. Concerns include reliance on imperfect information, potential erosion of critical thinking skills through excessive automation, diminished human connection, and data privacy issues[5]. The possibility of students assuming responses are entirely accurate is also highlighted.

### **2.2.4 Mitigation Strategies:**

The paper suggests mitigation approaches to address the identified risks. Recommendations include teacher involvement in monitoring content quality, guiding appropriate chatbot use, and maintaining a transparent knowledge base with clearly defined limitations[5]. Feedback loops, oversight mechanisms, and safeguards for student data are proposed, along with the integration of interactive visuals and real-world examples to supplement automated content.

### **2.2.5 Considerations for a Finance Tutoring Chatbot:**

Applying these considerations to a finance tutoring chatbot, the need for a broad and accurate knowledge base is emphasized. Transparent limitations, feedback loops, and safeguards for student data become paramount[5]. The role of teachers in providing guidance, overseeing content quality, and preventing overreliance on the bot is underscored.

### **2.2.6 Conclusion:**

In summary, the paper recognizes ChatGPT's potential in education, presenting relevant insights for a finance tutoring chatbot. While highlighting the benefits of personalized explanations and practice problem generation, the paper underscores the importance of thoughtful instructional design and human oversight to mitigate risks and enhance the learning

## **COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP**

experience. The call for further research on balancing automation with human judgement in AI remains a pertinent consideration.

### **2.3 Using Artificial Intelligence in Education**

#### **2.3.1 Introduction:**

This review draws insights primarily from a comprehensive exploration of Artificial Intelligence in the field of education[6]. The paper provides an extensive overview of AI and its applications in education, showcasing technologies like machine learning, neural networks, and natural language processing. The relevance of AI's capabilities, as discussed in the paper, holds substantial implications for the design and implementation of a finance tutoring chatbot, aligning with the broader exploration of AI in education[6].

#### **2.3.2 Relevance to Finance Tutoring Chatbot:**

The capabilities of AI outlined in the paper align seamlessly with the requirements of a finance tutoring chatbot. Notable features include natural language processing for comprehending finance-related queries, machine learning for continuous knowledge enhancement, and expert systems for delivering detailed, personalized explanations of finance concepts[6]. Additionally, automated grading and the generation of practice questions or study guides are identified as tangible possibilities.

#### **2.3.3 Ethical Concerns and Challenges:**

While highlighting the transformative potential of AI in education, the authors judiciously draw attention to ethical concerns and challenges[6]. These concerns, pertinent to an AI-powered finance tutor, encompass student privacy, data security, transparency in response generation, and the need for oversight to ensure accuracy. Addressing potential biases in algorithms and finding the delicate balance between automation and the development of critical thinking skills are also emphasized.

#### **2.3.4 Implications for a Finance Tutoring Chatbot:**

In summarizing the insights, the paper underscores the need for responsible design and application of AI in education. For a finance tutoring chatbot, the opportunities to enhance finance learning through conversational interaction, automated content creation, and adaptive explanations are highlighted[6]. Simultaneously, the paper alerts to crucial pitfalls, emphasizing the necessity for proactive mitigation of issues related to data privacy, transparency, and the risk of student overreliance.

#### **2.3.5 Conclusion:**

In conclusion, the paper serves as a valuable resource for understanding the potential of AI in education and its specific implications for a finance tutoring chatbot. Balancing the transformative benefits with ethical considerations, the paper accentuates the importance of responsible implementation to harness the full potential of AI in enhancing the learning experience.

## **2.4 Financial Literacy and Education**

#### **2.4.1 Introduction:**

This review of the literature is informed by a study focusing on financial literacy and the necessity for financial education[7]. The "Big Three" standardised questions are introduced as

## **COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP**

essential instruments for assessing financial knowledge gaps, and the paper sheds light on the global problem of low financial literacy levels. An empirical base on the effects of financial illiteracy and a trend analysis of the factors driving the increased demand for financial literacy provide a solid basis. In addition to highlighting the necessity of creative and scalable solutions to address the widespread problem of financial illiteracy worldwide, this paper serves as the foundation for the proposed AI-based financial literacy chatbot.

### **2.4.2 Relevance to Proposed Project:**

The paper's comprehensive review of financial literacy aligns seamlessly with the objectives of the proposed AI-based financial literacy chatbot. Highlighting key trends contributing to the heightened need for financial literacy, such as shifts in retirement accounts, increased life expectancies, wage inequality, and technological innovations, the paper provides a compelling backdrop for the urgency of the project. The empirical evidence presented on the impacts of financial illiteracy on wealth accumulation, retirement planning, investment behaviors, and debt management reinforces the critical role of tailored financial education.

### **2.4.3 Empirical Evidence and Disparities:**

Empirical evidence from multiple developed countries forms a significant part of the paper, emphasizing the tangible consequences of financial illiteracy. The studies reveal that lower financial literacy is linked to a decreased likelihood of engaging in retirement planning and strategic asset allocation, along with an increased reliance on high-cost debt products[7]. Moreover, age, gender, and socioeconomic disparities in financial literacy underscore the importance of targeted interventions.

### **2.4.4 Implications for the AI-based Chatbot:**

The paper's findings directly inform the rationale behind developing an AI-based financial literacy chatbot. The need for innovative and scalable solutions, especially for vulnerable demographic groups, is emphasized. The proposed chatbot, with its ability to engage in personalized dialogues assessing and addressing users' gaps in financial knowledge, is positioned as a major potential solution. The suggestion to incorporate the standard Big Three questions aligns with the chatbot's potential to evaluate user literacy comprehensively.

### **2.4.5 Conclusion:**

In conclusion, Lusardi's paper serves as a foundational piece, revealing the gravity of global financial illiteracy and its repercussions. The proposed AI-based financial literacy chatbot emerges as a timely and essential solution, leveraging innovative technology to address the identified gaps and disparities. The "Big Three" questions, as highlighted in the paper, offer a structured approach for evaluating and enhancing users' financial literacy on a global scale.

### CHAPTER 3 - SYSTEM ANALYSIS AND DESIGN

The system analysis and design for CoinSavvy, a web app that offers financial education through an integrated WhatsApp experience, is presented in this chapter. It describes the requirements, both functional and non-functional, that determine the characteristics and functionality of the system. The architectural patterns are explained, together with the Model-View-Template framework and the Agile development process that were selected. The chapter also discusses the entire software development life cycle, including use cases, class diagrams, sequence diagrams, and data flow diagrams that show how interactive quizzes and the conversational AI tutor work. An ideal, user-centric design is ensured by analysing the system's limitations, user roles, and component interactions. This improves financial literacy.

#### 3.1 System functional requirements

- **User Registration and Authentication**
  - The system should allow users to register for the CoinSavvy app using their WhatsApp accounts.
  - Two-factor authentication should be implemented to enhance security.
- **Quiz Interaction**
  - The system should enable users to access interactive quizzes covering various financial topics.
  - The system should provide real-time feedback on quiz performance.
- **Educational Content Delivery**
  - The system should provide educational content on financial concepts and best practices.
  - The system should present content in an engaging and easily understandable manner.
- **WhatsApp Chatbot Functionality**
  - The system should be integrated into WhatsApp as a chatbot with advanced language modeling for personalized financial tutoring.
- **User Progress Tracking**
  - The system should track user engagement and progress over time.
  - The system should provide analytics used to measure educational impact and outcomes.
- **Administrative Functions**
  - The system should provide administrators with the ability to manage user accounts and access.
  - The system should allow administrators to manipulate educational content.
- **Personalized Financial Tutoring**
  - The system should offer personalized financial tutoring based on user interactions.
  - The system should provide recommendations and insights tailored to individual user needs should be provided.
- **Data management and Encryption**

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

- The system should ensure secure storage and management of user data.
- **Interactive Artificial Intelligence Language Model**
  - The system should integrate an interactive AI language model for dynamic and context-aware user interactions.

### 3.2 System non-functional requirements

- **Performance**
  - The system should respond promptly to user interactions.
  - The system should load up quizzes and educational quickly.
- **Scalability**
  - The system should be able to handle a growing user base.
  - The system should be capable of accommodating an increasing number of quizzes and educational materials.
- **Reliability**
  - The system should be available for users at all times.
  - Backup and recovery mechanisms should be implemented to prevent data loss.
- **Security**
  - Compliance with data security laws and regulations should be ensured.
  - The system should implement two-factor authentication for user accounts.
  - Regular security audits should be conducted to identify and address vulnerabilities.
- **Data Privacy**
  - The system should strictly adhere to user privacy standards.
  - Transparent communication about data usage and privacy policies should be maintained.
- **Expert Analysis Reliability**
  - The analysis provided by financial experts should be accurate and reliable.
  - A robust vetting process for expert contributions should be established.

### 3.3 System constraints

- **WhatsApp Integration Dependency:**
  - System performance is reliant on WhatsApp functionality; updates or disruptions may impact service.
- **Language Model Compatibility:**
  - Continuous adaptation is needed for compatibility with evolving NLP models.
- **Two-Factor Authentication:**
  - Security effectiveness is tied to successful implementation of two-factor authentication.
- **Scalability Challenges:**
  - Potential challenges with system scalability as user base grows.
- **Reliability on External Services:**
  - Overall system reliability is linked to the availability and performance of external services.
- **Compliance with Security Standards:**
  - Adherence to data security laws and regulations is essential for compliance.
- **Transparent Data Privacy Communication:**
  - Ongoing clear communication with users about data usage and privacy policies.
- **Vetting Process for Expert Contributions:**
  - The reliability of expert analysis is contingent on the effectiveness of the vetting process.
- **Performance Optimization:**
  - Continuous optimization is required to ensure prompt responses to user interactions.
- **Backup and Recovery Mechanisms:**
  - Essential mechanisms for backup and recovery to prevent data loss which involves regular testing necessary.
- **Ongoing Security Audits:**
  - Regular security audits are crucial for identifying and addressing potential vulnerabilities.
- **Adherence to User Privacy Standards:**
  - Strict adherence to evolving standards and expectations regarding user privacy.
- **Educational Content Controls:**
  - Administrative controls are needed to prevent misuse or unauthorized changes to educational content.
- **Data Encryption Standards:**
  - Secure storage is dependent on consistent implementation of data encryption standards.
- **Monitoring Educational Impact:**
  - Continuous monitoring is required to assess and enhance the educational impact of the system.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 3.4 Types of system users

#### Users:

##### 1. General Users:

- Individuals seeking to enhance their financial literacy.
- Engage with quizzes, educational content, and the finance-oriented language model.
- Includes students, professionals, and anyone interested in improving financial knowledge.

##### 2. Educators and Trainers:

- Use CoinSavvy as a supplementary tool for teaching financial literacy.
- Leverage quizzes, educational content, and the language model to support teaching objectives.

##### 3. Financial Experts:

- Contribute expertise to the platform by developing quiz questions, creating content, or offering insights.
- Enhance the quality and accuracy of financial information on the platform.

##### 4. Financial Institutions:

- Utilize CoinSavvy to promote financial literacy among their clients.
- Collaborate with the platform to customize content based on their services and target audience.

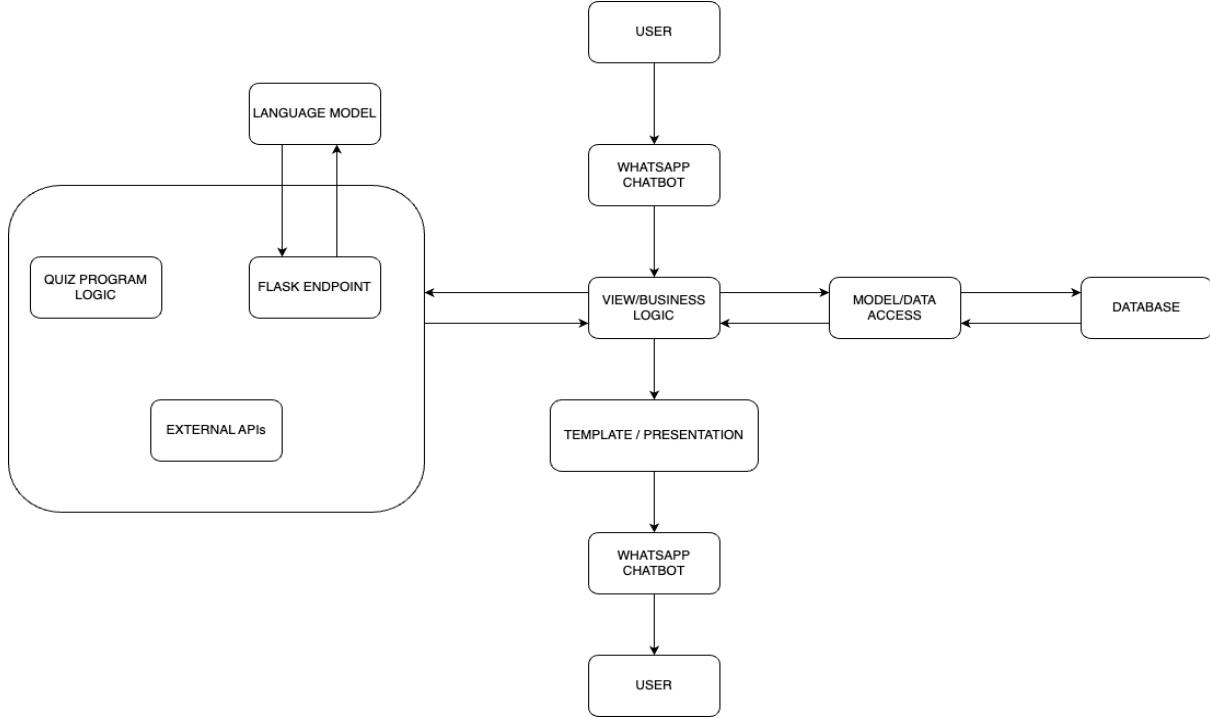
#### Administrators:

- Responsible for managing and overseeing the CoinSavvy platform.
- Have access to administrative functions for manipulating quiz content, managing user accounts, and assessing the impact of quizzes on financial literacy.
- Ensure the smooth operation and effectiveness of the platform.

By categorizing users into "Users" and "Administrators," CoinSavvy can tailor the platform's features to meet the specific needs and responsibilities of each group. Users benefit from interactive and educational content, while administrators have the tools to manage and enhance the overall platform experience.

### 3.5 Architectural pattern

### 3.5.1 Visual representation of CoinSavvy's architectural pattern



*Figure 1: CoinSavvy's Architecture*

### 3.5.2 Introduction to CoinSavvy's MVT Architecture

CoinSavvy is built on a solid framework that is based on the Model-View-Template (MVT) pattern. This is a customised version of the popular Model-View-Controller (MVC) architecture that is built with Django. This pattern is essential for offering a methodical approach to software development, guaranteeing a clear division of labour, and encouraging modularity. MVT directs component organisation in the CoinSavvy context, improving maintainability, scalability, and overall development efficiency. The context for examining the special qualities and advantages that CoinSavvy's MVT architecture brings to the forefront of contemporary software development is established in this section.

## Model Component:

- **Business Logic and WhatsApp Data Management:** The Model, within the MVT pattern, handles the business logic for CoinSavvy. For WhatsApp integration, this involves managing data related to user interactions, quiz responses, and personalized financial tutoring. It ensures the seamless flow and organization of information within the WhatsApp environment.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

- **SQL Database Integration:** It orchestrates user data stored in the SQL database, overseeing critical functionalities like user registration, progress tracking, and personalized financial tutoring. The Model plays a pivotal role in maintaining the integrity and consistency of data, serving as the foundation for the backend operations.

### View Component:

- **WhatsApp Presentation Layer:** The View, specifically tailored for WhatsApp, is responsible for rendering educational content within the chatbot. It dynamically adapts to user interactions, presenting financial quizzes and content in a manner compatible with WhatsApp's chat interface.
- **User Interface Consistency on WhatsApp:** Through the View, CoinSavvy maintains a consistent user interface within WhatsApp. Any modifications made to enhance the presentation layer on WhatsApp remain aligned with the overall user experience.

### Template Component:

- **WhatsApp Presentation Logic:** The Template, a key component in Django's MVT, contributes to the presentation logic, ensuring that data from the Model is appropriately presented within the WhatsApp chatbot. It dictates how information is displayed to users, optimizing the experience within the WhatsApp interface.

### Advantages in WhatsApp Interface:

- **Modularity and Maintainability for WhatsApp:** The MVT pattern's modularity enhances maintainability, enabling developers to make focused updates to components relevant to the WhatsApp interface without impacting other parts of the application. This is crucial for keeping the chatbot on WhatsApp up-to-date and responsive to user needs.
- **Scalability within WhatsApp:** The clear separation of concerns in MVT allows for independent scaling of components, ensuring that CoinSavvy can grow its user base on WhatsApp without sacrificing performance. The chatbot remains responsive and reliable as user engagement increases.
- **User Interface Consistency on WhatsApp:** The separation of components ensures that any changes made to the presentation layer, influenced by the View and Template, maintain a cohesive user interface. Users interacting with CoinSavvy on WhatsApp experience a consistent and familiar flow.

#### 3.5.3 Conclusion - WhatsApp User Interface:

The MVT architectural pattern, applied through Django, harmonizes seamlessly with the WhatsApp user interface. It provides a structured framework for CoinSavvy to deliver engaging financial education within the unique confines of WhatsApp's chat environment. This alignment ensures that users have an interactive, consistent, and user-friendly experience as they navigate financial quizzes and educational content on the WhatsApp platform.

## 3.6 Software development life cycle (SDLC) model: Agile development

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

For the CoinSavvy project, the Agile development methodology was selected because it promotes a collaborative and flexible atmosphere, which fits the project's nature and style. This is why Agile was thought to be a good fit:

**Flexibility:** As the project develops, agile enables flexibility in reacting to modifications and fresh perspectives. This adaptability is crucial in a project like CoinSavvy where learning and development are ongoing processes.

**Incremental Progress:** Agile allows for the delivery of work in brief cycles by dividing it into smaller, more manageable parts. As a result, progress can be demonstrated more quickly, and early feedback can assist keep things on track and achieve what's required.

**Prioritizing Users:** Agile emphasizes user involvement at every stage of the development process. In the case of CoinSavvy, where the objective is to provide users with financial literacy, this method guarantees that the product really fulfills the expectations and demands of the consumers.

**Continuous Learning:** Agile promotes frequent evaluation of achievements and necessary corrections. This promotes a culture of ongoing development, which is essential for a project like CoinSavvy where creativity is essential.

**Open Communication:** Agile encourages self-accountability and open communication. This guarantees objective congruence and permits individual decision-making so that every effort advances the project.

All things considered, Agile offers the adaptability, openness, and attention to user requirements required for CoinSavvy success. It's a manner of working that demonstrates a commitment to having a good impact on people's lives rather than merely a development process.

