**ChatWithQuran**



**Haider Ali**

**29223**

**Faizan Qureshi**

**28582**

**Imdad Ullah**

**28556**

**Supervised by:**

**Muhammad Usman Karim**

**Faculty of Computing**

**Riphah International University, Islamabad**

**Fall 2024**

**A Dissertation Submitted To**

**Faculty of Computing,**

**Riphah International University, Islamabad**

**As a Partial Fulfillment of the Requirement for the Award of the Degree of**

**Bachelors of Science in Computer Science**

**Faculty of Computing**

**Riphah International University, Islamabad**

Date: [26-12-2024]

**Final Approval**

This is to certify that we have read the report submitted by ***Haider Ali (29229), Faizan Qureshi (28582) and Imdad Ullah (28556)*** for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Computer Science (BSCS). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Computer Science (BSCS).

**Committee:**

| **1** | Muhammad Usman Karim  (Supervisor) |
| --- | --- |
|  |  |
| **2** | Dr. Musharraf  (Head of Department) |

**Declaration**

We hereby declare that this document **“ChatWithQuran”** neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanying report entirely on the basis of our personal efforts, under the proficient guidance of our teachers, especially our supervisor **Muhammad Usman Karim.** If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Haider Ali**

**29223**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Faizan Qureshi**

**28582**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Imdad Ullah**

**28556**

**Dedication**

This project is dedicated to the group family, friends, especially our supervisor **“Muhammad Usman Karim”** who has been our mentor and guidance throughout this final year project, and whose support was invaluable in completing this final year project.

**Acknowledgement**

In this section, we would like to express our sincere gratitude to all the individuals who supported us throughout this project. Firstly, we are thankful to our supervisor **“Muhammad Usman Karim”**, who provided invaluable guidance and support throughout the entire process. In particular, we dedicate this project to him for his invaluable mentorship and Secondly, we extend our gratitude to all our teachers who have imparted knowledge and wisdom, enriching our understanding of the subject matter. Thirdly, we would like to acknowledge our fellow colleagues who have been a source of encouragement and assistance along the way.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Haider Ali**

**29223**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Faizan Qureshi**

**28582**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Imdad Ullah**

**28556**

**Abstract**

A web-based platform called ChatWithQuran was created to give users an all-encompassing, easily navigable, and engaging experience as they explore Quranic literature. Modern digital platforms and AI-powered Islamic chatbots provide simple search functions and succinct answers, but they frequently fall short in providing thorough interpretations and comprehensive answers to intricate questions. By using cutting-edge Natural Language Processing (NLP) techniques, this research seeks to close these gaps and provide accurate, contextually relevant, and thorough responses. Through the use of keyword-based features, ChatWithQuran will allow users to access particular verses from the Quran, translations, and comprehensive tafseer (interpretations) in response to their queries. By using this strategy, ChatWithQuran hopes to make the experience more interesting and instructive while enabling users to comprehend the lessons of the Quran more thoroughly.

By solving certain issues with current Quranic search and AI-based Islamic chatbot solutions, the ChatWithQuran platform distinguishes itself. Many of the platforms in use today either don't understand sophisticated, context-rich user enquiries, offer insufficient or partial tafseer, or don't provide thorough answers. ChatWithQuran's integration of advanced natural language processing (NLP) models will improve user experience by enabling users to obtain Quranic content that is specifically related to their queries through both basic keyword matching and reinforcement learning. Furthermore, the platform's capacity to provide thorough, comprehensive answers with thorough interpretations seeks to close the knowledge gap and make the teachings of the Quran available to a wider audience. This research is an inventive use of AI in the field of Islamic studies, in addition to offering the potential to increase accessibility to holy texts.