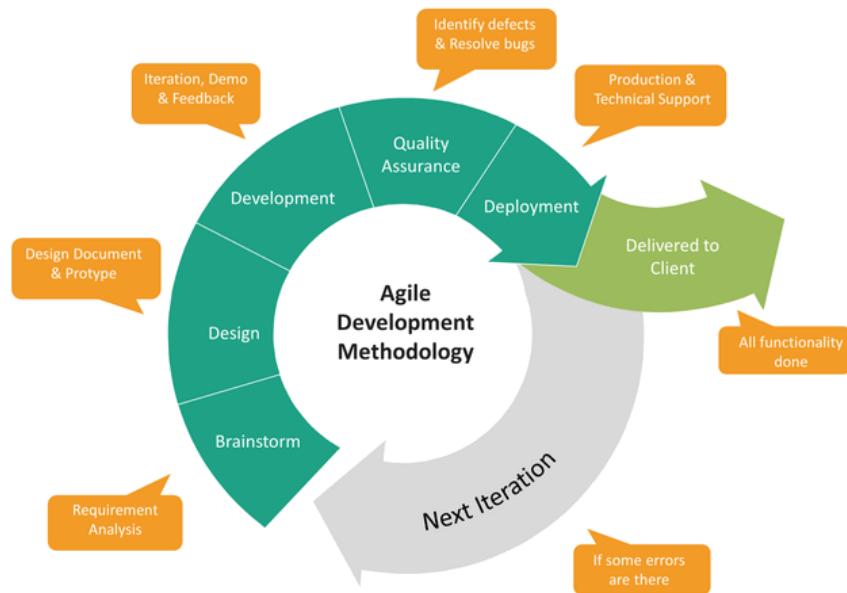


Figure 2: Illustration of the Agile Methodology phases[8]

## 3.7 Class representation

### 3.7.1 Class diagram for entire project

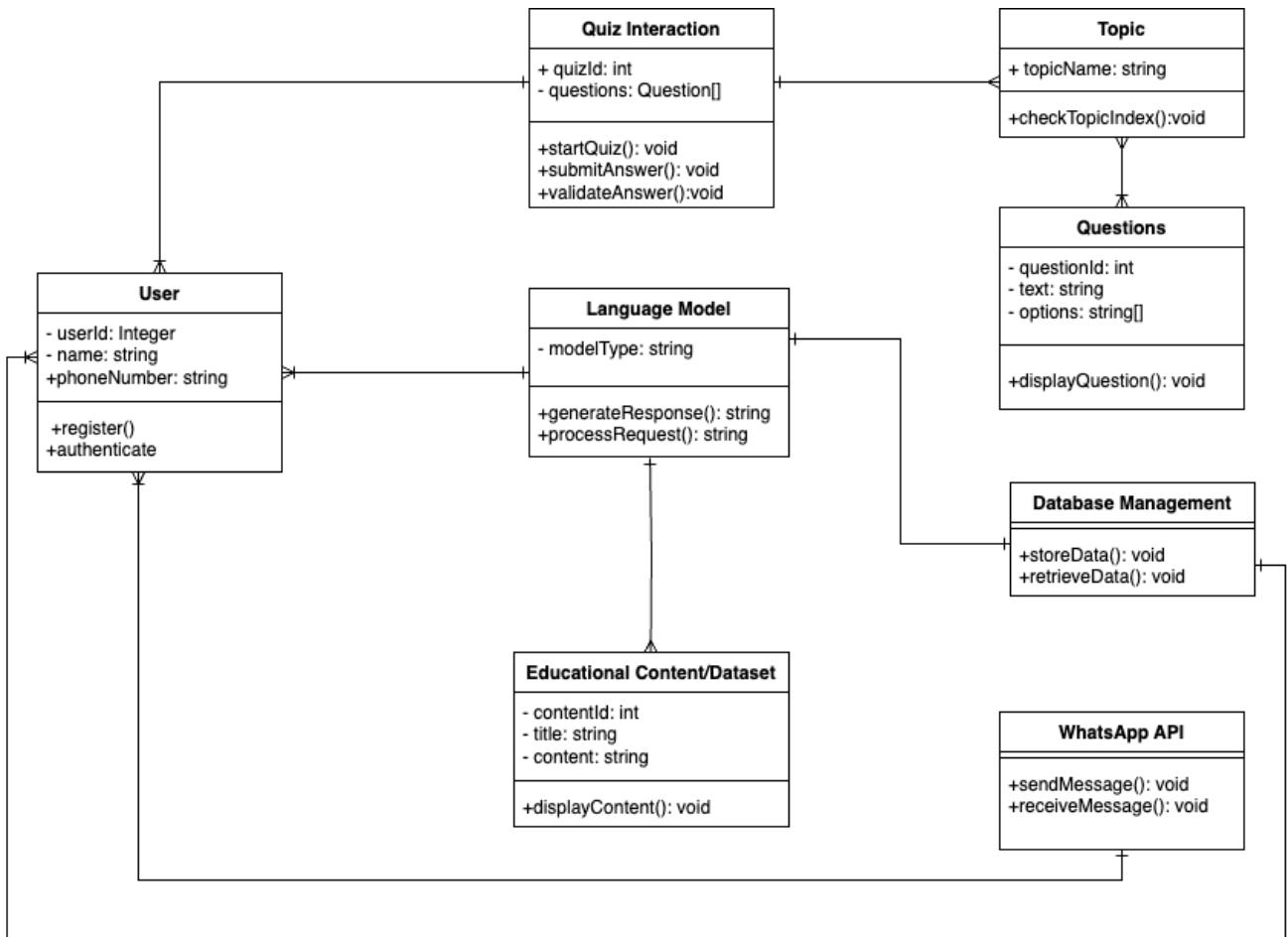


Figure 3: CoinSavvy's classes in diagram representation

### 3.7.2 Class description

#### User:

- **userId** (Integer): Unique identifier for each user.
- **name** (String): User's name.
- **phoneNumber** (String): User's phone number.
  - **register()**: Registers the user within CoinSavvy.
  - **authenticate()**: Authenticates the user, enhancing security with two-factor authentication.

#### Quiz Interaction:

- **quizId** (Integer): Identifier for the quiz.
- **questions** (Array of Question): Array containing quiz questions.
  - **startQuiz()**: Initiates the quiz.
  - **submitAnswer()**: Submits user's answer to a question.
  - **validateAnswer()**: Validates the submitted answer.

#### Topics:

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

- topicName (String): Name of the educational topic.
  - checkTopicIndex(): Checks the index of the educational topic.

### Questions:

- questionId (Integer): Identifier for the question.
- text (String): Text of the question.
- options (Array of String): Array containing answer options.
  - displayQuestion(): Displays the quiz question.

### Educational Content:

- contentId (Integer): Identifier for the educational content.
- title (String): Title of the content.
- content (String): Actual content.
  - displayContent(): Displays the educational content.

### WhatsApp API:

- sendMessage(): Sends a message via WhatsApp.
- receiveMessage(): Receives a message from WhatsApp.

### Database Management:

- storeData(): Stores data in the database.
- retrieveData(): Retrieves data from the database.

### Language Model:

- modelType (String): Type of language model.
  - generateResponse(): Generates a response based on input.
  - processRequest(): Processes a user's request.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 3.8 Use cases

#### 3.8.1 Use case diagram

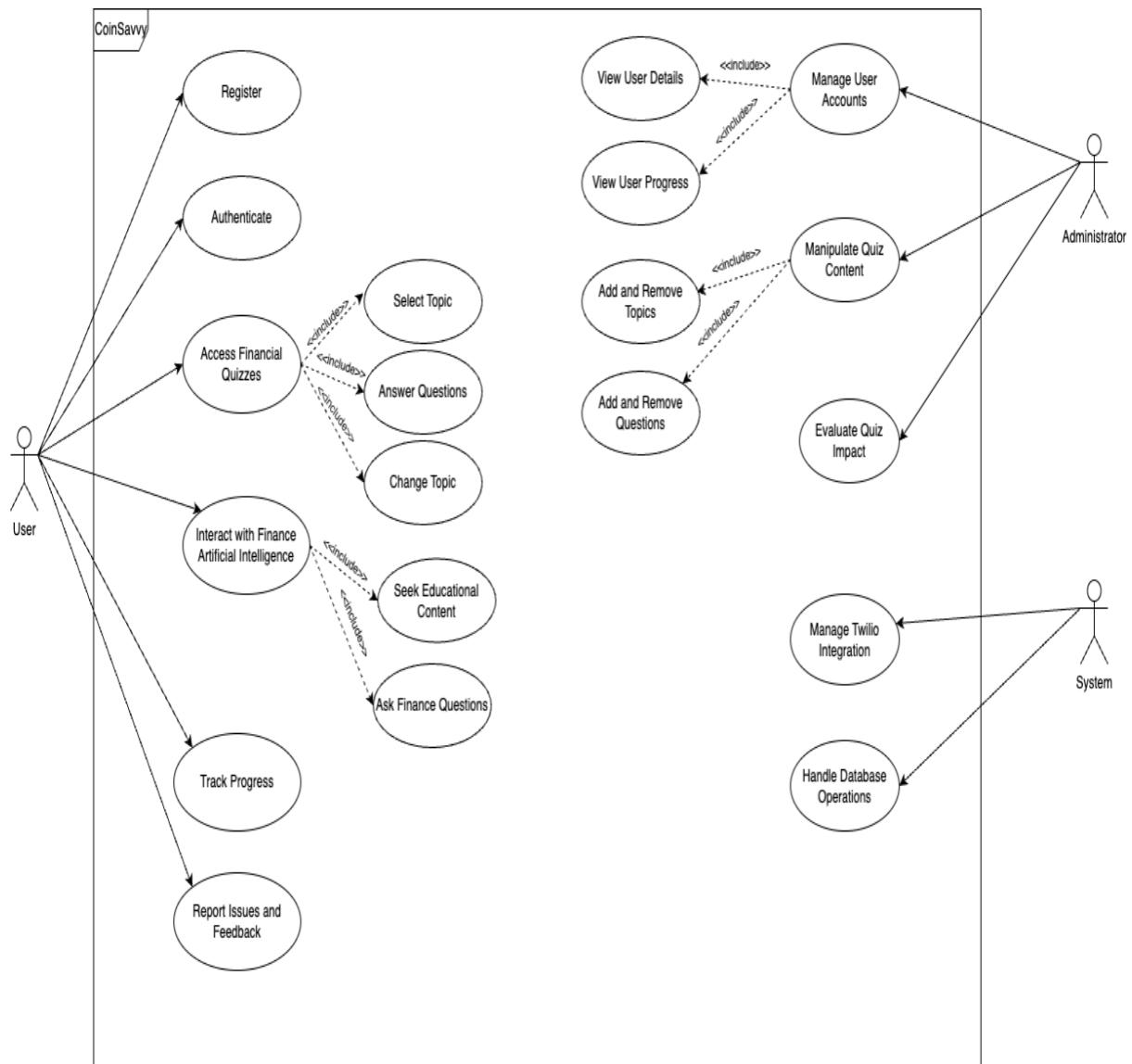


Figure 4: CoinSavvy's use cases illustration

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 3.8.2 Use case descriptions

<b>SYSTEM: USE CASE</b>	CoinSavvy: Register
<b>ACTORS</b>	User
<b>DESCRIPTION</b>	New users are automatically enrolled after engaging with CoinSavvy for the first time.
<b>DATA</b>	Phone number and Name
<b>STIMULI</b>	Send “Start” to CoinSavvy
<b>RESPONSE</b>	User is registered

<b>SYSTEM: USE CASE</b>	CoinSavvy: Authenticate
<b>ACTORS</b>	User
<b>DESCRIPTION</b>	The system authenticates the user whenever they use CoinSavvy
<b>DATA</b>	Phone Number and ID
<b>STIMULI</b>	Engage CoinSavvy
<b>RESPONSE</b>	Last session on CoinSavvy is resumed.

<b>SYSTEM: USE CASE</b>	CoinSavvy: Access Financial Quizzes
<b>ACTORS</b>	User
<b>DESCRIPTION</b>	Users can access the quizzes CoinSavvy has to offer.
<b>DATA</b>	NULL
<b>STIMULI</b>	Select “Quiz” from Options
<b>RESPONSE</b>	Various quiz topics will be displayed

<b>SYSTEM: USE CASE</b>	CoinSavvy: Select Topic
<b>ACTORS</b>	User
<b>DESCRIPTION</b>	Users will select a topic to be quizzed on

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

DATA	Topic Index
STIMULI	Selecting a quiz number
RESPONSE	Quiz on selected topic begins

SYSTEM: USE CASE	CoinSavvy: Answer Questions
ACTORS	User
DESCRIPTION	Users select an option from possible answers to questions
DATA	Option Index
STIMULI	Select answer option letter
RESPONSE	System checks answer and displays verdict

SYSTEM: USE CASE	CoinSavvy: Change Topic
ACTORS	User
DESCRIPTION	Users can change quiz topic anytime they desire
DATA	NULL
STIMULI	Type “Change” and send
RESPONSE	Current quiz ends and topics are displayed

SYSTEM: USE CASE	CoinSavvy: Interact with Finance Artificial Intelligence
ACTORS	User
DESCRIPTION	User can interact with language model designed for finance in certain domains
DATA	NULL
STIMULI	Select “CoinSavvy AI” in the beginning of program
RESPONSE	The system launches the language model

SYSTEM: USE CASE	CoinSavvy: Seek Educational Content
------------------	-------------------------------------

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

ACTORS	User
DESCRIPTION	User can ask language model for educational content within the defined domain
DATA	NULL
STIMULI	Type topic name
RESPONSE	Information on topic will be displayed

SYSTEM: USE CASE	CoinSavvy: Ask Finance Questions
ACTORS	User
DESCRIPTION	Users can ask specific questions within the defined domain
DATA	NULL
STIMULI	Type request
RESPONSE	Language model displays results

SYSTEM: USE CASE	CoinSavvy: Track Progress
ACTORS	User
DESCRIPTION	Users can check performance after each quiz
DATA	NULL
STIMULI	Finish a quiz or request performance for a quiz
RESPONSE	Details of performance is displayed

SYSTEM: USE CASE	CoinSavvy: Report Issues and Feedback
ACTORS	User
DESCRIPTION	Users are presented with a feedback form after every quiz completed
DATA	Feedback
STIMULI	Select quiz topic rating and issues

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

RESPONSE	Delivery confirmation
----------	-----------------------

SYSTEM: USE CASE	CoinSavvy: Manage User Accounts
ACTORS	Administrator
DESCRIPTION	Administrators can manage user information
DATA	NULL
STIMULI	Visit user database
RESPONSE	Display user information

SYSTEM: USE CASE	CoinSavvy: View User Details
ACTORS	Administrator
DESCRIPTION	Administrator can view all user details such as name, and phone number.
DATA	NULL
STIMULI	Visit user database
RESPONSE	Display user details

SYSTEM: USE CASE	CoinSavvy: View User Progress
ACTORS	Administrator
DESCRIPTION	Administrator can view all users and their respective activities and progress
DATA	NULL
STIMULI	Visit user database
RESPONSE	Display user progress

SYSTEM: USE CASE	CoinSavvy: Manipulate Quiz Content
ACTORS	Administrator
DESCRIPTION	Administrators can change contents of the quiz

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

DATA	Quiz content
STIMULI	Make changes
RESPONSE	Changes will reflect on the next usage of CoinSavvy

SYSTEM: USE CASE	CoinSavvy: Add and Remove Topics
ACTORS	Administrator
DESCRIPTION	Administrators can add or remove topics depending on importance in finance literacy
DATA	Quiz topics
STIMULI	Add, remove, or edit topics
RESPONSE	Changes will reflect on the next usage of CoinSavvy

SYSTEM: USE CASE	CoinSavvy: Add and Remove Questions
ACTORS	Administrator
DESCRIPTION	Administrators can add or remove questions depending on importance in finance literacy
DATA	Quiz questions
STIMULI	Add, remove, or edit questions
RESPONSE	Changes will reflect on the next usage of CoinSavvy

SYSTEM: USE CASE	CoinSavvy: Evaluate Quiz Impact
ACTORS	Administrator
DESCRIPTION	Administrators can evaluate the impact of the quizzes in contributing to financial literacy
DATA	NULL
STIMULI	Visit user feedback database
RESPONSE	Display feedback data

SYSTEM: USE CASE	CoinSavvy: Manage Twilio Integration
------------------	--------------------------------------

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

ACTORS	System
DESCRIPTION	The system configures and controls Twilio integration, managing API credentials and integration settings for continuous functionality.
DATA	Twilio API credentials, Integration settings
STIMULI	NULL
RESPONSE	System processes and updates Twilio integration settings, ensuring operational status.

SYSTEM: USE CASE	CoinSavvy: Handle Database Operations
ACTORS	System
DESCRIPTION	The system manages database operations, including user data storage, retrieval, and updates.
DATA	User registration, Quiz progress
STIMULI	User registration, Quiz progress
RESPONSE	System performs database operations, ensuring data consistency and integrity.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 3.9 Data flow

#### 3.9.1 Context diagram

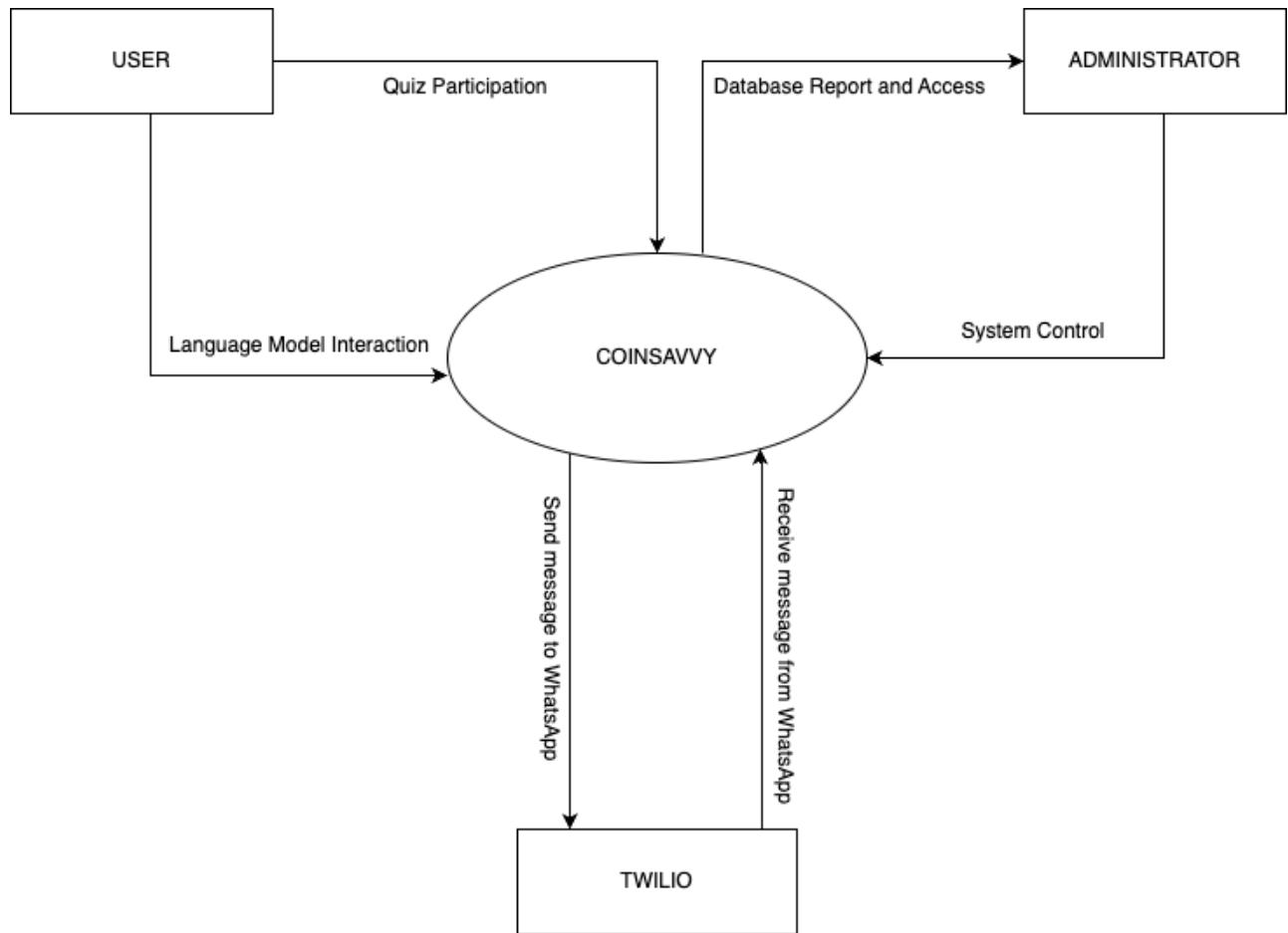


Figure 5: Data Flow illustrating how the system interacts with external entities

In the context diagram, "CoinSavvy" is depicted as the central process, and it interacts with various external entities. Below is a concise breakdown:

#### CoinSavvy:

- Serves as the central process or system under consideration.
- Manages interactions with different entities and performs essential functions.
- Acts as the core platform for providing financial education through quizzes, language model interactions, and user engagement.

#### External Entities:

1. **User:**
  - Participates in quizzes and interacts with the language model.
  - Receives educational content and feedback from CoinSavvy.
2. **Administrator:**
  - Manages and accesses reports from the database.
  - Controls and oversees the overall system functionality.
3. **Twilio:**

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

- Facilitates communication by sending and receiving messages to and from WhatsApp.
- Integrates with CoinSavvy to enable interaction with users on the WhatsApp platform.

The context diagram provides a high-level view of the system, showcasing its interactions with external entities and emphasizing CoinSavvy's central role in delivering financial education and managing user interactions.

### 3.10 Sequence diagram

#### 3.10.1 Quiz sequence

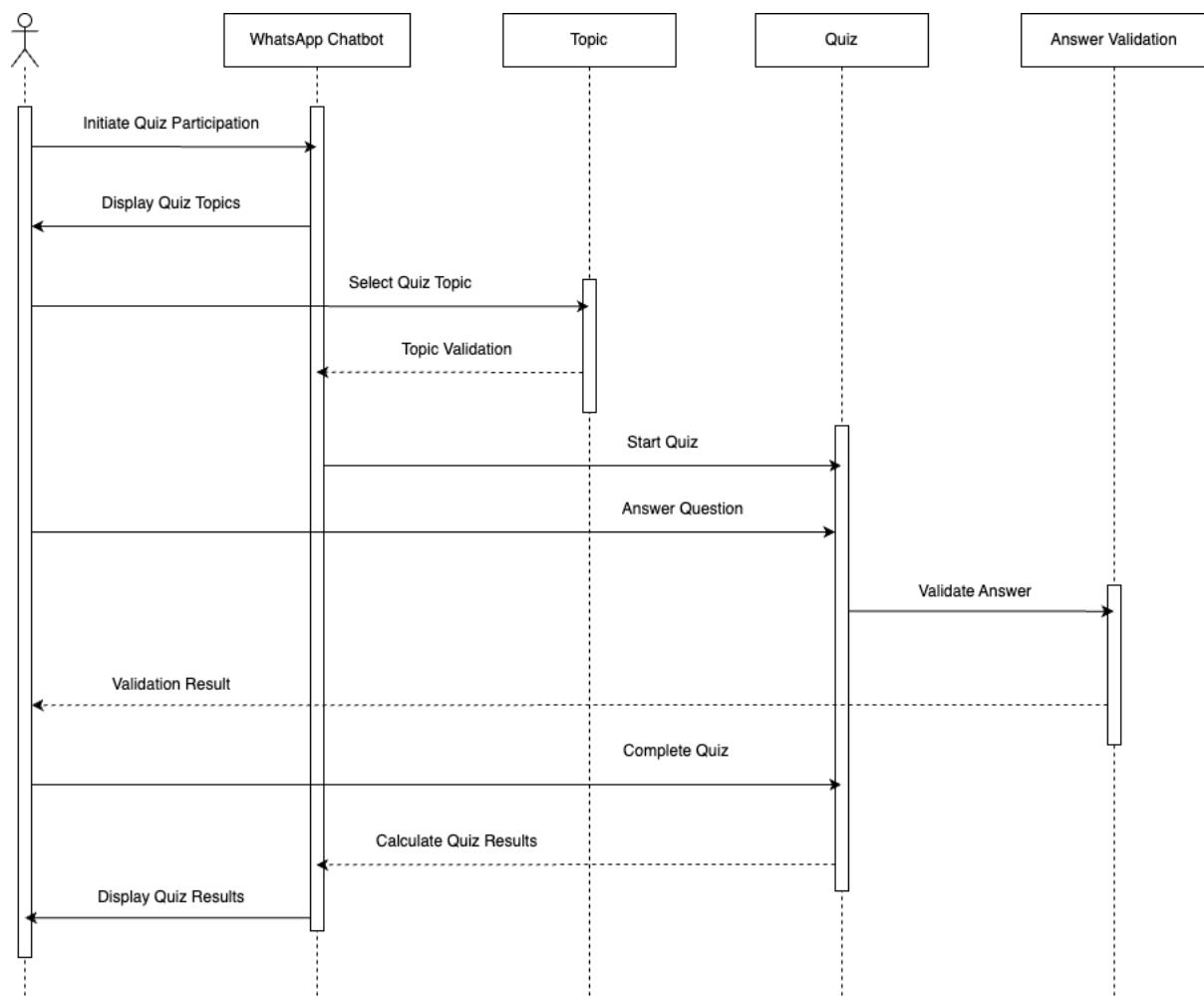
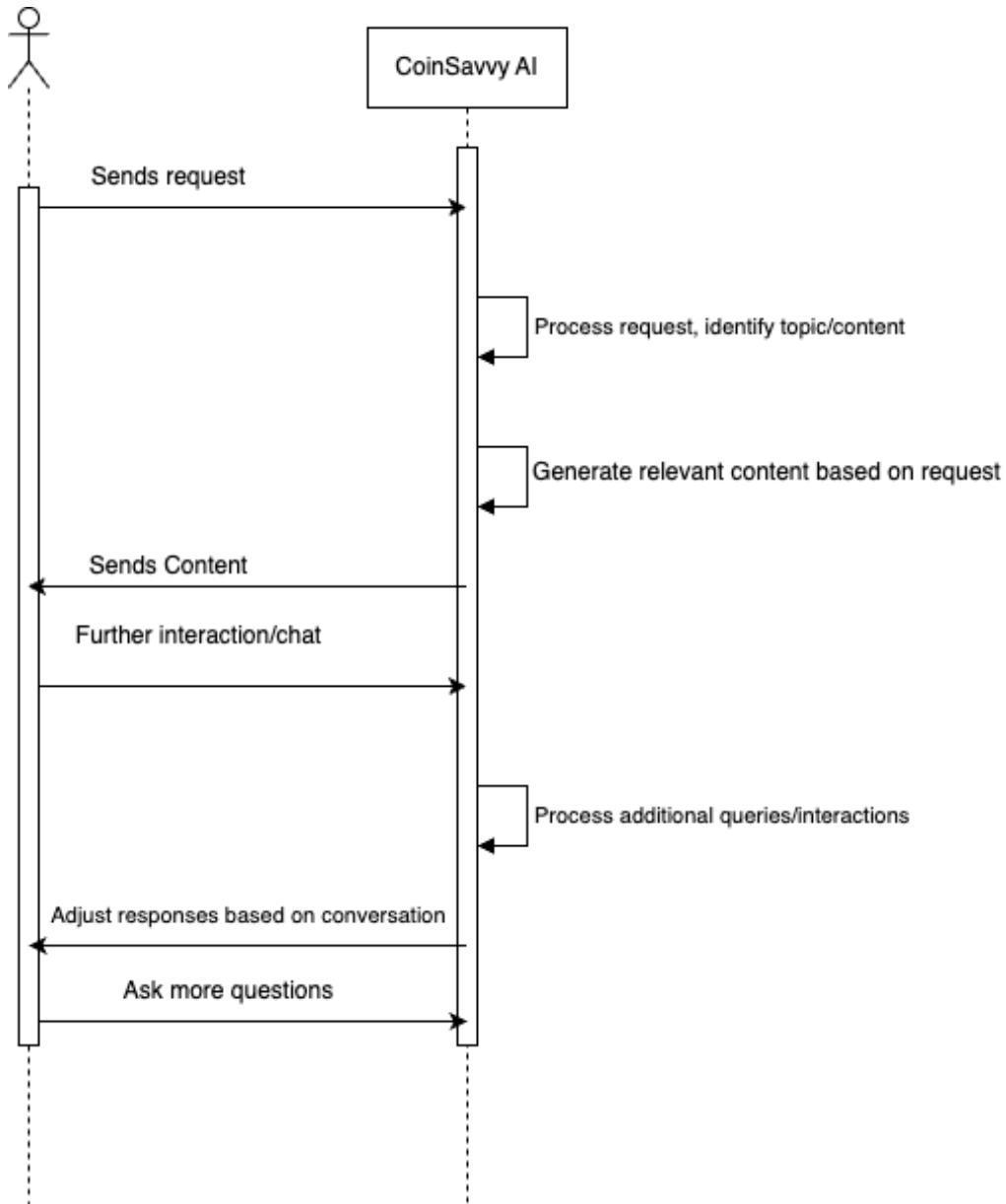


Figure 6: Sequence diagram illustrating the sequence of actions in quiz engagement

The sequence diagram represents the process flow of a user participating in a quiz on the CoinSavvy system. The user initiates the quiz participation, selects a quiz topic, and starts the quiz. The user answers each question, and the system validates the answers. This process is repeated for each question until the user completes the quiz. Finally, the system calculates and displays the quiz results to the user.

### 3.10.2 Language model sequence



*Figure 7: Sequence diagram illustrating interaction with language model*

The sequence diagram illustrates the procedural flow of a user interacting with the CoinSavvy AI. It begins with the user sending a request to the CoinSavvy AI, prompting the system to process the request. The system then identifies the topic or content of the user's request. Following this, the CoinSavvy AI generates relevant content based on the user's query. The system sends the generated content back to the user, facilitating further interaction or clarification. This conversational loop may continue with the user posing additional queries, and the system responding accordingly to enhance the user experience.

## CHAPTER 4 – IMPLEMENTATION

### 4.1 PROGRAMMING LANGUAGE

Python was the main programming language used to create the "CoinSavvy" app because of its adaptability, strong ecosystem, and aptitude for data processing, web development, and artificial intelligence applications[9]. The choice of Python was influenced by several important factors:

**Flexibility and Versatility:** Python's versatility makes it useful in a variety of fields, such as data processing, artificial intelligence, and web development. The implementation of the numerous features needed for the "CoinSavvy" application, such as integrating with WhatsApp, handling user inquiries, and producing responses using specially designed AI models, depended heavily on this flexibility[10].

**Rich Ecosystem of Frameworks and Libraries:** The "CoinSavvy" project benefited greatly from the vast ecosystem of Python frameworks and libraries[9][10]. Frameworks such as Django and Flask enabled the web application to be developed quickly, and libraries like Twilio, Langchain, OpenAI and more enabled the installation of other features, like AI capabilities and WhatsApp integration.

**Readability and Ease of Learning:** Python's readability and clear, straightforward syntax made it suitable for developers of all skill levels. Furthermore, Python's clarity made complicated techniques and solutions easier to understand, facilitating easy solo development[9].

**Community Support and Documentation:** Python is fortunate to have a thriving and helpful developer community that helps to spread information and enhance the language continuously. The copious documentation that is accessible for Python modules and frameworks functioned as invaluable resources, offering direction and support during the entirety of the development procedure[9].

### 4.2 FRAMEWORKS FOR API DEVELOPMENT

API frameworks are essential for developing online applications because they offer the structures and tools needed to design, implement, and communicate with Application Programming Interfaces (APIs)[11]. Two main API frameworks were chosen: Flask and Django, for the AI chatbot and WhatsApp-based interaction finance application

#### 4.2.1 Django Framework

The "CoinSavvy" project harnesses the Django framework extensively to construct a robust backend system for its WhatsApp-based finance quiz application and AI chatbot. Here's an in-depth look at how Django is utilized across various aspects of the project:

##### 4.2.1.1 Project Creation and App Setup

Django's command-line utility **django-admin** or **manage.py** facilitates the creation of projects and apps[12]. *To initiate the project, the following command is executed:*

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

```
django-admin startproject coinsavvy_project
```

Figure 8:Django project creation

This command creates a directory named **coinsavvy\_project** containing the initial project structure, including configuration files like **settings.py**, **urls.py**, and **wsgi.py**[12].

*To add an app to the project, developers execute:*

```
python manage.py startapp coinsavvy_app
```

Figure 9:Django app creation

This command generates a directory named **coinsavvy\_app** within the project directory, encapsulating components such as models, views, and tests.

### 4.2.1.2 File Structure in Django

1. **Settings.py**: This file, located within the project directory, consolidates project-wide settings such as database configuration, middleware, static files, and installed applications.
2. **Urls.py**: Also residing within the project directory, this file defines URL patterns and routes requests to appropriate views within the project.
3. **Wsgi.py and Asgi.py**: These files, found within the project directory, facilitate running the Django application on WSGI and ASGI servers, respectively.
4. **Models.py**: Located within each app directory, the **models.py** file defines database models using Django's ORM system. It encapsulates the data structure and relationships between different entities.
5. **Migrations Directory**: Nested within each app directory, the **migrations** directory manages database schema changes using Django's migration system. Developers generate migrations using the following commands:

```
python manage.py makemigrations  
python manage.py migrate
```

Figure 10:Making migrations in Django

These commands create migration files based on changes in models and apply them to the database, ensuring schema consistency.

6. **Views.py**: Also residing within each app directory, the **views.py** file contains view functions responsible for handling HTTP requests, interacting with models, and generating responses.
7. **Admin.py, Apps.py, Tests.py**: These files, located within each app directory, handle admin configuration, app configuration, and test cases, respectively.
8. **Manage.py**: The **manage.py** script, located at the root of the project directory, serves as a command-line utility for interacting with the Django project. It allows tasks such as running the development server, creating applications, applying migrations, and more. The command below is used to start the Django server.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

```
python manage.py runserver
```

Figure 11: Executing the Django server

### 4.2.1.3 Integration with External APIs

Django facilitates seamless integration with external APIs, enabling the "CoinSavvy" project to interact with third-party services such as Twilio and the custom built AI API. The integration with the Twilio API allows the application to send and receive messages via WhatsApp, while integration with the AI test API enhances the bot's capabilities in providing informative responses to user queries. Django's flexibility and extensibility make it easy to integrate external APIs into the application and leverage their functionalities to enrich the user experience[13].

### 4.2.1.4 Database Management

Django's built-in ORM (Object-Relational Mapping) system simplifies database management by abstracting away the complexities of SQL queries. The "CoinSavvy" project utilizes Django models to define the data structure and relationships between different entities, such as users, quiz questions, quiz responses, and phone number. These models are defined in the **models.py** file within each app and provide a high-level Pythonic interface for interacting with the database[14].

## 4.2.2 Quiz Functionality Description

The quiz functionality in the "CoinSavvy" project is seamlessly and directly integrated into the Django framework, specifically within the **coinsavvy\_app** application. Here's how the quiz is implemented across the Django project:

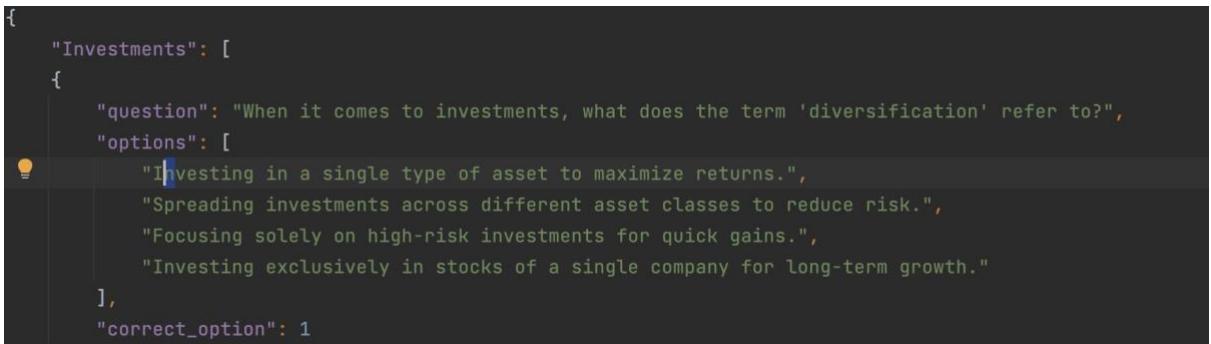
### 4.2.2.1 Django App Structure

1. **Quiz App Creation:** Within the Django project, a dedicated app named **coinsavvy\_app** is created to manage quiz-related functionalities.
2. **Models:** The **models.py** file within the **coinsavvy\_app** directory defines Django models representing quiz questions, options, and user responses. These models specify the database schema necessary for storing quiz-related data.
3. **Views:** In the **views.py** file of the **coinsavvy\_app**, view functions are implemented to handle quiz-related HTTP requests. These views are responsible for loading quiz questions, processing user answers, calculating scores, and providing feedback.

### 4.2.2.2 Quiz Logic Implementation

1. **Loading Questions:** The **load\_questions()** function loads quiz questions from a JSON file within the project directory.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP



```
{  
    "Investments": [  
        {  
            "question": "When it comes to investments, what does the term 'diversification' refer to?",  
            "options": [  
                "Investing in a single type of asset to maximize returns.",  
                "Spreading investments across different asset classes to reduce risk.",  
                "Focusing solely on high-risk investments for quick gains.",  
                "Investing exclusively in stocks of a single company for long-term growth."  
            ],  
            "correct_option": 1  
        }  
    ]  
}
```

Figure 12: JSON file containing questions

2. **Displaying Questions:** The `ask_next_question()` function retrieves the next question from the loaded dataset and constructs a message containing the question and options.
3. **Processing Answers:** The `process_answer()` function processes user answers, validates them against the correct option, updates the user's score, and provides feedback on the correctness of the answer.
4. **User Interaction:** Users interact with the quiz bot by sending messages via WhatsApp. The bot processes these messages, presents questions, collects answers, and provides feedback in real-time.

The quiz functionality in the "CoinSavvy" project is directly integrated into the Django framework (i.e Views.py), providing users with an interactive and engaging experience directly within the WhatsApp chat interface. By leveraging Django's models, views, templates, and logic, the quiz feature enhances user engagement and promotes financial literacy effectively. This integration contributes to one aspect of the overall objectives of the "CoinSavvy" project by empowering users to improve their financial knowledge and decision-making skills through self-assessment and knowledge acquisition.

### 4.2.3 Flask framework

Flask was selected to provide an API for communicating with the custom built language model for the "CoinSavvy" project. Flask is a good choice for this because of its simplicity and small weight[15]. Let's see how the API was constructed using Flask and incorporated into the Django project:

#### 4.2.3.1 Why Flask?

Flask is renowned for its lightweight and minimalistic design. As a micro-framework, Flask provides just the essentials for building web applications, making it efficient for creating APIs without unnecessary overhead[16]. This aligns perfectly with the project's goal of keeping the AI API lightweight and focused on its core functionality.

#### 4.2.3.2 Flask API Implementation

The Flask API is implemented with a single route `/query` to handle POST requests containing JSON data with a 'query' field. This route is decorated with `@app.route('/query', methods=['POST'])`, indicating that it will handle HTTP POST requests sent to the '/query'

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

endpoint[17]. Upon receiving a query, the API extracts the query data from the request JSON, executes the finance AI logic using the language model, and returns the response in JSON format using `jsonify()`.

```
from flask import Flask, jsonify, request

app = Flask(__name__)

@app.route('/query', methods=['POST'])
def query():
    data = request.get_json()
    query = data['query']

    # Execute the finance AI
    response = agent_executor({"input": query})

    return jsonify({"response": response['output']})

if __name__ == '__main__':
    app.run(debug=True)
```

Figure 13: Flask End-Point in Language Model Code

The above sample code snippet indicates how flask is used to receive and respond to requests from the Django project to the language model for processing and response.

### Integration with Django Views

In the Django project, the `ai_test_api()` function is implemented within the views to make API calls to the Flask API. This function constructs a POST request to the Flask API endpoint `/query` with the user's query data. Upon receiving the response from the Flask API, it extracts the 'response' field from the JSON and returns it.

```
import requests

def ai_test_api(query):
    url = 'http://127.0.0.1:5000/query'  # Update with your API URL
    data = {'query': query}
    try:
        response = requests.post(url, json=data)
        response.raise_for_status()  # Raise an exception for HTTP errors
        result = response.json()
        return result['response']
    except Exception as e:
        return f"Error: {str(e)}"

query_response = ai_test_api(incoming_message)
msg.body(query_response)
```

Figure 14: Flask End-Point in Django Views.py

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

The above sample code snippet indicates how the views.py in the Django project forwards incoming message from WhatsApp as queries to the Finance AI through the API built using flask.

### Conclusion

The utilization of Flask as the API framework for the language model in the "CoinSavvy" project was driven by its lightweight nature and simplicity, which perfectly suited the project's requirements. Given that both the Django project and the AI reside on the same machine but in different locations, Flask provided an ideal solution for building the API. Its lightweight design allowed for efficient communication between the components without adding unnecessary overhead. By integrating Flask with the Django project, the application achieved seamless communication with the language model, enabling users to interact with the AI component via WhatsApp. This integration not only enhanced the functionality and user experience of the application but also maintained efficiency and focus on core features. Overall, Flask proved to be a reliable choice for building the API, facilitating smooth interaction between different components of the "CoinSavvy" project.

### 4.3 DATABASE

In the "CoinSavvy" project, a MySQL database is utilized via Django's ORM. The database stores user data captured by the **User** model, including quiz progress, scores, username and survey responses. Django's manage.py utility manages database operations such as migrations for seamless integration of model changes.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view shows two databases: 'bot1' and 'csdb'. The 'bot1' database is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. A query editor window on the right contains the following SQL code:

```
1 • show databases;
2 • show tables in bot1;
3 • select * from financebot_user;
```

Below the code, the 'Result Grid' shows a table with the following data:

ID	PHONE_NUMBER	TOPIC	CURRENT_QUESTION_IN...	SCORE	TIMESTAMP	SURVEY_RESPONSE
35	whatsapp:+234███████████	Investments	0	0	2023-09-14 13:30:02.897595	NULL
36	whatsapp:+234███████████	Banking Fees	2	2	2023-09-14 13:30:49.579097	NULL
37	whatsapp:+234███████████	Banking Fees	2	0	2023-09-14 13:30:51.934497	NULL
38	whatsapp:+234███████████	Investments	7	5	2023-09-14 13:30:52.130077	NULL
39	whatsapp:+234███████████	Banking Fees	4	2	2023-09-14 13:31:10.264176	3
40	whatsapp:+234███████████	Banking Fees	4	4	2023-09-14 13:31:42.703116	NULL
41	whatsapp:+234███████████	Identity Theft	1	1	2023-09-27 16:29:25.906270	NULL
42	whatsapp:+447███████████	Investments	1	0	2023-10-09 10:24:04.846851	NULL
43	whatsapp:+234███████████	Identity Theft	0	0	2023-10-09 10:31:32.231510	NULL
44	whatsapp:+234███████████	Interest	10	6	2023-10-30 22:09:58.676679	4

Figure 15: Snippet of sample data from quiz engagement

### 4.4 TUNNELING

**Ngrok** played a crucial role in enabling external access to the Django project during development, while Twilio facilitated access to the WhatsApp interface. Let's delve into how Ngrok was utilized for tunneling and facilitating access to the Django project, while Twilio was used for WhatsApp integration:

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 4.4.2 Ngrok Setup for Django Project

Ngrok was employed to expose the Django project to the internet during development. By creating a tunnel to the local Django server, Ngrok provided external access to the project, facilitating testing and demonstration of functionalities to stakeholders and collaborators[18]. This allowed for real-time feedback and validation of features without the need for deploying the project to a remote server. Through this setup, Ngrok facilitated the tunneling necessary for external access to Twilio's WhatsApp API, enabling interaction with WhatsApp users. This integration allowed the project to send and receive messages from WhatsApp users, thereby enabling the functionality of the finance quiz application and AI chatbot.

```
Last login: Wed Feb 21 23:46:22 on ttys000
(base) ahmed@Ahmeds-MacBook-Pro ~ % ngrok http 8000
```

Figure 16:Starting Ngrok

```
ngrok
Take our ngrok in production survey! https://forms.gle/aXibFWzEA36DudFn6

Session Status      online
Account            Ahmed Darkye (Plan: Free)
Update             update available (version 3.8.0, Ctrl-U to update)
Version            3.3.4
Region             South America (sa)
Latency            291ms
Web Interface     http://127.0.0.1:4040
Forwarding         https://4116-41-218-206-146.ngrok-free.app -> http://localhost:8000

Connections        ttl     opn     rt1     rt5     p50     p90
                   324      0     0.00    0.00   90.32   93.78

HTTP Requests
-----
POST /FYP/          200 OK
```

Figure 17:Ngrok Interface

Ngrok and Twilio played complementary roles in the development of the "CoinSavvy" project. While Ngrok provided external access to the Django project for testing and collaboration purposes, Twilio facilitated integration with the WhatsApp interface, enabling interaction with users[19]. Together, these tools contributed to the success of the project by streamlining development workflows and ensuring widespread accessibility and user engagement.