Table of Contents

[Chapter 1: Introduction](#_2s8eyo1) 9

[1.1 Goals and Objectives](#_17dp8vu) 9

[1.2 Scope of the Project](#_3rdcrjn) 10

[Chapter 2: Literature Review](#_26in1rg) 11

[2.1 Introduction](#_lnxbz9) 11

[2.2 Background and Problem Elaboration](#_35nkun2) 11

[2.3 Detailed Literature Review](#_1ksv4uv) 12

[2.3.1 Definitions](#_44sinio) 12

[2.3.2 Related Research Work 1](#_2jxsxqh) 12

[2.3.3 Related Research Work 2](#_z337ya) 13

[2.3.4 Related Research Work 3 1](#_z337ya)3

[2.3.5 Related Research Work 4 1](#_z337ya)4

[2.3.6 Related Research Work 5 1](#_z337ya)4

[2.4 Literature Review Summary Table 1](#_3j2qqm3)5

[2.5 Research Gap 1](#_1y810tw)7

[2.6 Problem Statement 1](#_4i7ojhp)9

[2.7 Conclusion](#_4i7ojhp) 20

[Chapter 3: Requirements and Design](#_2xcytpi) 21

[3.1 Requirements](#_1ci93xb) 21

[3.1.1 Functional Requirements](#_3whwml4) 22

[3.1.2 Non-Functional Requirements](#_2bn6wsx) 23

[3.1.3 Hardware and Software Requirements](#_qsh70q) 23

[3.2 Proposed Methodology](#_3as4poj) 24

[3.3 System Architecture](#_1pxezwc) 26

[3.4 Use Cases](#_49x2ik5) 27

[3.4.1 Use Case Diagram (user)](#_2p2csry) **27**

[3.5 Database Design *(Optional)*](#_147n2zr)

[3.6 Class diagram *(Optional)*](#_3o7alnk) **.**

[3.7 Sequence diagram *(Optional)*](#_23ckvvd)

[3.8 Any Other Artifact…](#_ihv636)

[3.9 Graphical User Interfaces(GUI) (*Optional)*](#_32hioqz)

[Chapter 4: Implementation and Test Cases](#_1hmsyys) 39

[4.1 Implementation](#_41mghml) 39

[4.1.1 Implementation of First Component/Algorithm](#_2grqrue) .

[4.2 Test case Design and description](#_vx1227) .

[4.2.1 Test case No.1](#_3fwokq0) 41

[4.2.2 Test case No.2](#_1v1yuxt) 42

[4.3 Test Metrics](#_4f1mdlm) 43

[Chapter 5: Experimental Results and Analysis](#_28h4qwu) 45

[Chapter 6: Conclusion and Future Directions](#_nmf14n) 50

[References](#_37m2jsg) 54

[Appendix](#_1mrcu09) .

[Appendix A: Guidelines](#_46r0co2) .

[Appendix B: Heading of Sample Appendix B](#_2lwamvv) .

[Formatting Guidelines](#_111kx3o) .

[Chapter 1: Heading 1](#_3l18frh) .

[1.1 Heading 2](#_206ipza) .

[1.1.1 Heading 3](#_4k668n3) .

[Tables and Figures](#_2zbgiuw) .

[Equations](#_1egqt2p) .

[Header/Footer](#_3ygebqi) .

[Other Formatting Guidelines](#_2dlolyb) .