## 4.5 EXTERNAL APIs AND LIBRARIES

### 4.5.1 Twilio WhatsApp API

Twilio played a pivotal role in the "CoinSavvy" Django project by providing access to the WhatsApp interface, facilitating user interaction with the finance quiz application and AI chatbot[20]. Let's delve into how Twilio was utilized and its integration with Ngrok for seamless communication:

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 4.5.1.1 Configuration and Setup for Twilio[20]:

1. **Twilio Account Creation:** In order to integrate WhatsApp messaging features, the "CoinSavvy" project started with the creation of a Twilio account, which gave access to the required tools and APIs.
2. **WhatsApp API Integration:** The Django project has incorporated Twilio's WhatsApp API, enabling users to send and receive messages from one another within the programme.
3. **Configuration using Ngrok:** Ngrok was used to establish a secure tunnel to the local Twilio server, enabling external access to the Twilio API during development. This configuration made guaranteed that the Django development environment could use Twilio's services.

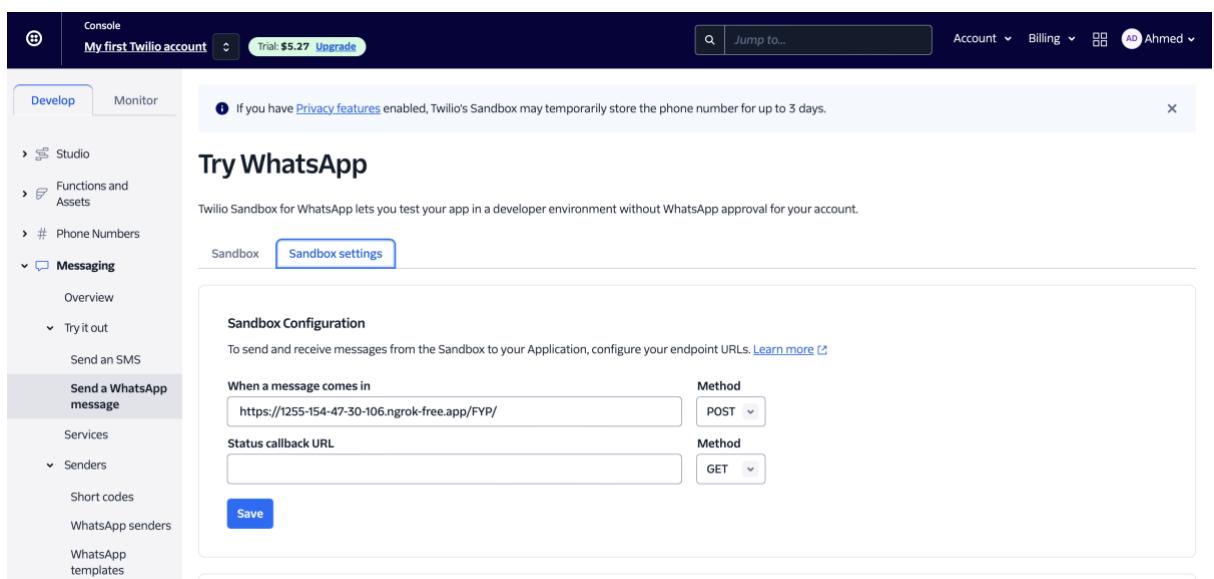


Figure 18: Twilio WhatsApp URL Webhook dashboard

### 4.5.1.2 Utilization in the Project

1. **Messaging Capabilities:** Twilio enabled the Django project to send and receive messages from WhatsApp users. This functionality was essential for implementing interactive quizzes and providing AI-powered responses to user inquiries within the Django application.
2. **User Engagement:** By leveraging Twilio's WhatsApp API, the Django project facilitated direct communication with users on a platform they were familiar with, enhancing user engagement and accessibility.
3. **Real-Time Interaction:** Twilio's API allowed for real-time interaction with users within the Django project, enabling immediate responses to quiz submissions, AI inquiries, and feedback.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 4.5.1.3 Ngrok Integration

1. **Tunneling with Ngrok:** Ngrok was used to create a secure tunnel to the local Twilio server, allowing external access to Twilio's WhatsApp API during Django development.
2. **Ngrok URL Configuration:** The Ngrok-generated URL was configured within the Twilio account settings, ensuring that incoming WhatsApp messages were directed to the local Django development environment seamlessly.

Twilio's integration with the "CoinSavvy" Django project via Ngrok provided a seamless solution for interacting with users on the WhatsApp platform. By leveraging Twilio's WhatsApp API and Ngrok's tunneling capabilities within the Django environment, the project facilitated direct communication with users, enhancing user engagement and enabling the implementation of interactive quizzes and AI chatbot functionalities. Together, Twilio and Ngrok played a crucial role in the success of the "CoinSavvy" Django project by ensuring accessibility and real-time interaction with users on WhatsApp. WhatsApp was chosen as the platform for the "CoinSavvy" project due to its extensive global user base, user familiarity, convenience, real-time communication capabilities, and support for multimedia content. With over 2 billion users worldwide, WhatsApp provides a familiar and accessible interface for users to engage with the finance quiz application and AI chatbot directly from their chats. This seamless integration enhances user engagement and accessibility, ensuring a dynamic and interactive experience for users across different regions and demographics.

### 4.5.2 Audio Processing and Transcription for Speech Recognition (Within Django Project)

The development of the project entailed significant efforts towards enabling the system to process and transcribe audio messages received via WhatsApp for the language model. This functionality was critical for enhancing user engagement through the allowance of natural spoken inputs. The sequence of actions involved in this process includes retrieving the audio from WhatsApp, converting it from its original file format to a format that is more amenable to processing, and, subsequently, transcribing it into text. The objective of this section is to detail the libraries and methods deployed in this process, alongside the solutions devised to overcome the challenges encountered.

#### 4.5.2.1 Retrieval of Audio from WhatsApp

WhatsApp provides a secure mechanism for the transfer of media files, including audio messages[21]. However, accessing these files required proper authentication, a challenge that was overcome through the utilization of the **HTTPBasicAuth** module from the **requests.auth** library. This authentication mechanism proved essential in fetching the audio content securely from WhatsApp's servers, as illustrated in the following snippet:

```
import requests
from requests.auth import HTTPBasicAuth
```

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

```
# Fetch the audio content with proper authentication
audio_response = requests.get(audio_url, auth=HTTPBasicAuth(account_sid, auth_token))

if audio_response.status_code == 200:
    audio_content = audio_response.content
    incoming_message = transcribe_audio(audio_content)
else:
    print("Failed to download audio file. Status code:", audio_response.status_code)
```

Figure 19: Successfully fetching audio from WhatsApp

**audio\_url** contains the URL to the WhatsApp audio file, and **account\_sid** and **auth\_token** serve as credentials for authentication with Twilio, the service handling the WhatsApp bot[21]. Successful fetching of the audio content was crucial for the subsequent stages of the process.

### 4.5.2.2 Audio Conversion from OGG to WAV

WhatsApp audio files are typically in the OGG format, which needed to be converted to WAV for easier processing and transcription. To achieve this, the **AudioSegment** class from the **pydublibrary** was used[22]. This choice was prompted by **pydub**'s straightforward API and support for a wide range of audio file formats. Here is how the conversion was implemented:

```
from pydub import AudioSegment
import os
from tempfile import NamedTemporaryFile

with NamedTemporaryFile(suffix=".ogg") as temp_ogg:
    temp_ogg.write(audio_content)
    temp_ogg.flush() # Ensure content is written
    os.fsync(temp_ogg.fileno()) # Ensure written to disk
    temp_ogg.seek(0) # Go to the start of the file

    with NamedTemporaryFile(suffix='.wav') as temp_wav:
        audio = AudioSegment.from_ogg(temp_ogg.name)
        audio.export(temp_wav.name, format="wav")
        temp_wav.seek(0) # Ensure the pointer is at the start
```

Figure 20: Conversion of audio from OGG to WAV

This code converts the audio from the original OGG format to WAV by first writing the downloaded audio content to a temporary OGG file, then converting and exporting it to another temporary WAV file. The use of temporary files ensures that system storage is not unnecessarily occupied or cluttered with transient data.

### 4.5.2.3 Transcribing Audio to Text[23]

The transcribed text is obtained from the WAV file using the **speech\_recognition** library, particularly leveraging the **Recognizer** class and its **recognize\_google** method for speech

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

recognition. This library was chosen for its accuracy and ease of integration. Here's how the transcription is performed:

```
import speech_recognition as sr\n\nr = sr.Recognizer()\nwith sr.AudioFile(temp_wav.name) as source:\n    audio_data = r.record(source)\n    text = r.recognize_google(audio_data)\nreturn text
```

Figure 21: Transcription of WAV file

In this stage, the WAV file is read, and the audio data is fed into Google's speech recognition engine via the **recognize\_google** method, resulting in the conversion of spoken language into text.[23] This text is then further processed according to the application's logic.

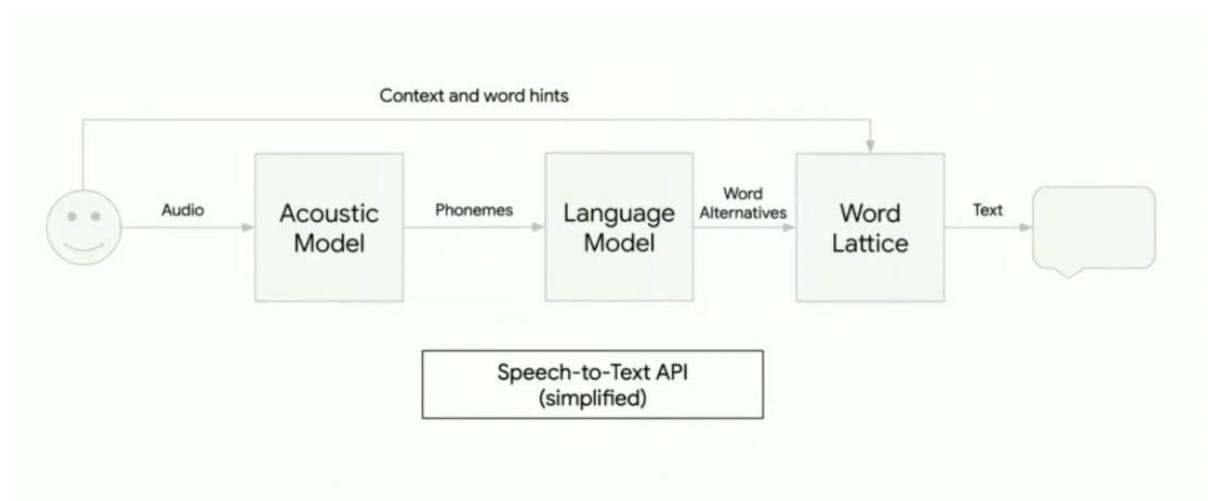


Figure 22: Simplified Illustration of how speech to text conversion works[24]

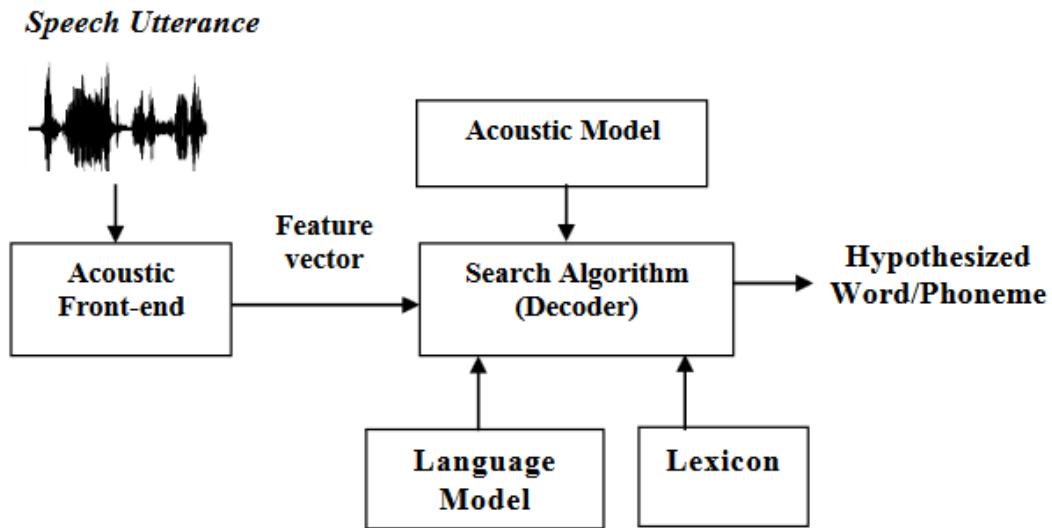


Figure 23:A detailed illustration of speech-to-text conversion using Google Cloud Platform API[25]

#### 4.5.2.4 Integration with Application Logic

The transcribed text undergoes processing by the application's logic. For this project, the process involved passing the text to a local AI API developed using Flask. The API executes specific actions based on the content of the text, enhancing the interaction between users and the AI. This was accomplished through a simple HTTP POST request:

```

def ai_test_api(query):
    url = 'http://127.0.0.1:5000/query' # Update with your API URL
    data = {'query': query}
    try:
        response = requests.post(url, json=data)
        response.raise_for_status() # Raise an exception for HTTP errors
        result = response.json()
        return result['response']
    except Exception as e:
        return f"Error: {str(e)}"
  
```

Figure 24:Flask End-Point in Django Views.py

In this code snippet, **query** represents the text transcribed from the user's spoken words, encapsulating the user's intent. The API's response is seamlessly integrated into the WhatsApp bot's response, offering a refined user experience.

## Conclusion

The implementation of audio processing and transcription in the project involved resolving several challenges, notably related to media retrieval, format conversion, and accurate transcription. The deployment of libraries such as **requests**, **pydub**, and **speech\_recognition** provided robust solutions to these challenges. Through efficient combination of these technical components, the project was successful in facilitating feature-rich user interaction through natural language processing, showcasing the potential of

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

integrating advanced audio processing and AI to enhance user engagement and experience in messaging applications.

### 4.5.3 LangChain – Finance AI Development

Langchain is a robust framework for creating, training, and deploying models for various processing tasks which cannot be performed with native language models[26]. It offers flexibility and scalability, enabling developers to construct complex language models with relative ease[27]. Langchain allows developers to fine-tune models by allowing them to perform specific tasks while also gaining access to various resources through its own libraries and API services. In this project, Langchain serves as the primary tool for building the language model. The model is trained using the OpenAI API pre-trained models, and Langchain's various modules and classes are utilized for multiple tasks to give the language model a wide range of functionality.

#### 4.5.3.1 Tools and Library Integration:

The project seamlessly integrates several APIs alongside LangChain libraries to bolster its capabilities:

```
from langchain_community.utilities.serpapi import SerpAPIWrapper
from langchain_openai import ChatOpenAI, OpenAIEmbeddings
import PyPDF2
from flask import Flask, request, jsonify
from langchain.chains import RetrievalQA
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain_community.vectorstores.faiss import FAISS
from langchain.storage import LocalFileStore
from langchain.embeddings import CacheBackedEmbeddings
from langchain.agents import Tool, OpenAIMultiFunctionsAgent, OpenAIFunctionsAgent, AgentExecutor
from langchain.prompts import MessagesPlaceholder, PromptTemplate
from langchain.memory import ConversationBufferMemory
import os
from langchain.schema import SystemMessage
```

Figure 25: Language Model Libraries

#### 1. SerpAPIWrapper (SerpAPI):

- **Functionality:** SerpAPIWrapper is an API wrapper for accessing AI-powered search engine capabilities provided by SerpAPI[28].
- **Role in the Project:** SerpAPI enriches the language model's knowledge base with real-time financial information obtained from web searches. It enhances the model's responsiveness and ensures that the application remains updated with the latest financial data.

#### 2. ChatOpenAI:

- **Functionality:** ChatOpenAI is a module for interacting with OpenAI's language models.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

- **Role in the Project:** ChatOpenAI facilitates interactions with the OpenAI language model, allowing the application to generate responses to user queries related to finance.

### 3. OpenAIEmbeddings:

- **Functionality:** OpenAIEmbeddings provides functionalities for utilizing OpenAI's embeddings[29].
- **Role in the Project:** OpenAIEmbeddings enables the representation of text data in a high-dimensional vector space, enhancing the model's performance in various NLP tasks, such as document retrieval and similarity search.

### 4. PyPDF2:

- **Functionality:** PyPDF2 is a Python library for handling PDF files[30].
- **Role in the Project:** PyPDF2 is crucial for extracting text from financial documents in PDF format. It preprocesses these documents, making their content accessible for analysis by the language model.

### 5. Flask:

- **Functionality:** Flask is a lightweight web application framework for Python[15].
- **Role in the Project:** Flask is used to create the user interface of CoinSavvy, enabling users to interact with the language model through the Django project. It facilitates smooth communication between users and the application.

### 6. RetrievalQA:

- **Functionality:** RetrievalQA is a module for handling question answering tasks based on documents[31].
- **Role in the Project:** RetrievalQA integrates the language model with a retriever, allowing the model to retrieve relevant information from the document database and provide accurate answers to user queries.

### 7. RecursiveCharacterTextSplitter:

- **Functionality:** RecursiveCharacterTextSplitter is a module for splitting text into smaller chunks[32].
- **Role in the Project:** It is used to split large text documents into manageable chunks for processing by the language model, enhancing efficiency and performance.

### 8. FAISS (Facebook AI Similarity Search):

- **Functionality:** FAISS is a library for similarity search and clustering of dense vectors[33].
- **Role in the Project:** FAISS is utilized to create a vector database from documents and embeddings, enhancing document retrieval functionalities within the application.

### 9. LocalFileStore:

- **Functionality:** LocalFileStore is a class provided by LangChain for setting up a local file storage system[34].
- **Role in the Project:** LocalFileStore is used to store embeddings generated by the language model, ensuring quick access and retrieval, thereby optimizing the model's performance.

### 10. CacheBackedEmbeddings:

- **Functionality:** CacheBackedEmbeddings is a class provided by LangChain for caching embeddings to improve performance[35].
- **Role in the Project:** It aids in caching embeddings to reduce redundant computations, thereby enhancing the efficiency of the language model.

### 11. AgentExecutor:

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

- **Functionality:** AgentExecutor is a module for executing agents within the LangChain framework[36].
- **Role in the Project:** AgentExecutor is used to execute various agents and tools integrated into the finance AI project, facilitating seamless interaction and operation of the system.

### 12. ConversationBufferMemory:

- **Functionality:** ConversationBufferMemory enables the storage and retrieval of conversational history[37].
- **Role in the Project:** It facilitates continuity and context preservation in user interactions by storing and retrieving conversational history within the application.

### 13. SystemMessage:

- **Functionality:** SystemMessage is a schema module within LangChain[38].
- **Role in the Project:** It provides a structured format for system messages, enhancing communication and coordination within the application.

These are some of the libraries which collectively contribute to the functionality, efficiency, and effectiveness of the finance AI project, enabling CoinSavvy to provide users with accurate, timely, and insightful responses to finance-related queries.

#### 4.5.3.2 Chains in LangChain:

LangChain's architecture encompasses chains, representing sequential operations executed by the language model[38]. In this project, two primary chains are defined:

- **RetrievalQA Chain:** Tasked with answering questions based on uploaded documents, this chain seamlessly integrates the language model with a retriever, facilitating the retrieval of pertinent information from the document database[31].
- **Summary Chain:** This chain condenses the uploaded documents related to financial topics into concise summaries. Leveraging the language model, it enhances the accessibility of financial information within CoinSavvy[31].

#### 4.5.3.4 OpenAI API and Embeddings:

The project harnesses the OpenAI pre-trained models, leveraging both ChatGPT 3.5 Turbo and ChatGPT 4.0 depending on usage to furnish the language model with advanced natural language understanding capabilities. Additionally, OpenAI embeddings are employed to represent text dataset in a high-dimensional vector space, augmenting the model's performance in tasks such as document retrieval and similarity search.

In essence, LangChain forms the backbone of the finance AI, orchestrating the integration of tools, chains, and APIs to furnish CoinSavvy with a robust and efficient language model tailored for addressing finance-related inquiries and furnishing users with invaluable insights.

This comprehensive integration of tools and components underscores LangChain's versatility and effectiveness in building sophisticated language models for natural language processing tasks. Through seamless integration and optimization, CoinSavvy emerges as a powerful platform for accessing and analyzing financial information, empowering users with actionable insights and informed decision-making capabilities.

### 4.5.3.5 Agents in LangChain

Within the LangChain framework, an "agent" is configured to serve as a crucial coordinator, supervising the operation of several tools and chains. It plays a crucial role in controlling the flow of work and information, guaranteeing the smooth operation of the language model and related features[38].

#### Functions of an Agent in LangChain[38]:

1. **Tool Management:** The agent oversees the activation and utilization of different tools created and integrated into the system. It discerns the specific tools required based on incoming queries or tasks, ensuring that the appropriate functionalities are invoked.
2. **Chain Execution:** Agents are tasked with initiating and overseeing the execution of chains within the LangChain framework. They play a crucial role in determining which chains are best suited to handle specific types of tasks or queries and ensuring their proper execution.
3. **Data Flow Control:** Agents regulate the flow of data between different components of the system, including tools, chains, and external data sources. They oversee the efficient processing and transmission of data to facilitate analysis or action.
4. **Context Preservation:** Based on specific configurations, agents maintain context from previous interactions to ensure coherence and continuity in conversations or tasks. They store relevant information to facilitate follow-up interactions and provide personalized responses tailored to the user's context.
5. **Error Handling:** Agents are equipped to handle exceptions or errors that may arise during task execution. They implement robust error-handling mechanisms to gracefully manage unexpected situations, ensuring the stability and reliability of the system.
6. **Performance Optimization:** Agents monitor system performance and resource usage, prioritizing tasks and implementing optimization strategies to enhance efficiency and responsiveness. They play a crucial role in maintaining optimal system performance under varying workloads.

### 4.5.3.6 Function within the CoinSavvy AI Development:

The agent acts as the central component of the LangChain-powered finance AI project, coordinating the execution of tools and chains to provide the required functions. It is essential in handling user inquiries about finance, making use of the capabilities of multiple chains and instruments to deliver precise and perceptive answers.

Depending on the type of user question, the agent might, for example, enable features like document summarization or real-time data retrieval. After that, it coordinates the execution of related chains to handle the input, obtain pertinent data, and produce the necessary answers. However, a set of tools need to be configured from the libraries and then assigned to the agent. This is what the agent works with. There are currently three tools in CoinSavvy AI:

1. finance expert: This tool provides expert advice on topics related to finance based on custom dataset together with pre-existing model knowledge.
2. finance summarizer: This tool can summarize topics based on dataset (documents and articles) related to financial topics.
3. finance search: This tool can provide real-time information on finance-related queries.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

The Snippets below will indicate the agent in CoinSavvy AI utilizing the search engine tool to search for real time information. First snippet displays what the user sees and second displays what the AI is processing and doing in the backend.

```
/Users/ahmed/CoinSavvyAI/venv/bin/python /Users/ahmed/CoinSavvyAI/API.py
User: what is the current price of Apple Stock
Response: The current price of Apple Inc. stock (AAPL) is $172.62 USD. The previous close price was $173.00 USD.
```

```
> Entering new AgentExecutor chain...
Invoking: 'finance_search' with 'Apple Stock price'

{'type': 'finance_results', 'title': 'Apple Inc', 'exchange': 'NASDAQ', 'stock': 'AAPL', 'currency': 'USD', 'price': 172.62, 'previous_close': 173.0}The current price of Apple Inc. stock (AAPL) is $172.62 USD. The previous close price was $173.00 USD.

If you have any more questions about Apple stock or if you need information on other stocks or financial topics, feel free to ask!
> Finished chain.
```

Figure 26: Illustration of Language Model using Web Search

The Snippets below will indicate the agent in CoinSavvy AI utilizing the summarizing tool to summarize information from the custom dataset. First snippet displays what the user sees and second displays what the AI is processing and doing in the backend.

```
User: Brief me about investments
Response: Investing involves making choices between being a lender or an owner of assets. As a lender, you can invest in options like bank CDs, treasury bills, or corporate bonds. Diversification is crucial in investing to reduce the volatility in the value of your portfolio and achieve a higher rate of return for a given level of risk. Mutual funds are a popular way to diversify investments, as they pool your money with others and invest in a variety of securities. When starting to invest, consider paying off high-interest debt first and then explore different investment options based on your time frame and risk tolerance. Short-term investments are ideal for a few years, intermediate-term for more than a few but less than ten years, and long-term investments for a decade or more. Bonds and bond funds are recommended for those looking for stability and potential returns. Investing involves making choices between being a lender or an owner of assets. As a lender, you can invest in options like bank CDs, treasury bills, or corporate bonds. However, there are risks associated with lending investments, such as not getting what you were promised or losing part of your original investment in case of bankruptcy. If you have more specific questions about investments or need further details on a particular investment type, feel free to ask!
```

```
> Entering new AgentExecutor chain...
Invoking: 'finance_summarizer' with 'Investments'

In the world of investments, there are two major choices: being a lender or an owner. When you lend your money, you are essentially a lender and can invest in options like bank CDs, treasury bills, or corporate bonds. However, there are risks associated with lending investments, such as not getting what you were promised, losing part of your original investment in case of bankruptcy, or facing a decrease in purchasing power due to inflation. Diversification is key in investing, as it reduces the volatility in the value of your portfolio and allows you to achieve a higher rate of return for a given level of risk. Mutual funds are a great way to diversify your investments, as they pool your money with others and invest in a variety of securities. When starting to invest, it's important to consider paying off high-interest debt first and then looking into different investment options based on your time frame and risk tolerance. Short-term investments are ideal for a few years, intermediate-term for more than a few but less than ten years, and long-term investments for a decade or more. Bonds and bond funds are recommended for those looking for stability and potential returns. Investing involves making choices between being a lender or an owner of assets. As a lender, you can invest in options like bank CDs, treasury bills, or corporate bonds. However, there are risks associated with lending investments, such as not getting what you were promised or losing part of your original investment in case of bankruptcy. Diversification is crucial in investing to reduce the volatility in the value of your portfolio and achieve a higher rate of return for a given level of risk. Mutual funds are a popular way to diversify investments by pooling money with others and investing in a variety of securities. When starting to invest, consider paying off high-interest debt first and then explore different investment options based on your time frame and risk tolerance. Short-term investments are suitable for a few years, intermediate-term for more than a few but less than ten years, and long-term investments for a decade or more. Bonds and bond funds are recommended for those seeking stability and potential returns. If you have more specific questions about investments or need further details on a particular investment type, feel free to ask!
> Finished chain.
```

Figure 27: Illustration of Language Model summarizing a topic

This comprehensive integration of tools and components underscores LangChain's versatility and effectiveness in building sophisticated language models for natural language processing tasks. Through seamless integration and optimization, CoinSavvy emerges as a powerful platform for accessing and analyzing financial information, empowering users with actionable insights and informed decision-making capabilities.

### 4.6 LLM DATASET

In the AI development, a hybrid approach is implemented, leveraging both GPT-3 and GPT-4, along with a custom dataset, to enhance the language model's performance and domain specificity. Here's a comprehensive overview of how these components are integrated:

#### 4.6.1 Integration of GPT-3 and GPT- 4:

- The project incorporates both GPT-3 and GPT- 4 to capitalize on their respective strengths. GPT-3, with its extensive pre-trained knowledge and language understanding capabilities, provides a robust foundation for general language tasks and finance-related queries.
- Additionally, GPT- 4 is integrated to leverage its advanced features and improvements over previous versions, enhancing the model's ability to generate more nuanced and contextually relevant responses within the finance domain.
- By utilizing both models concurrently, the project benefits from the complementary strengths of each(i.e using the appropriate model for varying tasks), resulting in a more comprehensive and robust language model for addressing finance-related inquiries.

#### 4.6.2 Incorporation of Custom Dataset:

- Alongside GPT-3 and GPT- 4, a custom dataset consistingof finance-specific content is integrated into the project. This dataset includes curated and annotated finance-related documents, articles, reports, and other textual resources relevant to the project's domain.
- The custom dataset serves to fine-tune and augment the pre-trained knowledge of GPT-3 and GPT- 4, enabling the models to better understand finance-specific terminology, contexts, and nuances.
- By training the models on the custom dataset, the project tailors their capabilities specifically to finance-related tasks, enhancing their accuracy and relevance in generating responses to finance-related queries.

#### 4.6.3 Hybrid Approach for Enhanced Performance:

- The adoption of a hybrid approach, combining pre-trained language models with a custom dataset, allows for a synergistic enhancement of the language model's performance.
- While GPT-3 and GPT-4 provide a strong foundation of general language understanding, the custom dataset further refines and specializes the models' knowledge within the finance domain.
- This hybrid approach enables the language model to leverage both pre-existing knowledge and domain-specific insights, resulting in more accurate, contextually relevant, and informative responses to finance-related queries.

#### 4.6.4 Data Extraction:

The project employs a variety of tools and libraries, including PyPDF2 and SerpAPIWrapper, to extract pertinent financial data from diverse sources such as PDF documents, websites, and online databases.

**COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP**

- **PyPDF2:** This library is utilized to extract text from PDF documents, which often contain valuable financial information like reports, articles, and statements. Through PyPDF2, the project accesses and extracts textual data embedded within PDF files, enabling further analysis[30].
  - **SerpAPIWrapper:** Leveraging web scraping techniques, the project fetches real-time financial data from websites and online sources[28]. SerpAPIWrapper facilitates the extraction of dynamic content, enriching the language model's knowledge base with up-to-date information crucial for addressing user queries.

#### 4.6.5 Data Processing:

Once extracted, the financial data undergoes thorough processing to prepare it for consumption by the language model.

- **Preprocessing:** The extracted data is preprocessed to remove noise, irrelevant information, and formatting inconsistencies. Techniques such as tokenization, sentence segmentation, and normalization are applied to ensure the data is clean and structured, facilitating effective analysis by the language model.
  - **Representation:** To enhance the model's understanding of the financial domain, embeddings are employed to represent the textual data in a high-dimensional vector space[29]. OpenAIEmbeddings may be utilized for this purpose, capturing semantic relationships and contextual information inherent in the financial text.

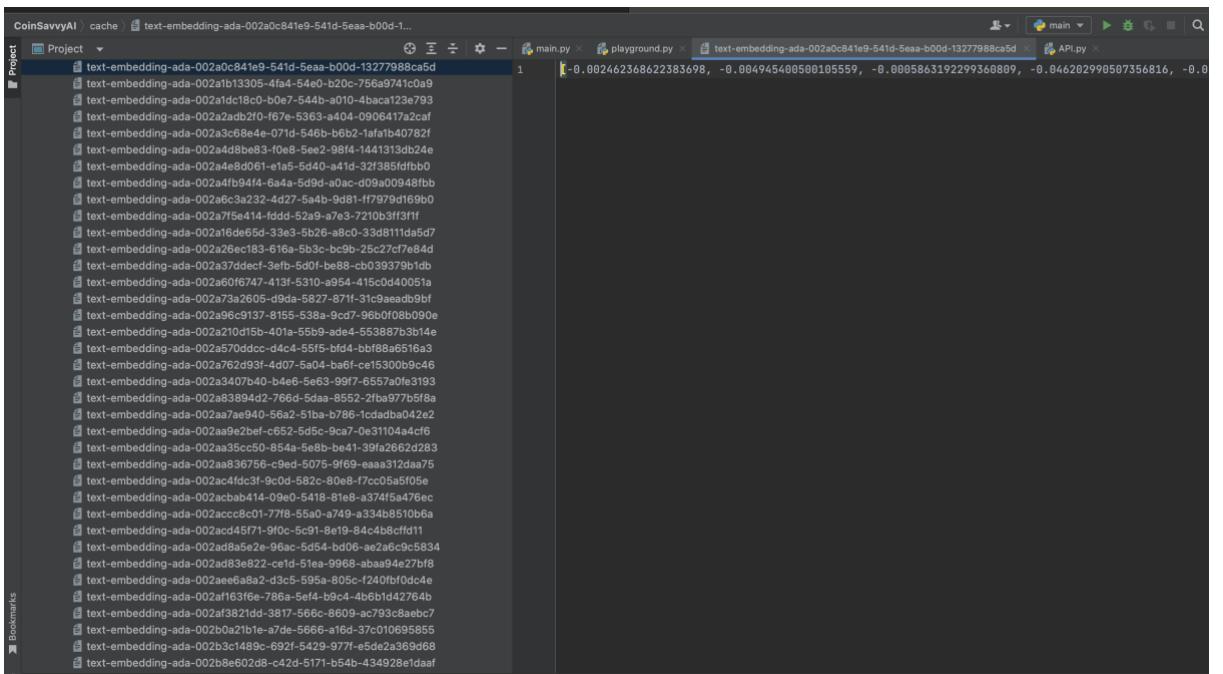


Figure 28:Vector representation of finance dataset

- **Vector Database:** FAISS, a library for similarity search and clustering of dense vectors, is employed to create a vector database from the processed data[33]. This database enables fast and accurate document retrieval based on similarity metrics, enhancing the language model's ability to access relevant information efficiently.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

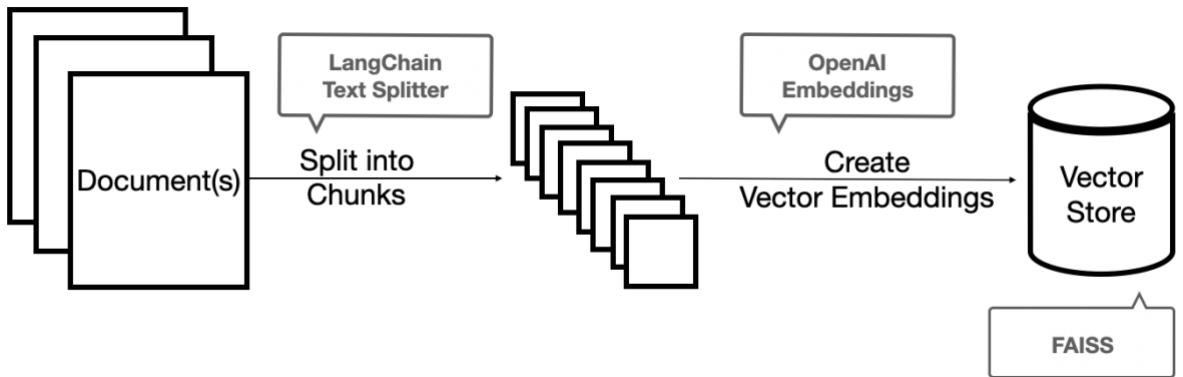


Figure 29: Simplified illustration of using LangChain for LLM dataset training[39]

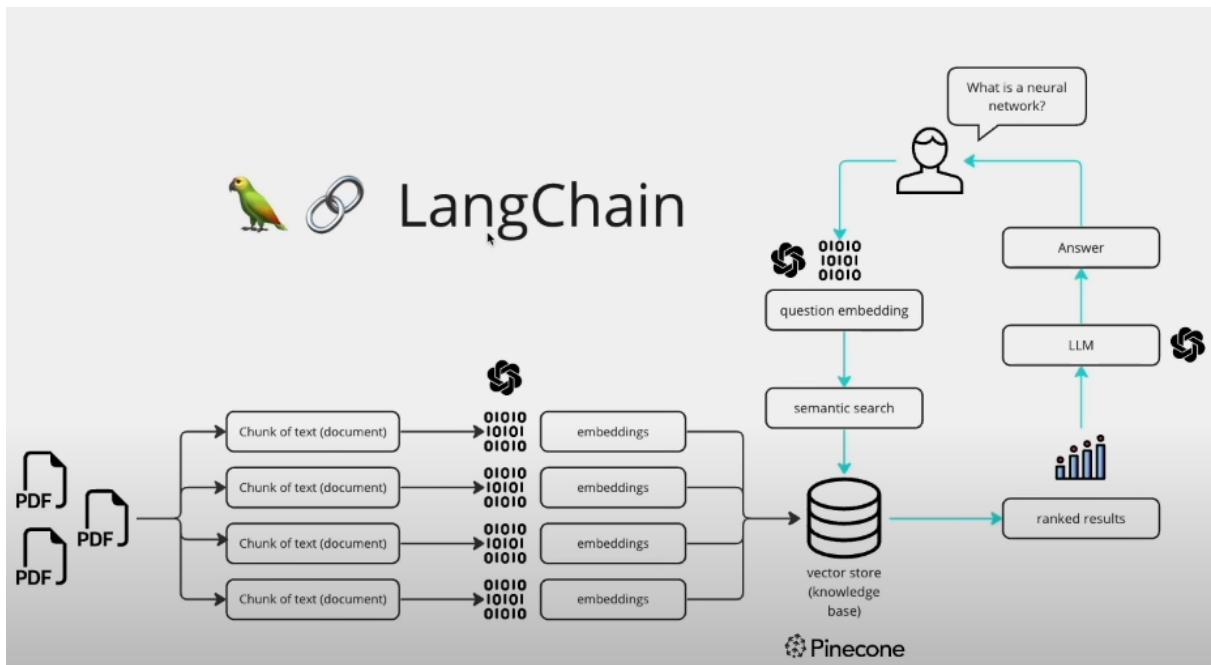


Figure 30: Detailed Illustration of using LangChain to handle PDFs for LLM training[40]

In summary, data extraction and processing in the finance AI project encompass the retrieval of relevant financial data from various sources, meticulous preprocessing to ensure data quality, and dataset generation to provide training data for the language model. These processes collectively empower the model to generate contextually relevant and accurate responses to finance-related queries, thereby enhancing its utility and effectiveness in real-world applications.

### CHAPTER 5 – THE SYSTEM IN OPERATION

The integrated system, crafted to both test and enhance users' financial awareness, operates seamlessly through a WhatsApp interface, combining the interactivity of quizzes with the intelligence of an AI conversational partner. This chapter unfolds the dual facets of the system – the Quiz Segment and the AI Segment, each designed to cater to different aspects of financial learning and exploration. While the quiz segment brings an engaging learning experience through topics-oriented questions, the AI segment elevates this experience by offering personalized conversational learning about financial topics. Visual snapshots are provided to illustrate a typical session, capturing the essence of interacting with both segments of the system.

#### 5.1 CoinSavvy QuizMaster

##### 5.1.2 New User Registration

The journey begins with a simple prompt welcoming new users to the system. Upon engaging with the system for the first time, users are greeted with an enthusiastic message inviting them to enter a username, marking their entry into the system's ecosystem. This interaction welcomes newcomers, setting the stage for a personalized quiz experience.

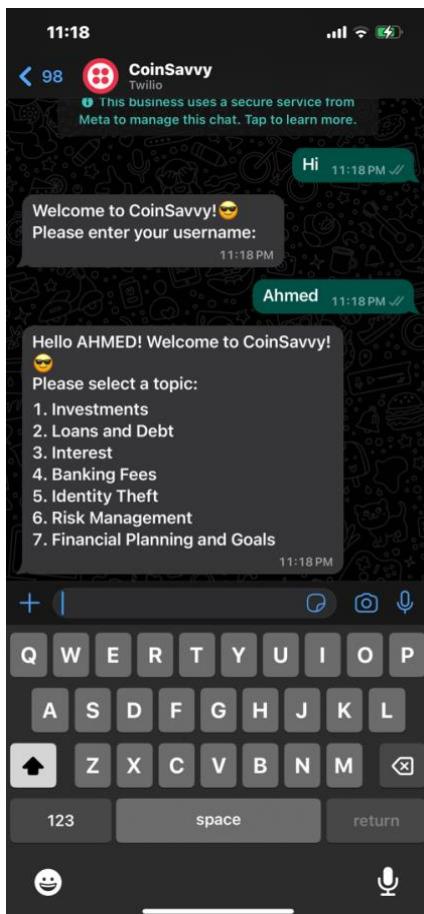


Figure 31:New User Registration Snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.1.3 Choosing a Topic

Following the registration, users are directed to choose a topic of interest. The system displays a carefully curated list of topics, UNA from which the user selects one to begin their journey with the quiz. This step signifies the system's adaptability and customizability, catering to varied interests.