[References 5](#_sqyw64)4

List of Tables

[Table 2.1 :](#_3cqmetx)Literature Review Summary Table15

Table 3.1 : Functional Requirements 22

Table 3.2 : Use Case Description (UC-U01) 28

Table 3.3 : Use Case Description (UC-U02) 29

Table 3.4 : Use Case Description(UC-U03) 30

Table 3.5 : Use Case Description(UC-U04) 31

Table 3.7 : Use Case Description(UC-U06) 32

Table 3.8 : Use Case Description(UC-U07) 34

Table 3.9 : Use Case Description(UC-U08) 35

Table 3.10 : Use Case Description(UC-U09) 36

Table 3.11 : Use Case Description(UC-U10) 37

Table 3.12 : Use Case Description(UC-U11) 38

Table 4.1: Test Case 1 41

Table 4.2 : Test Case 2 42

Table 4.3 : Test Case 3 42

Table 4.4 : Test Case 4 43

Table 4.5 : Test Matrix 1 43

List of Figures

Figure 3.1: System Architecture Diagram 27

Figure 3.2: Use Case Diagram(User) 29

Figure 3.3: Use Case Diagram(Admin) 40

Figure 5.1: Query Flow Diagram 53

**Chapter 1:**

**Introduction**

Quranic content is now easier to access because of the digitization of the Quran and the creation of AI-powered Islamic chatbots, but there are still a lot of holes in the current solutions. Basic Quranic searches and brief, automated responses are provided by websites like MyQuran.online and IslamandAI.com, however they lack thorough tafseer and frequently give users insufficient answers to their questions. In a similar vein, FlowGPT.com employs sophisticated GPT models for Quranic search nonetheless, its response size and absence of comprehensive references are its limitations. According to research, such as the publication "The Holy Quran: Digitization Challenges and Concerns," it can be challenging to guarantee the precision and comprehensiveness of digital readings of the Quran. By offering a platform that provides thorough tafseer and contextually rich, comprehensive responses, the ChatWithQuran project seeks to remove these constraints and meet the demands of users who want to comprehend the Quran more deeply.

The Quran is the main source of guidance and knowledge for millions of people worldwide. However, especially for those who are not familiar with the Quran, it might be challenging to locate specific verses, understand what they mean, and look into the associated tafseer (interpretation). With the use of an easy-to-use search interface, our project, ChatWithQuran, seeks to develop a web-based platform that will enable users to quickly query Quranic verses and acquire accurate translations and tafseer. By incorporating advanced Natural Language Processing (NLP) algorithms, the platform will enhance the search experience by allowing users to find relevant verses even when their queries are vague or incomplete. In order to foster a deeper engagement and understanding of the text, this effort aims to make the Quran easier to read and more accessible.

**1.1 Goals and Objectives:**

Make Quranic knowledge more accessible by offering a user-friendly platform that enables a wide audience to access Quranic passages, translations, and tafseer. Accurate and Context Rich Responses: By employing advanced natural language processing techniques, it ensures that users obtain comprehensive, accurate, and relevant answers to their queries. Close the Gaps in Current Islamic Chatbot Systems: To address the limitations of the current Quranic chatbots, offer comprehensive, reliable, and in-depth tafseer for a deeper understanding.

**1.1.1 Objectives:**

Use Natural Language Processing (NLP) methods to evaluate and interpret user enquiries.

By teaching a specialised large language model (LLM) to translate queries to a structured database, can guarantee 100% accuracy.

Continuous testing and improvement of NLP algorithms is necessary to improve the quality and relevancy of the answers.

To serve a range of user needs and maintain credibility, provide translations and tafsir from two distinguished experts.

Ensure efficient processing, retrieval, and storage of Quranic verses, translations, and tafseer from the dataset.

Provide a user-friendly, accessible, and intuitive interface so that users can easily enter queries and retrieve results.

Collaborate with the backend and NLP teams to ensure smooth operation, precise result display, and overall user satisfaction.

1.2 Scope of the Project

Give consumers access to a well-organised database that contains Quranic verses, translations, and thorough tafseer so they can obtain accurate Islamic information for each verse. Use state-of-the-art natural language processing (NLP) techniques, such as semantic approach, to accurately evaluate user searches and provide relevant verse, translation, and tafseer results. Even without particular keywords, users can find verses and tafseer that are contextually pertinent to their query by using semantic search methods (like LLM). Provide consumers with options for both flexible and in-depth tafseer responses based on their preferences and the query's level of complexity. Create an easy-to-use online interface with straightforward navigation so that users can see, search, and interact with Quranic material with ease. Answers to user enquiries should be thorough, contextually accurate, and consistent across inquiry kinds in order to maintain clarity and comprehension depth.