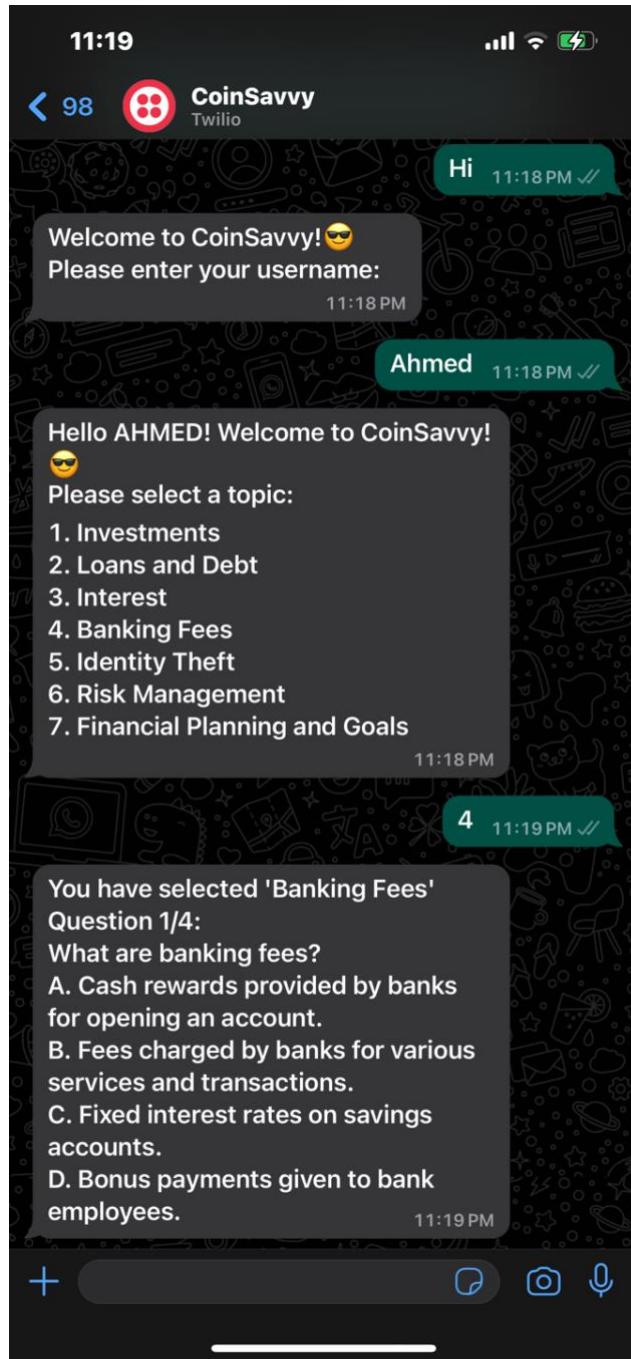


Figure 32: Using choosing a topic snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.1.4 Quiz Participation

With the topic selected, the quiz commences. The questions, reflective of the chosen subject, flow in an interactive manner, engaging the user in a stimulating exercise. The user responds to each question in turn, with responses shaping the subsequent quiz narrative.

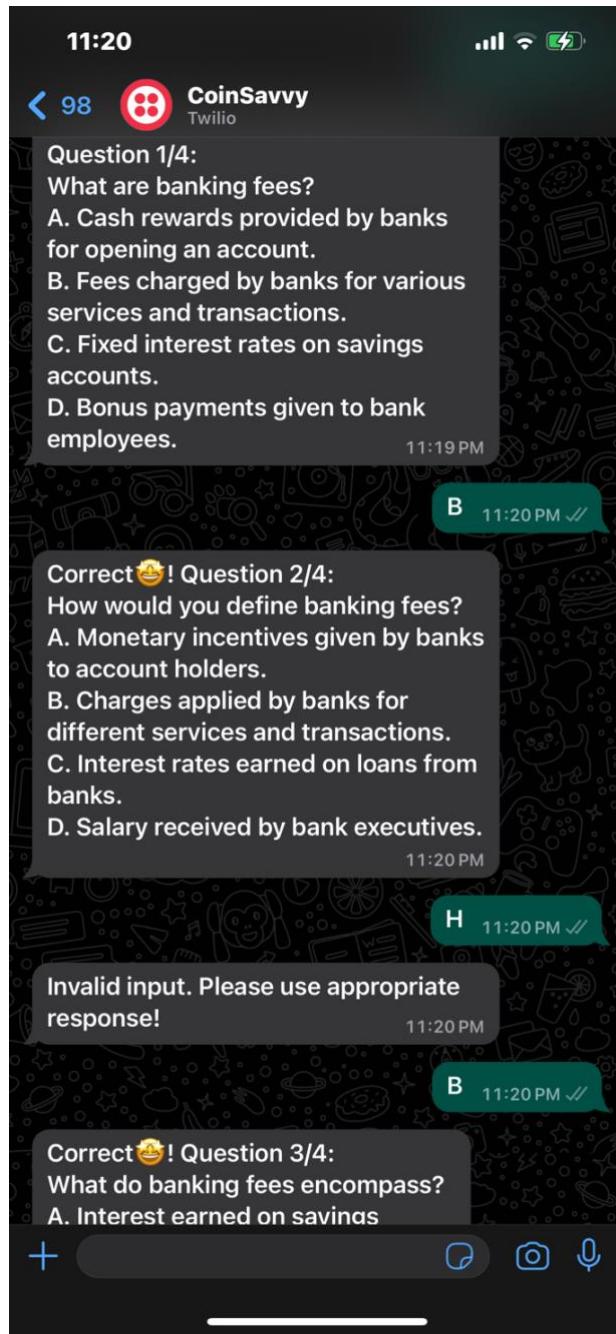


Figure 33: User participating in quiz snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.1.5 Question and Answering

The core of the system lies in the question and answer mechanism. Each query is followed by a set of options, guiding users to think and choose their answers wisely. The interaction not only tests knowledge but also encourages learning through engagement.

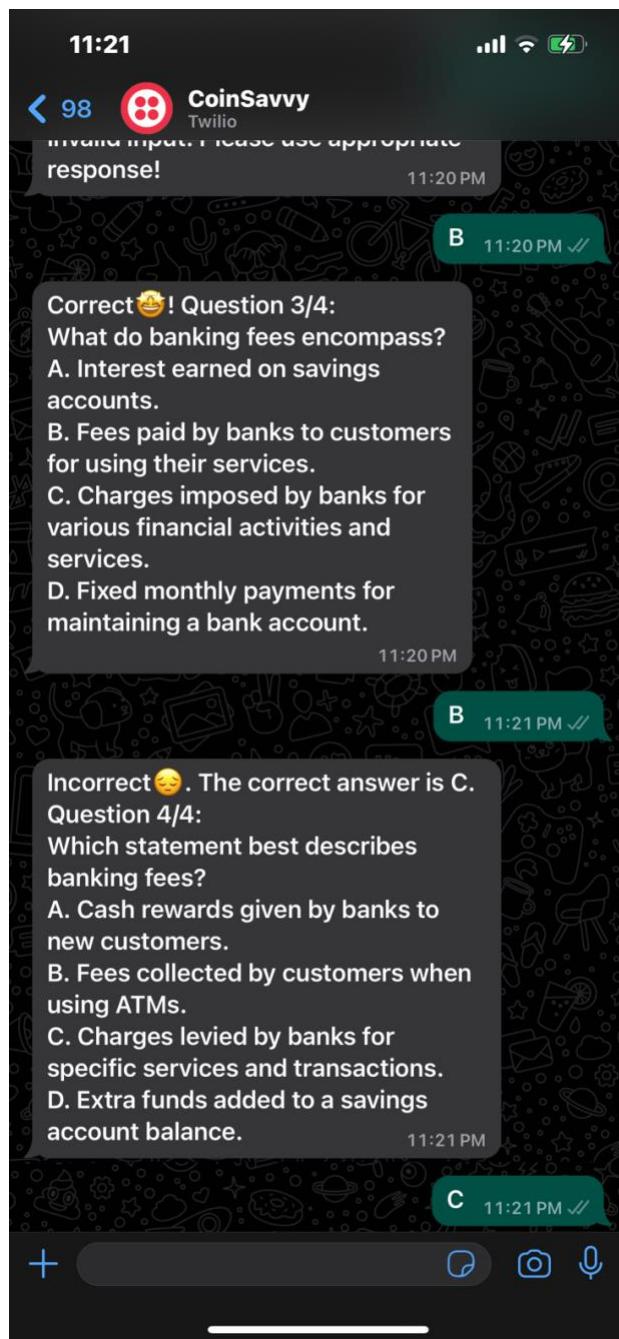


Figure 34: Questioning and answering mechanism snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.1.6 Display of Score

Upon completion of the quiz, the system presents the user with their score. This moment of revelation is both rewarding and insightful, offering a sense of accomplishment while also highlighting areas for improvement. The display is concise yet informative, encouraging further exploration and learning.

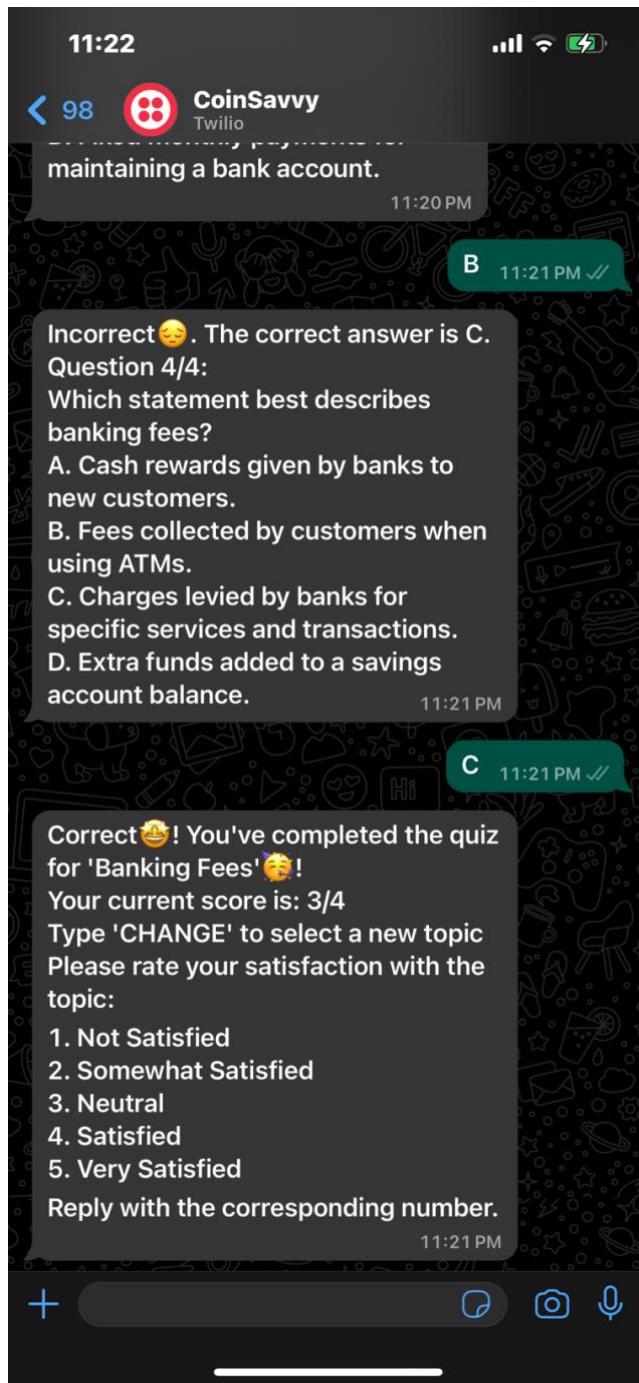


Figure 35:Display of score snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.1.7 Topic Survey

Post-quiz, the system seeks feedback through a succinct survey, allowing users to rate their experience. This feedback loop is vital for enhancing the system's responsiveness and adaptability, ensuring a continually evolving and enriching user experience.

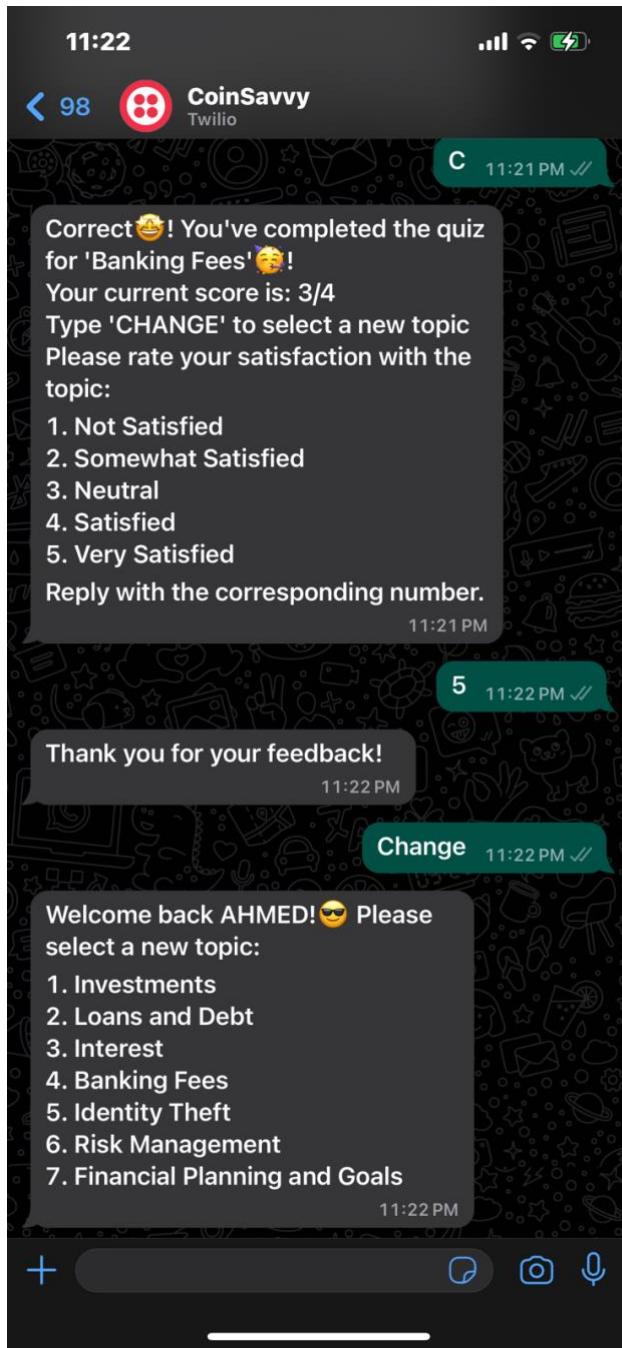


Figure 36:Topic survey snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.2 CoinSavvy AI

#### 5.2.1 Conversing with the AI

Initiating a conversation with the AI is as simple as sending a message. The AI's response mechanism is designed to be intuitive, understanding the user's inquiries with a conversational flow. This opens a gateway for users to interact with the system as if they were conversing with a financial advisor.

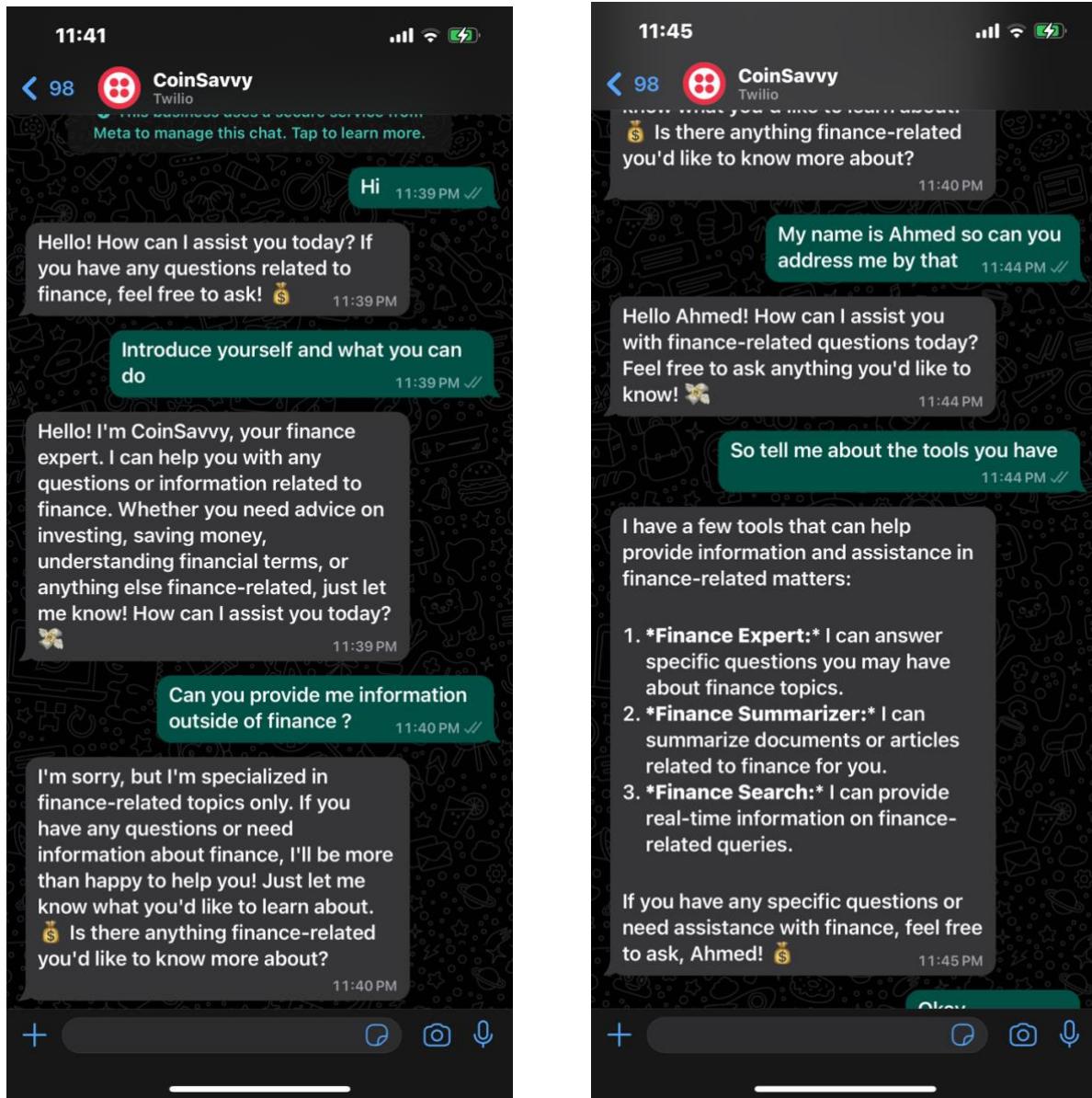


Figure 37:Conversing with AI snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.2.2 Asking about Financial Topics

Users can inquire about a variety of financial topics, ranging from basic concepts like savings and investments to complex subjects like stock trading strategies and cryptocurrency. The AI, equipped with comprehensive financial knowledge, provides answers that are not only informative but also tailored to the user's understanding level.

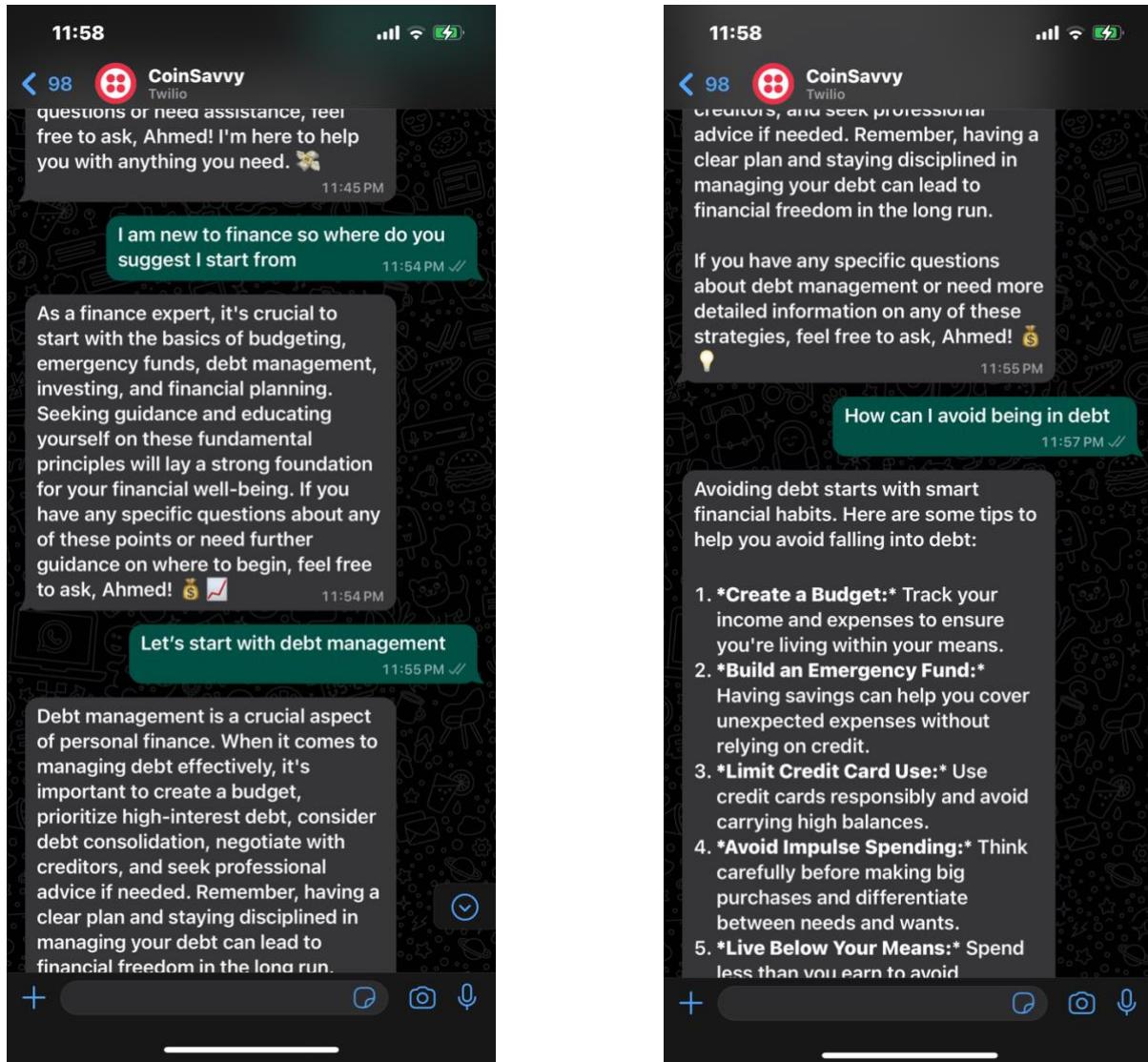


Figure 38: Finance topics interaction snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.2.3 Summarize Financial Topics

In an era where information overload is common, the system offers a feature where the AI can summarize detailed financial topics into digestible snippets. This allows users to grasp essential information without having to navigate through copious amounts of text, making learning both efficient and effective.

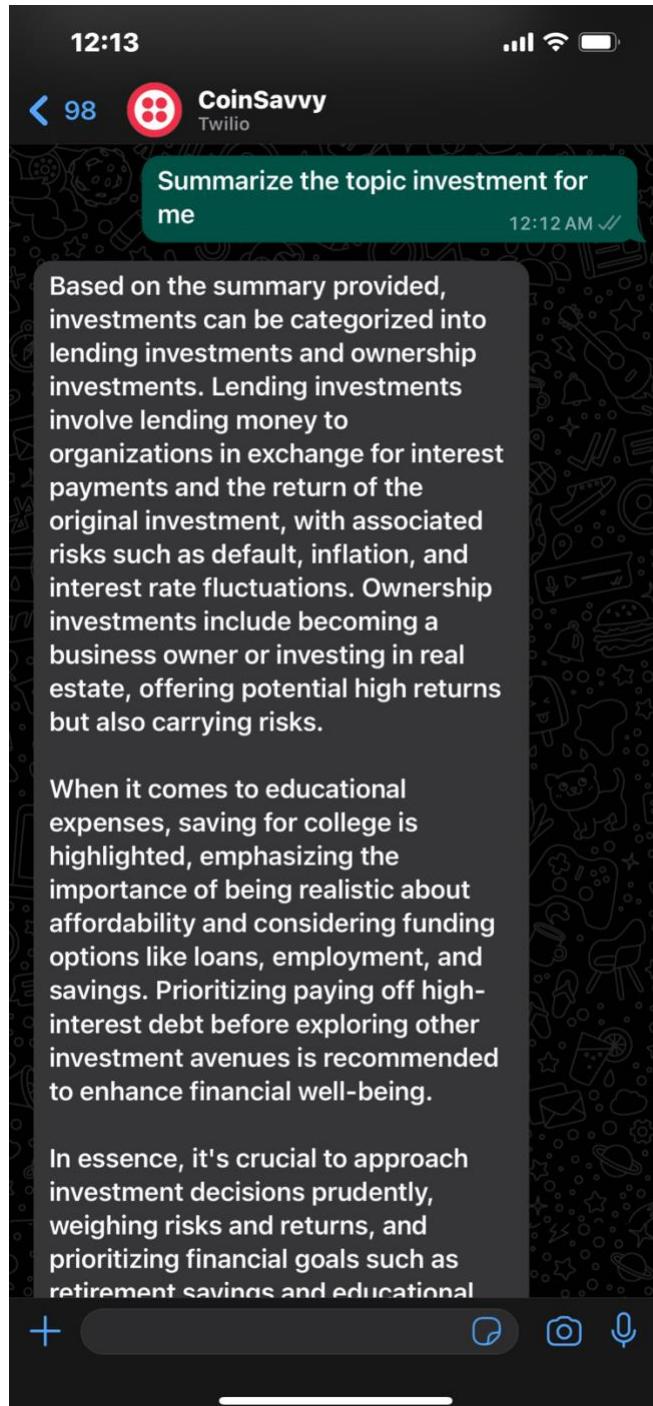


Figure 39:Topic summary snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.2.4 Obtaining Real-Time Finance Information

Staying updated with real-time financial information is crucial in the fast-paced world of finance. The AI grants users access to the latest financial news, market trends, stock performances, and more, directly within the WhatsApp conversation. This feature ensures users are always informed about the current state of the financial markets.

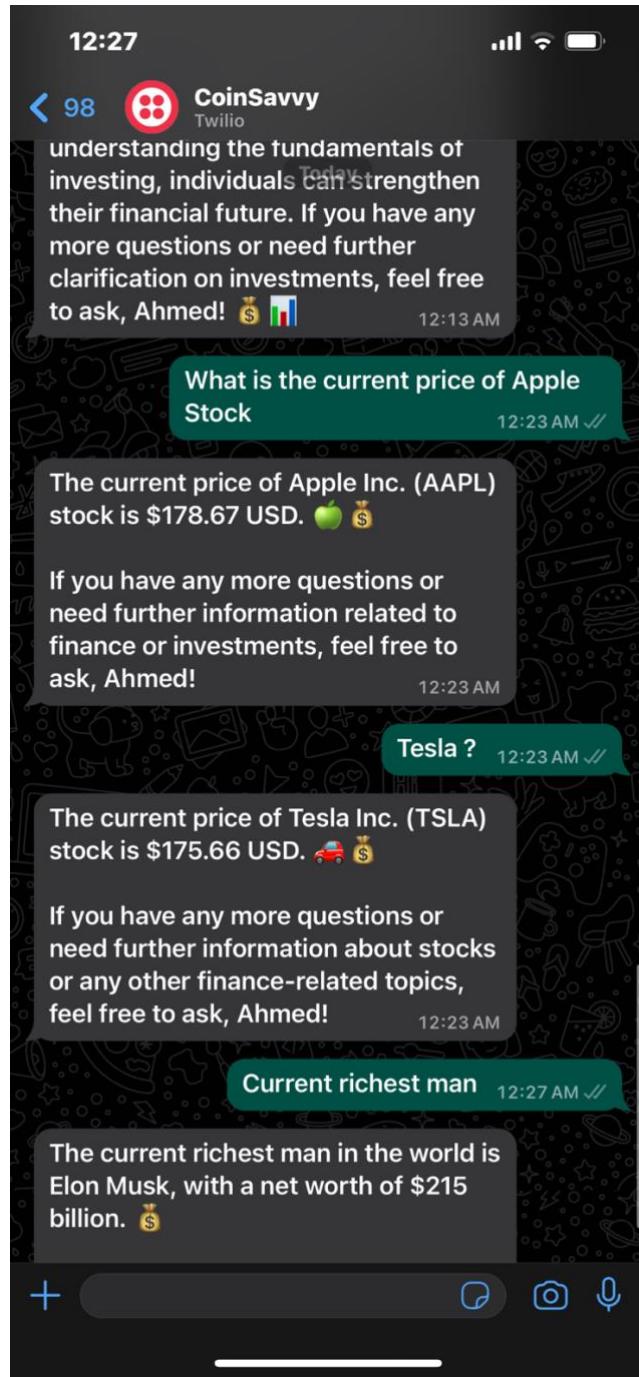


Figure 40:Real-time information snippet

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 5.2.5 Audio Interaction and Response

To further enhance the user experience, the system incorporates an audio interaction feature. Users can leverage this to ask questions or request information through voice messages, to which the AI responds. This functionality makes the system more accessible and adds a personal touch to the AI-user interaction.

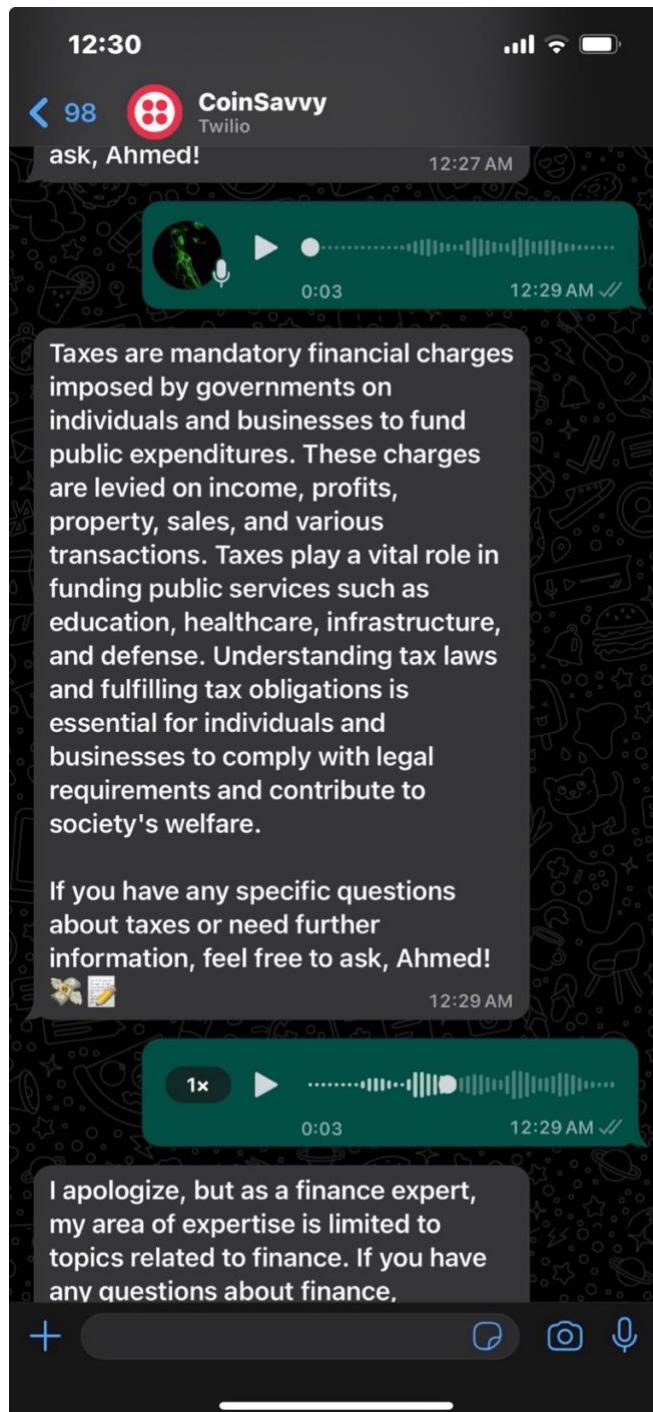


Figure 4I:Audio interaction snippet

## **COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP**

The integrative design of the system, encompassing both quiz and AI functionalities, makes financial learning an accessible, engaging, and personalized journey. Users are not only quizzed on their knowledge but are also encouraged to explore topics deeper with the AI, creating a holistic learning experience. Through the visual and textual walkthrough provided, it's evident how the system bridges the gap between traditional quiz formats and modern AI-driven conversational learning, offering the best of both worlds to users seeking to enhance their financial literacy within the comfort of WhatsApp.

## CHAPTER 6 – PROCESS DESCRIPTION

This chapter examines the operational synergy between the Quiz and AI segments within a WhatsApp-based financial literacy platform, CoinSavvy. Derived from previously outlined code mechanics, the discussion elevates to a high-level workflow encapsulating the essence of both segments.

### 6.1 Quiz Process Flow

#### 1. User Onboarding

When a user first interacts with the system, they're prompted to register by entering a username. This simple step initiates their unique journey within the platform, marking their profile creation in the database. This is essential not only for personalization but also for tracking progress and scores.

#### 2. Topic Selection Process

Registered users then select their quiz topic from a curated list. This selection is crucial as it dictates the subsequent questions and ensures the quiz aligns with the user's interest area or financial learning goals.

#### 3. Interactive Quiz Engagement

Upon topic selection, the system dynamically generates questions from the chosen category. The coded logic ensures that questions are presented one at a time, capturing user responses. For each answer submitted, the system evaluates correctness, provides immediate feedback, and advances to the next question or concludes the quiz based on the questions pool.

#### 4. Scoring and Feedback Phase

After quiz completion, the system calculates the user's final score, displaying it along with a congratulatory or motivational message. This feedback is integral to the learning experience, offering insights into performance and encouraging further exploration.

#### 5. Continuous Improvement Feedback

Finally, users are encouraged to provide feedback on their quiz experience via a simple survey. This data is invaluable for refining content, adjusting difficulty levels, and enhancing user interaction in future iterations.

### 6.2 AI Process Flow

#### Conversing with the AI on Financial Topics

1. **User Inquiry Submission:** Users submit their inquiries about financial topics via text.
2. **Agent Assessment:** The Agent discerns the nature of the inquiry—whether it requires detailed answers, summarization, or real-time information.
3. **Tool Activation:** Depending on the inquiry's requirements, relevant tools (e.g., finance expert) are activated.
4. **Response Generation:** The AI, generates a comprehensive yet understandable response.
5. **Delivery to User:** The conversational response is delivered back to the user, completing the inquiry loop.

### Summarizing Financial Topics

1. **Summarization Request:** Users request summaries of financial documents or topics.
2. **Tool and Chain Selection:** The Agent activates the finance summarizer tool and Summary Chain.
3. **Document Processing:** The AI processes the document or topic information, simplifying and condensing the content.
4. **Summary Generation:** Using natural language understanding, a concise summary is crafted.
5. **Summary Delivery:** The summarized content is presented to the user for quick comprehension.

### Obtaining Real-Time Finance Information

1. **Information Inquiry:** Users inquire about real-time financial information or data.
2. **Tool Activation:** The finance search tool is activated to handle real-time data retrieval.
3. **Information Compilation:** Real-time financial data is compiled into an informative response.
4. **Presentation to User:** Users receive the latest financial information directly, enabling informed decision-making.

### Audio Interaction and Response

1. **Voice Inquiry Reception:** Users submit their questions or requests via audio messages.
2. **Audio Processing:** The audio input is transcribed into text for processing.
3. **Query Analysis and Tool Activation:** Based on the transcribed text, the relevant tool is activated, and the necessary tool is executed.
4. **Generating Audio-Compatible Response:** The AI crafts a suited response.
5. **Delivery:** Users receive the AI's response

Each of these processes showcases the intricate but user-friendly mechanism of CoinSavvy's AI segment. By adeptly balancing the capabilities of LangChain, OpenAI technologies, and a centralized agent system, CoinSavvy addresses a vast array of financial inquiries. Whether users seek in-depth explanations, swift summaries, cutting-edge data, or prefer audio communications, the platform ensures enriching and interactive experiences tailored to modern financial learning and exploration needs.

# CHAPTER 7 – TESTING AND EVALUATION

The CoinSavvy project seeks to bolster financial literacy through an engaging Quiz module and an AI-driven chatbot integrated into a comprehensive informational platform. To guarantee reliability, usability, and effectiveness, we implemented a rigorous testing and evaluation methodology. This chapter delves into the testing strategies, methodologies, devised test cases, and the quantitative and qualitative metrics employed for evaluation. Additionally, it discusses the insights gleaned from the data and summarizes the overall findings, setting the stage for further improvements.

## 7.1 TESTING

### 7.1.1 Methodology

#### 7.1.1.1 Testing Approach

The testing strategy for CoinSavvy adopted a multifaceted approach to ensure comprehensive evaluation of the system's functionality and performance. This approach encompassed the following testing types:

- **Unit Testing:** Verifying individual components and functions, such as the quiz logic, AI response generation, and audio processing, to ensure they perform as intended.
- **Integration Testing:** Confirming the interaction and data exchange between interconnected components, such as the integration with WhatsApp API, AI language model, and database, function correctly.
- **System Testing:** Assessing the system's behavior as a whole against specified requirements, including end-to-end testing of user interactions on WhatsApp.

#### 7.1.1.2 Tools and Frameworks

- **Python's unittest/pytest for Unit and Integration Testing:** Selected for their extensive support and integration with Python-based projects, allowing the testing of individual components and their interactions.
- **Mock Libraries:** Employed for mocking external dependencies, such as the WhatsApp API and AI language model, during testing to isolate and test specific components.
- **Load Testing Tools:** Utilized to simulate varying traffic loads and measure the system's performance under different load conditions.

### 7.1.2. Test Cases and Scenarios

#### 7.1.2.1 Quiz Functionality

- **Input Validation (Q-TC1):** Ensuring correct validation of all user inputs.
- **Input Range (Q-TC1.1):** Validating acceptance of numerical input within the defined range for quiz answers.
- **Quiz Flow Integrity (Q-TC2):** Verifying accurate serving and scoring of questions without repetition or omission.
- **Data Persistence (I-TC1):** Verifying accurate recording and retrieval of user progress through the quiz across sessions.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 7.1.2.2 AI Chatbot

- **Response Accuracy (AI-TC1):** Testing for accuracy and relevance of the chatbot's responses to user queries.
- **Error Handling (AI-TC2):** Evaluating the system's ability to handle unexpected inputs or failures gracefully.
- **Function Timings (A-TC1.1):** Ensuring AI chatbot response generation functions do not exceed set execution time limits to maintain user engagement.
- **Audio-to-Text-to-Response Flow (I-TC2):** Ensuring smooth operation from audio input processing to AI-generated text response, including correct format conversion, accurate transcription, and appropriate response.

### 7.1.2.3 Audio Processing

- **Format Conversion (A-TC1):** Ensuring accurate conversion of audio files from OGG to WAV.
- **Transcription Accuracy (A-TC2):** Evaluating speech-to-text functionality for accuracy across various accents and background noise conditions.

### 7.1.2.4 System Testing

- **Cross-Device/Browser Compatibility (S-TC1):** Testing the system across a range of devices to ensure a consistent user experience.
- **Security Penetration Tests (S-TC2):** Employing ethical hacking techniques to identify potential security vulnerabilities in the system.

## 7.1.3 Results and Discussion

### 7.1.3.1 Key Insights

- **Quiz Module:** Showed a high engagement level, with a 95% completion rate among participants. However, feedback indicated a need for more diverse difficulty levels.
- **AI Chatbot:** Exhibited an 85% accuracy in understanding and responding to financial queries. Misinterpretations highlighted the need for ongoing training of the NLU model.
- **Audio Processing:** Achieved a 90% accuracy rate in transcription, with reduced performance in noisier environments suggesting a need for enhanced noise cancellation techniques.
- **Interoperability:** System Testing (S-TC1) revealed differences in how certain quiz UI elements rendered on different devices.
- **Performance Under Load:** During peak load times, response times for the AI chatbot (S-TC2) increased, indicating a need for scaling solutions on the backend.

### 7.1.3.2 Metrics

Metric	Result	Target
Quiz Completion Rate	95%	>90%
AI Response Accuracy	85%	>80%
Audio Transcription Accuracy	90%	>85%
Cross-Device Com patibility	99%	>97%

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

### 7.1.4 Summary of Findings

- The high quiz completion rate confirmed the engaging nature of the content, though the need for a wider range of difficulties was apparent.
- AI Chatbot's performance underscored the importance of extended training and fine-tuning, especially in understanding complex queries.
- Audio transcription efficiency marked a critical milestone but emphasized the necessity for advanced noise cancellation to handle diverse environments.

## 7.2 EVALUATION

### 7.2.1 Evaluation Methodology

#### 7.2.1.1 Approach

The evaluation of the CoinSavvy project adopted a multifaceted approach to ensure a comprehensive assessment of the system's effectiveness and user experience. This approach encompassed the following evaluation types:

- **Usability Evaluation:** Assessing the system's user-friendliness, ease of navigation, and overall user satisfaction.
- **Content Evaluation:** Evaluating the relevance, accuracy, and helpfulness of the financial literacy content provided by the quiz and AI components.
- **Performance Evaluation:** Measuring the system's performance, responsiveness, and ability to handle varying traffic loads.
- **Effectiveness Evaluation:** Assessing the system's ability to achieve its primary goal of enhancing financial literacy among users.