**Chapter 2:**

**Literature Review**

**2.1 Introduction**

Users can utilise the website to enter keywords or queries and receive accurate, contextually relevant responses, including translations, verses from the Quran, and tafseer (interpretations). Unlike many other solutions that only provide limited and often inadequate responses, ChatWithQuran aims to provide comprehensive and broad answers by utilising a wide collection of Quranic knowledge. The platform's features, such as keyword matching search, and alternatives to summarised tafseer, are intended to be a reliable source for anyone wishing to learn more about the Quran. The digitisation of the Quran and the development of AI-powered Islamic chatbots have made it easier to access Quranic text, yet the existing solutions still have several flaws.

ChatWithQuran seeks to improve the user experience for people all around the world by bridging the gap between traditional Quranic study and contemporary digital accessibility with an easy-to-use interface and multilingual support. By offering a platform with comprehensive, contextually rich responses together with in-depth tafseer, the ChatWithQuran initiative seeks to remove these restrictions and meet the demands of people who want to comprehend the Quran.

2.2 Background and Problem Elaboration

ChatWithQuran seeks to improve the user experience for people all around the world by bridging the gap between traditional Quranic study and contemporary digital accessibility with an easy-to-use interface and multilingual support. By offering a platform that provides thorough tafseer and contextually rich, in-depth responses, the ChatWithQuran initiative seeks to remove these restrictions and meet the demands of users who want to get a deeper comprehension of the Quran.

Explaining tafseer in a way that honours and elucidates the diverse ways that different schools of thought interpret the same Quranic verses can be challenging. It is necessary to have an easy-to-use, interactive platform that can adapt to the individual needs of every

learner and provide customised guidance through Quranic content. Many fields are seeing an increase in the use of conversational AI tools for self-study, but Islamic studies in particular, including the interpretation of the Quran, have not yet fully benefited from these advancements. Simple keyword matching is used by most Quranic systems currently in use, which limits their ability to handle complex queries when the user may not know exactly what terms to search for.

**2.3 Detailed Literature Review**

The use of Natural Language Processing (NLP) in Quranic studies has grown in popularity recently as a means of enhancing understanding and interpretation of holy texts. The goal of this study is to enhance an LLM or BERT-based model designed specifically for multilingual jobs in order to classify and assess Quranic verses and their translations. By using a dataset that includes Arabic text, multiple English translations, and tafsir (interpretation), the study seeks to address the complexity of semantic variations between languages and interpretations. Preprocessing techniques like tokenization, stopword removal, and diacritical removal ensure that the data is prepared for deep learning. The objective is to create an AI-powered system that can categorize verses according to their themes (such as Surah or Ayah), improving accessibility for academics, teachers, and everyday readers. This project not only exhibits the potential of natural language processing (NLP) in multilingual and multifaceted textual analysis, but it also builds a bridge between technology and theology.

**2.3.1 Definitions**

The primary objective of ChatWithQuran is to develop an AI-based system for classifying and assessing Quranic verses, their translations, and tafsir (interpretations). Using state-of-the-art machine learning approaches such as multilingual BERT or LLM, it aims to provide accurate classification and deeper insights into Quranic texts for better accessibility and comprehension.

To classify and assess Quranic verses and their translations, ChatWithQuran develops an AI-powered model using state-of-the-art natural language processing techniques. The system trains a multilingual LLM or BERT-based model on Quranic text and translations in an effort to streamline user exploration and classification of Quranic content.

**2.3.2 Related Research Work 1**

The goal of this web-based Quran platform is to make Quranic texts, translations, and interpretations easily accessible to users. Using state-of-the-art natural language processing techniques, the platform's multilingual query feature enables users to search and examine Quranic text in Arabic, English, and other languages. In order to increase the precision and relevance of search results and ensure that users can find specific verses or subjects fast, it uses state-of-the-art machine learning models, such as LLM or BERT and its multilingual variants. The website also offers a large number of translations and tafsir (interpretations) to satisfy a variety of user preferences and academic needs. The platform's user-friendly interface and advanced backend make it a valuable tool for academics, students, and general audiences seeking to understand the Quran more thoroughly.

**2.3.3 Related Research Work 2**

A web-based Quranic platform designed to provide users with convenient access to Quranic texts, interpretations, and translations is thoroughly examined in this study. In order to optimise keyword search and ensure accurate retrieval of Quranic text in several languages, the platform employs advanced machine learning and natural language processing (NLP) techniques, including multilingual models such as BERT or LLM. By including many translations and tafsir resources, the system promotes a greater understanding of the Quran and benefits a wide range of users, including academics, students, and regular users. The study highlights the platform's architecture, which combines an intuitive user interface with sophisticated backend processing to enable efficient exploration of themes, keywords, and situations inside the Quran. This study demonstrates how AI-powered solutions can increase the accessibility of religious texts and emphasises the importance of technology in assisting Quranic studies.

**2.3.4 Related Research Work 3**

IslamandAI, a beta-version AI-powered chatbot that can offer concise and accurate responses to frequently asked Quranic enquiries, is presented in this study. To facilitate communication, the chatbot interprets user queries or keywords using state-of-the-art natural language processing (NLP) models. It then retrieves relevant Quranic verses, translations, or explanations. This early-stage system aims to make Islamic knowledge more accessible by giving people a quick and efficient way to learn about the Quran. Because the creation places a high value on correctness, simplicity, and usefulness, it is suitable for customers seeking a basic understanding or clarification on common Quranic themes. The study shows how AI may address problems like upholding theological truth and effectively handling a range of linguistic inputs while assisting people in developing a deeper connection with Islamic beliefs.

**2.3.5 Related Research Work 4**

From statistical models to today's potent Large Language Models (LLMs), the essay traces the development of machine language capabilities. Humans are linguistically inclined by nature, while machines need sophisticated AI algorithms to comprehend and produce language. The development of LLMs, which are fuelled by deep learning, massive datasets, and a wealth of computer power, has greatly enhanced machines' capacity to handle challenging language problems.

The study places a strong emphasis on LLMs, classifying them into four categories and emphasising their function in computer vision and natural language processing. As an example of a state-of-the-art application in this field, it also presents ChatGPT (based on GPT-3.5), which is renowned for producing responses that resemble those of a human.

**2.3.6 Related Research Work 5**

This study offers a thorough analysis of the Text-to-SQL job, which attempts to translate natural language (NL) queries into precise SQL statements in order to close the gap between non-technical users and sophisticated relational databases.It draws attention to the difficulties with cross-domain generalisation, schema comprehension, and ambiguity in modern systems. In order to improve SQL generation, the paper presents Retrieval-Augmented Generation (RAG), a promising technique that combines a retrieval module with a generating module to dynamically fetch and integrate pertinent schema or information. It also shows how text-to-SQL systems have advanced from rule-based techniques to deep learning and pre-trained language models, demonstrating notable advancements in comprehending and producing retrieved data from databases. The study highlights RAG and Graph RAG as cutting-edge methods to increase accuracy and adaptability in practical applications and ends with a taxonomy of existing methodologies.