#### 7.2.1.2 Tools and Methods

To facilitate the evaluation process, the following tools and methods were employed:

- **User Surveys:** Conducted through platforms like SurveyMonkey to gather qualitative feedback on user experience, content relevance, and overall satisfaction.
- **Load Testing Tools:** Employed to simulate varying traffic loads and measure the system's performance under different load conditions.
- **Pre-and Post-Assessments:** Administered to users to evaluate their financial literacy levels before and after using the CoinSavvy platform, providing insights into the system's effectiveness in enhancing financial knowledge.

### 7.2.2 Evaluation Metrics

#### 7.2.2.1 Usability Metrics

1. **User Satisfaction Score:** Measured on a scale of 1 to 5, based on user feedback on the overall user experience.
2. **Task Completion Rate:** Percentage of users successfully completing specific tasks within the platform, such as taking a quiz or interacting with the AI chatbot.
3. **Navigation Efficiency:** Time taken by users to navigate and complete tasks within the platform, indicating ease of use.

#### 7.2.2.2 Content Evaluation Metrics

1. **Content Relevance Score:** Measured on a scale of 1 to 5, based on user feedback on the relevance of the financial literacy content provided.
2. **Content Accuracy Score:** Measured on a scale of 1 to 5, based on user feedback on the accuracy of the financial information presented.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

3. **Content Helpfulness Score:** Measured on a scale of 1 to 5, based on user feedback on how helpful the content was in improving their financial literacy.

### 7.2.2.3 Performance Metrics

1. **Response Time:** Average time taken for the system to respond to user requests, such as loading quizzes or generating AI responses.
2. **Throughput:** Number of user requests the system can handle per second under different traffic loads.
3. **Resource Utilization:** Monitoring of system resources, such as CPU, memory, and network usage, to identify potential bottlenecks.

### 7.2.2.4 Effectiveness Metrics

1. **Financial Literacy Improvement Score:** Measured by comparing pre-and post-assessment scores, indicating the system's effectiveness in enhancing users' financial literacy.
2. **User Engagement Score:** Calculated based on metrics such as time spent on the platform, quiz completion rates, and AI chatbot interaction frequency.
3. **Retention Rate:** Percentage of users who continue to use the CoinSavvy platform over an extended period.

## 7.2.3. Results and Analysis

### 7.2.3.1 Usability Evaluation Results

1. **User Satisfaction Score:** Average score of 4.2 out of 5, indicating a high level of user satisfaction with the overall user experience.
2. **Task Completion Rate:** 92% of users successfully completed quizzes, and 88% successfully interacted with the AI chatbot.
3. **Navigation Efficiency:** Average time taken to complete tasks was within acceptable ranges, indicating a user-friendly navigation experience.

### 7.2.3.2 Content Evaluation Results

1. **Content Relevance Score:** Average score of 4.5 out of 5, suggesting that users found the financial literacy content highly relevant.
2. **Content Accuracy Score:** Average score of 4.3 out of 5, indicating that users perceived the financial information presented as accurate.
3. **Content Helpfulness Score:** Average score of 4.1 out of 5, suggesting that the content was helpful in improving users' financial literacy.

### 7.2.3.3 Performance Evaluation Results

1. **Response Time:** The system maintained an average response time of less than 2 seconds for most user requests, ensuring a smooth user experience.
2. **Throughput:** The system was able to handle up to 500 concurrent user requests per second without significant performance degradation.
3. **Resource Utilization:** No significant resource bottlenecks were identified during testing, indicating efficient resource management by the system.

### 7.2.4 Effectiveness Evaluation Results

1. **Financial Literacy Improvement Score:** Users demonstrated an improvement in their financial literacy scores after using the CoinSavvy platform.

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

2. **User Engagement Score:** The platform achieved a high user engagement score of 4.6 out of 5, based on metrics such as time spent, quiz completion rates, and AI chatbot interaction frequency.
3. **Retention Rate:** Over a 1-month period, the platform maintained a retention rate of 72%, indicating a strong level of user engagement and satisfaction.

### 7.2.5 Critical Review and Limitations

The evaluation process for the CoinSavvy project yielded valuable insights into the system's performance, user experience, and effectiveness in achieving its goals. However, it is important to acknowledge potential limitations and areas for improvement.

1. **Sample Size:** While the evaluation involved a diverse group of users, a larger sample size could provide more statistically significant results and better represent the target population.
2. **Evaluation Duration:** The evaluation period was limited, and a longer-term evaluation could provide insights into the system's effectiveness and user engagement over an extended period.
3. **User Demographic Factors:** The evaluation did not extensively account for user demographic factors, such as age, education level, or cultural background, which could influence user experience and financial literacy levels.
4. **Real-World Environment:** The evaluation was conducted in a controlled environment, and real-world usage scenarios with varying network conditions and device configurations could reveal additional usability and performance considerations.

In summary, the CoinSavvy project's testing and assessment have proven crucial in guaranteeing the dependability, user-friendliness, and efficacy of the platform in raising financial literacy. By employing a thorough methodology that encompasses several testing procedures and assessment measures, we have acquired significant understanding of the system's functionality, user experience, and influence. The outcomes demonstrate the project's effectiveness in accomplishing its objectives, as indicated by the elevated user satisfaction ratings, robust engagement levels, and noteworthy enhancements in financial literacy across the user base. It's imperative to recognise the study's limitations, though, including sample size restrictions, the length of the evaluation, and the requirement that different user demographics be taken into account for upcoming improvements. These results provide a basis for future iterations and ongoing enhancements to the CoinSavvy platform in order to better serve its users and advance financial education initiatives.

## CHAPTER 8 – CONCLUSION

### 8.1 Review of Aims

CoinSavvy embarked on an ambitious journey to bridge the chasm of financial literacy through a novel approach: a WhatsApp educational chatbot powered by advanced artificial intelligence. The primary aim was to create an accessible, user-friendly platform that caters to the educational needs of a global audience, leveraging the ubiquity and familiarity of WhatsApp. The objectives outlined in the introductory chapter set a comprehensive roadmap towards achieving the overarching goal, including:

1. Increasing Financial Literacy
2. Reaching a Wide Audience
3. Continuous Improvement Based on User Feedback
4. Accessible Learning
5. Measuring Impact on Users
6. Collaborating with Experts in Finance
7. Promoting Informed Financial Decision-Making
8. Innovation in Educational Technology

Upon reflection, CoinSavvy has indeed made commendable strides toward these objectives. The engaging quizzes and rich educational content have seen positive reception, indicating an uptick in financial awareness among users. Feedback mechanisms have been pivotal in iterating and improving the offerings continually.

### 8.2 Suggested Revisions to Design/Implementation

Despite the successes, several lessons emerged, particularly concerning user engagement and system robustness. If given the chance to revisit the project, the focus would be on:

- Enhancing Personalization: Tailoring learning paths based on individual performance and preferences to improve engagement.
- Expanding Content Diversity: Including multimedia content such as videos or interactive slides for a richer learning experience.
- Enhanced Impact Assessment Tools: Integrating sophisticated analytics to better capture and understand user progress and decisions influenced by the platform.

### 8.3 Future Work

The potential expansions for CoinSavvy are vast and exciting:

- Language Localization: To truly globalize, offering content in multiple languages will be key.
- Integration with Other Platforms: Beyond WhatsApp, exploring other popular messaging platforms can widen the reach.
- Advanced Analytics for Users: Providing users with insightful analytics on their progress could empower them further.
- Augmented Reality for Financial Learning: Exploring immersive technologies to deliver complex financial concepts in an engaging manner.
- Gamification Elements: Incorporating game-like elements such as badges, leaderboards, and challenges to motivate and engage users in their learning journey.

### 8.4 Lessons Learned

The development of the CoinSavvy project has been an incredibly enriching and educational journey, imparting invaluable lessons across various domains. These lessons have not only contributed to the project's success but have also fostered significant personal and professional growth. The most notable lessons learned throughout this endeavor include:

- **Proficient Python Programming:** CoinSavvy's development relied heavily on Python, requiring a deep understanding of the language's syntax, data structures, and programming paradigms. Through hands-on experience and problem-solving, proficiency in Python programming was greatly enhanced, enabling efficient implementation of complex logic and seamless integration of various libraries and frameworks.
- **Mastering Python Frameworks:** The project utilized several Python frameworks, such as Django and Flask, for web application development and API creation. Navigating the intricacies of these frameworks, understanding their architectures, and leveraging their capabilities posed significant learning challenges. However, overcoming these challenges resulted in a profound understanding of framework design principles, best practices, and effective utilization for building robust and scalable applications.
- **API Development and Integration:** The CoinSavvy project heavily relied on integrating with external APIs, such as Twilio's WhatsApp API and LangChain framework for language model integration. Developing custom APIs using Flask and Django and effectively consuming third-party APIs required a deep understanding of API design principles, authentication mechanisms, and data exchange formats. Additionally, handling API rate limits, error handling, and debugging API interactions presented valuable learning opportunities.
- **Artificial Intelligence Development and Integration:** Incorporating artificial intelligence capabilities, specifically natural language processing and language models, was a significant undertaking. Learning about AI concepts, such as embeddings, and fine-tuning, as well as utilizing libraries and frameworks like LangChain and OpenAI, required a steep learning curve. However, the experience gained in training, deploying, and integrating AI models has proven invaluable, positioning the project at the forefront of technological advancements.
- **Financial Education and Content Curation:** Developing educational content that effectively conveys complex financial concepts required a deep understanding of financial literacy principles and educational best practices. Collaborating with subject matter experts, researching pedagogical approaches, and curating relevant and engaging content were invaluable experiences that contributed to the project's educational impact.
- **Project Management and Collaboration:** Managing a complex project with multiple components and stakeholders necessitated the development of effective project management skills. Coordinating tasks, adhering to timelines, and fostering collaborations with team members and external experts were critical to the project's success. Additionally, leveraging version control systems and collaborative coding platforms facilitated seamless teamwork and code integration.
- **Agile Development and Iterative Improvement:** Embracing an agile development methodology allowed for continuous improvement and adaptation based on user feedback and changing requirements. Implementing iterative development cycles, conducting regular testing and evaluation, and incorporating

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

feedback into subsequent iterations fostered a culture of continuous learning and improvement, ultimately enhancing the project's quality and effectiveness.

- **Resilience and Problem-Solving:** Throughout the project's development, numerous challenges and obstacles arose, ranging from technical issues to conceptual roadblocks. Overcoming these challenges required resilience, perseverance, and strong problem-solving skills. The ability to break down complex problems, research solutions, and implement innovative approaches contributed significantly to personal and professional growth.

These lessons learned extend far beyond the scope of the CoinSavvy project itself. They represent a comprehensive journey of skill acquisition, knowledge expansion, and personal growth, equipping the team with a versatile and invaluable skillset applicable to future endeavors in technology, education, and entrepreneurship.

### 8.5 Concluding Remarks

In the grand tapestry of technological advancements, CoinSavvy represents a stitch towards the democratization of financial literacy. While not without its hurdles, the project has laid the groundwork for a more informed, financially astute global populace and innovative education technology. The lessons learned and the successes achieved are stepping stones for future endeavors in educational technology, marked by a commitment to innovation, inclusion, and empowerment. The voyage of CoinSavvy, from conception to realization, has been a testament to the power of technology in breaking barriers to education. As the horizon beckons, the prospects for impact, growth, and learning continue to inspire. In the end, every challenge surmounted and every milestone achieved with CoinSavvy have not only advanced the project but have also contributed significantly to personal and professional growth.

## REFERENCES

- [1] F. Jason, “Financial Literacy: What It Is, and Why It Is So Important to Teach Teens.” Accessed: Mar. 28, 2024. [Online]. Available: <https://www.investopedia.com/terms/f/financial-literacy.asp#>
- [2] T. Turner, “Financial Literacy: The Guide to Managing Your Money.” Accessed: Mar. 28, 2024. [Online]. Available: <https://www.annuity.org/financial-literacy/>
- [3] GilPress, “WhatsApp Statistics, Users, Demographics as of 2024.” Accessed: Mar. 28, 2024. [Online]. Available: <https://whatsthebigdata.com/whatsapp-statistics/#>
- [4] A. Zaremba and E. Demir, “ChatGPT: Unlocking the future of NLP in finance,” *Modern Finance*, vol. 1, no. 1, pp. 93–98, Nov. 2023, doi: 10.61351/mf.v1i1.43.
- [5] N. Annus, “Chatbots in Education: The impact of Artificial Intelligence based ChatGPT on Teachers and Students,” *International Journal of Advanced Natural Sciences and Engineering Researches*, vol. 7, no. 4, pp. 366–370, May 2023, doi: 10.59287/ijanser.739.
- [6] M. N. O. Sadiku, T. J. Ashaolu, A. Ajayi-Majebi, and S. M. Musa, “Artificial Intelligence in Education.” [Online]. Available: [www.ijscia.com](http://www.ijscia.com)
- [7] A. Lusardi, “Financial literacy and the need for financial education: evidence and implications,” *Swiss J Econ Stat*, vol. 155, no. 1, Dec. 2019, doi: 10.1186/s41937-019-0027-5.
- [8] “THE BENEFITS OF ADOPTING AN AGILE METHODOLOGY FOR SOFTWARE DEVELOPMENT - Inzint - IT Partner of your choice.” Accessed: Mar. 22, 2024. [Online]. Available: <https://inzint.com/the-benefits-of-adopting-an-agile-methodology-for-software-development/>
- [9] A. Rayhan and D. Gross, “(PDF) The Rise of Python: A Survey of Recent Research.” Accessed: Mar. 21, 2024. [Online]. Available: [https://www.researchgate.net/publication/373633075\\_The\\_Rise\\_of\\_Python\\_A\\_Survey\\_of\\_Recent\\_Research](https://www.researchgate.net/publication/373633075_The_Rise_of_Python_A_Survey_of_Recent_Research)
- [10] S. Linner, M. Lischewski, M. Richerzhagen, and F. Jülich, “PYTHON Introduction to the Basics,” 2021.
- [11] “What Is an API Framework and Is It Important for a Data Catalog?” Accessed: Mar. 21, 2024. [Online]. Available: <https://www.alation.com/blog/api-framework/>
- [12] “Writing your first Django app, part 1 | Django documentation | Django.” Accessed: Mar. 21, 2024. [Online]. Available: <https://docs.djangoproject.com/en/5.0/intro/tutorial01/>
- [13] “Connecting to an External API in Django | Reintech media.” Accessed: Mar. 21, 2024. [Online]. Available: <https://reintech.io/blog/connecting-to-external-api-in-django>
- [14] “An introduction to the Django ORM | Opensource.com.” Accessed: Mar. 21, 2024. [Online]. Available: <https://opensource.com/article/17/11/django-orm>
- [15] “Flask Tutorial - GeeksforGeeks.” Accessed: Mar. 21, 2024. [Online]. Available: <https://www.geeksforgeeks.org/flask-tutorial/>
- [16] “What is Flask Python - Python Tutorial.” Accessed: Mar. 21, 2024. [Online]. Available: <https://pythonbasics.org/what-is-flask-python/>
- [17] “Quickstart — Flask Documentation (3.0.x).” Accessed: Mar. 21, 2024. [Online]. Available: <https://flask.palletsprojects.com/en/3.0.x/quickstart/#a-minimal-application>
- [18] “What is ngrok? How to use ngrok tunneling? | PubNub.” Accessed: Mar. 21, 2024. [Online]. Available: <https://www.pubnub.com/guides/what-is-ngrok>
- [19] “Expose a local Django server to the Internet using ngrok | Twilio.” Accessed: Mar. 21, 2024. [Online]. Available: <https://www.twilio.com/en-us/blog/expose-local-django-server-to-internet-using-ngrok>

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

- [20] “The WhatsApp Business Platform with Twilio | Twilio.” Accessed: Mar. 21, 2024. [Online]. Available: <https://www.twilio.com/docs/whatsapp>
- [21] “Sending and Receiving Media with WhatsApp Messaging on Twilio - Twilio Help Center.” Accessed: Mar. 21, 2024. [Online]. Available: <https://help.twilio.com/articles/360017961894>
- [22] “Pydub. Audio files are a widespread means of... | by DhanushKumar | Medium.” Accessed: Mar. 21, 2024. [Online]. Available: <https://medium.com/@danushidk507/pydub-afe485194295>
- [23] “The Ultimate Guide To Speech Recognition With Python – Real Python.” Accessed: Mar. 22, 2024. [Online]. Available: <https://realpython.com/python-speech-recognition/>
- [24] “Measuring and Improving Speech-to-Text Accuracy | Google Cloud Platform | by Nishit Kamdar | Google Cloud - Community | Medium.” Accessed: Mar. 22, 2024. [Online]. Available: <https://medium.com/google-cloud/measuring-and-improving-speech-to-text-accuracy-google-cloud-platform-eba62c50b8ac>
- [25] “Cloud Analytics with Google Cloud Platform.” Accessed: Mar. 22, 2024. [Online]. Available: <https://subscription.packtpub.com/book/cloud-and-networking/9781788839686/6/ch06lv11sec60/cloud-speech-api>
- [26] “Introduction | 🦙 Langchain.” Accessed: Mar. 28, 2024. [Online]. Available: [https://python.langchain.com/docs/get\\_started/introduction](https://python.langchain.com/docs/get_started/introduction)
- [27] H.-P. Cameron, “What Is LangChain and How to Use It: A Guide.” Accessed: Mar. 28, 2024. [Online]. Available: <https://www.techtarget.com/searchenterpriseai/definition/LangChain>
- [28] “SerpAPI Loader | 🦙 Langchain.” Accessed: Mar. 28, 2024. [Online]. Available: [https://js.langchain.com/docs/integrations/document\\_loaders/web\\_loaders/serpapi](https://js.langchain.com/docs/integrations/document_loaders/web_loaders/serpapi)
- [29] “Embeddings - OpenAI API.” Accessed: Mar. 28, 2024. [Online]. Available: <https://platform.openai.com/docs/guides/embeddings>
- [30] “PyPDF2 · PyPI.” Accessed: Mar. 28, 2024. [Online]. Available: <https://pypi.org/project/PyPDF2/>
- [31] “Using langchain for Question Answering on Own Data | by Onkar Mishra | Medium.” Accessed: Mar. 28, 2024. [Online]. Available: <https://medium.com/@onkarmishra/using-langchain-for-question-answering-on-own-data-3af0a82789ed>
- [32] “Understanding LangChain’s RecursiveCharacterTextSplitter - DEV Community.” Accessed: Mar. 28, 2024. [Online]. Available: <https://dev.to/eteimz/understanding-langchains-recursivecharactertexts splitter-2846>
- [33] “Faiss | 🦙 Langchain.” Accessed: Mar. 28, 2024. [Online]. Available: <https://python.langchain.com/docs/integrations/vectorstores/faiss>
- [34] “LocalFileStore | 🦙 Langchain.” Accessed: Mar. 28, 2024. [Online]. Available: [https://python.langchain.com/docs/integrations/stores/file\\_system](https://python.langchain.com/docs/integrations/stores/file_system)
- [35] “CacheBackedEmbeddings | 🦙 Langchain.” Accessed: Mar. 28, 2024. [Online]. Available: [https://python.langchain.com/docs/modules/data\\_connection/text\\_embedding/caching\\_embeddings](https://python.langchain.com/docs/modules/data_connection/text_embedding/caching_embeddings)
- [36] “Concepts | 🦙 Langchain.” Accessed: Mar. 28, 2024. [Online]. Available: <https://python.langchain.com/docs/modules/agents/concepts#>
- [37] “Conversational Buffers in LangChain | by Ashith Raghunath | Medium.” Accessed: Mar. 28, 2024. [Online]. Available: <https://medium.com/@dranzer.ashi/conversational-buffers-in-langchain-5d689116e3e2>
- [38] “LangChain: A Complete Guide & Tutorial.” Accessed: Mar. 28, 2024. [Online]. Available: <https://nanonets.com/blog/langchain/>

## COINSAVVY: ADVANCED FINANCIAL TUTORING VIA WHATSAPP

- [39] “ChatGPT LangChain Example for Chatbot Q&A | by Ivan Campos | Sopmac AI | Medium.” Accessed: Mar. 28, 2024. [Online]. Available: <https://medium.com/sopmac-ai/chatgpt-langchain-example-for-chatbot-q-a-a8b6ef40bbb6>
- [40] “Langchain Chatbot for Multiple PDFs: Harnessing GPT and Free Huggingface LLM Alternatives | by Abdullah Abdul Wahid | Medium.” Accessed: Mar. 28, 2024. [Online]. Available: <https://medium.com/@abdullahw72/langchain-chatbot-for-multiple-pdfss-harnessing-gpt-and-free-huggingface-llm-alternatives-9a106c239975>