**2.4 Summary Table**

## Table 2.1 : Literature Review Summary Table

| S.No | Year | Title | Methodology | Results | Limitations |
| --- | --- | --- | --- | --- | --- |
|  | 2024 | Arabic Chatbots Challenges and Solutions | To make chatbot intelligent using NLP techniques | To make chatbot intelligent using NLP techniques | Manual data collection limits dataset size and quality |
|  | 2024 | A New Semantic  Approach for the  Holy Quran | Proposes a new semantic search approach for the Holy Quran using three modules | Proposes a new semantic search approach for the Holy Quran using three modules | No use transformer-based models to detect semantically related verses to the user’s query |
|  | 2024 | Semantic search engine for Holy Quran | The project aimed to improve Quranic search by collecting and evaluating multiple data sources, testing various models and using techniques like cosine similarity and word2vec to find the best way to match Quranic text with user queries. | Shot  GPT-based Responses | The main challenge was the lack of high-quality training data, which affected the accuracy of results, particularly with pre-trained models and topic modeling techniques. Despite some success with word2vec. |
|  | 2025 | A Survey on Large Language Models: Applications,  Challenges, Limitations, and Practical Usage | The project uses state-of-the-art deep learning (transformer-based LLMs) trained on large datasets and fine-tuned for natural language tasks, with a particular focus on conversation and understanding, as seen in models like ChatGPT. | High-quality conversation generation Strong language understanding capabilities Transferability to vision-language models in fields like Computer Vision | large language models (LLMs) have several limitations. They lack true understanding and instead rely on pattern prediction, which can lead to plausible but inaccurate responses. The quality of their output heavily depends on the data they were trained on, making them susceptible to biases, misinformation, and gaps in domain-specific knowledge |
|  | 2025 | From Natural Language to SQL: Review of  LLM-based Text-to-SQL Systems | It identifies key challenges such as linguistic ambiguity, complex SQL operations, schema understanding, and cross-domain generalization. | The key results from the given paper highlight that Retrieval-Augmented Generation (RAG) significantly improves the performance of Text-to-SQL systems | major limitations such as schema understanding, query complexity, and domain generalization |

## 

## **2.5 Research Gap**

The research gap for the ChatWithQuran project lies in addressing limitations found in current Addressing the shortcomings of existing AI-driven Quranic chatbots and digital resources which frequently fall short in terms of tafseer (interpretation) completeness, inquiry depth, and contextual understanding represents the research gap for the ChatWithQuran project. Current websites like FlowGPT.com, IslamandAI.com, and MyQuran.online typically offer simple keyword search features and succinct responses. However, particularly for users looking for in-depth interpretations of complicated inquiries, they are unable to provide nuanced, thorough, and contextually appropriate solutions. An insufficient comprehension of the verses results from the fact that many of these platforms do not completely incorporate the tafseer with the Quranic text in a way that enables users to interact with the content in a meaningful way.

Furthermore, there is a weakness in the platforms' comprehension and response to natural language queries, especially when those queries are ambiguous or contextually complicated. For instance, without user-supplied keywords, many AI-based solutions cannot correctly handle semantically rich inquiries or decipher the deeper meanings of phrases. Their response depth and length are likewise limited, either due to simplified language models or technical limitations.

By using natural language processing (NLP) techniques, the ChatWithQuran project seeks to close these gaps by providing context-aware, semantically correct responses, including comprehensive and accurate tafseers for verses. ChatWithQuran aims to provide a more satisfying platform for customers looking for both fundamental and in-depth insights into Quranic verses by emphasizing the provision of thorough interpretations, enhanced natural language question handling, and a more engaging user experience. This strategy fills the noted research vacuum and offers creative fixes for digital Islamic resources.

**2.5.1 MyQuran.online**  
With basic search capabilities, the website MyQuran.online provides an extensive library of Quranic passages and translations. Although it makes Quranic content accessible, the platform's capacity to offer comprehensive tafseer and contextual explanations is constrained. The platform might not include enough in-depth readings for users who want to grasp particular passages more thoroughly, which would make it less helpful for people who need thorough direction.

**2.5.2 IslamandAI.com (ChatBetaVersion)**  
The IslamandAI.com chatbot represents an early attempt to combine AI and Islamic principles is the chatbot on IslamandAI.com. This chatbot's beta version provides brief, automated answers to user inquiries. Nevertheless, it has serious drawbacks, such as lacking tafseer and providing insufficient answers. Although the brief answers might be adequate for simple questions, they fall short in offering the thorough, contextualized explanations that people frequently look for when studying the Quran.

**2.5.3 FlowGPT.com (Quran-GPT)**  
The FlowGPT.com platform introduces a more advanced AI-powered Quranic search engine as presented by the FlowGPT.com platform, which uses GPT-based models to respond to user queries. Notwithstanding its potential, the platform is restricted by the small number of responses and the lack of citations or thorough justifications. Users might get pertinent but cursory responses that don't completely address the intricacies of their questions or offer thorough tafseer.

**2.6 Problem Statement**

The growing need for accurate and easily accessible Quranic information makes it difficult to respond to user inquiries in a timely, trustworthy, and contextually appropriate manner. To find certain verses or explanations, traditional search techniques may call for a great deal of physical labor and skill. The procedure is further complicated by linguistic diversity, differing interpretations, and the requirement for theological truth. By creating an AI-powered chatbot, including cutting-edge natural language processing models to process user queries, and providing precise Quranic responses, this project seeks to address these problems. The method aims to improve user engagement and comprehension while upholding Islamic values by simplifying access to Quranic teachings.

**2.7 Conclusion:**

In conclusion, to sum up, ChatWithQuran wants to develop into a groundbreaking online tool that enables users to study the Quran in a way that is clear, comprehensive, and contextual. This project intends to provide a more thorough and user-friendly experience by correcting the shortcomings of the current platform, such as incomplete tafseer, poor response accuracy, and a lack of customisation. Our project will incorporate sophisticated natural language processing to deliver contextually aware responses, enabling users to ask questions in a natural way and get comprehensive and pertinent answers. As a result, users from a variety of language backgrounds and comprehension levels will find the Quran's lessons more approachable and captivating.

Additionally, our project is intended to be a dependable and flexible platform that is a useful resource for academic institutions as well as individual individuals. With the project's comprehensive translations, linguistic support, and intuitive interface, users will be able to interact with the Quran in a reliable, accurate, and customized way. The ultimate goal of this project is to close current gaps in digital Quranic materials and promote a deep relationship with the Quran for scholarly and personal study. With these improvements, our effort will make a substantial contribution to improving the accuracy, accessibility, and applicability of Quranic knowledge for users worldwide. The focus of Chapter 3 will be on outlining the process and strategy we will employ in order to create our solution.

**Chapter 3:**

**Requirements And Design**

The ChatWithQuran project to fulfil its objective of developing an all-inclusive, user-friendly platform for Quranic queries, a number of prerequisites must be met. First and foremost, the platform needs a comprehensive and organized dataset with numerous translations, Quranic verses, and thorough tafseer for every verse. For this dataset to accurately handle a variety of user queries, Natural Language Processing (NLP) capabilities must be included. To make sure the platform can efficiently answer inquiries with different languages, basic keyword matching and more sophisticated functions are required. When a broad query is entered, the system should also include summarizing tools to offer succinct, pertinent interpretations. This would solve the shortcomings of current platforms, which frequently provide answers that are shallow, and guarantee that customers receive thorough yet understandable responses.

In order to improve ChatWithQuran's usability and accessibility, the platform must also have a clear, simple user interface that enables users to submit queries and get answers without the need for technical expertise. To reach a wider audience, this interface should have capabilities like search history, keyword suggestions, and language support. To handle queries and provide seamless integration between the frontend and the NLP model, the backend system needs reliable API endpoints. In order to handle the possible pressure from numerous users, it is also essential to create a scalable infrastructure. Given the religious significance of the Quranic text, security and data integrity procedures must also be in place to safeguard the content and guarantee the veracity of the information displayed.

**3.1 Requirements**

### 3.1.1 Functional Requirements

Table 3.1 : Functional Requirements

| ID | Category | Requirement  Description |
| --- | --- | --- |
| FR1 | User Query Processing | Allow users to input queries and retrieve relevant Quranic verses, translations, and tafseer. Support basic and search. |
| FR2 | Result Generation and Display | Display exact matches along with related content in an organized, responsive format. Provide summarized tafseer for broad queries. |
| FR3 | User Interface and Accessibility | Ensure a user-friendly interface with responsive design for various devices, and accessible navigation features. |
| FR4 | Security and Data Integrity | Implement secure data handling to protect Quranic content and user privacy. Prevent unauthorized modifications of religious texts. |

**3.1.2 Non-Functional Requirements**

**3.1.2.1** **Accuracy**:

Check to determine if the Quranic verses, translations, and tafseer on the site are accurate, authentic, and sourced from reputable Islamic experts.

**3.1.2.2 Performance**:

The platform's search results should load in two to three seconds for optimal user experience.

**3.1.2.3 Scalability**:

The platform's search results should load in two to three seconds for optimal user experience.

**3.1.2.4 Usability**:

With tooltips, a help section, and easy navigation, the interface should be user-friendly.

**3.1.2.5 Security**:

Prevent unwanted access to or alteration of Quranic data, and make that user information and inquiries are handled securely.

**3.1.2.6 Data Privacy**:

Observe data privacy regulations by ensuring that user data is securely stored and encrypted, if accounts are supported.

### 3.1.3 Hardware and Software Requirements

The hardware required to train the model on the dataset can vary depending on the amount of data, the complexity of the model architecture, and the training process. However, for moderately sized datasets, the following requirements are considered enough.

**3.1.3.1 Processor**:

For seamless processing, a minimum Intel i5 or similar processor with multi-core capabilities is required.

**3.1.3.2 RAM**:

For best results while processing queries and retrieving data, 8GB or more is required.

**3.1.3.3 Storage**:

For database and content storage, there should be a minimum of 256GB of accessible storage.

**3.1.3 Software Requirements:**

**3.1.3.1 Operating System:**

For our system to be developed and deployed, the operating system is necessary.

**3.1.3.2 Web Server**:

local web application.

**3.1.3.3 Database Management System**:

For storing Quranic data, translations, and tafseer, use MySQL or SQLite.

**3.1.3.4 Programming Languages**:

**3.1.3.4.1 Frontend**: Python

**3.1.3.4.2 Backend**: Python.

**3.1.3.5 Natural Language Processing Tools**:

For text processing and query comprehension, SpaCy, Streamlit, vanna.ai(RAG), GPT3(LLM) etc.

**3.2 Proposed Methodology**

The proposed methodology for this project involves the following steps to ensure the effective development of an AI-powered Quranic chatbot:

**3.2.1 Data Collection and Preprocessing:**

Assemble a comprehensive collection of Tafsir, translations (including a number of English translations), and Quranic texts from reputable sources. Clean and preprocess the data by removing redundant information, superfluous wording, and errors while preserving theological accuracy.

**3.2.2 Text Representation:**

Modern multilingual tokenizers like LLM should be used to tokenize the Quranic text and translations. Handle linguistic variation by ensuring that English language is supported using the appropriate NLP techniques.

**3.2.3 Model Selection and Training:**

Adjust Retrieval Augmentation Generation (RAG) models that have already been trained (like LLM and vanna AI) for tasks involving question answering or sequence categorization.To identify user intents and obtain pertinent verses and translations, train the model using the Quranic dataset.

**3.2.4 System Architecture Development:**

Build a backend system that integrates the learnt AI model and has an easy-to-use user interface. Allow the chatbot to receive user input (a keyword or question) and respond with accurate Quranic responses.

**3.2.5 Evaluation and Optimization:**

Use a range of questions to test the chatbot's accuracy, usefulness, and operation. Handle edge cases, adapt the model for better contextual knowledge, and refine it in response to user feedback.

**3.2.6 Ethical and Theological Review:**

Conduct a thorough review by Islamic scholars to validate the accuracy and compliance of responses with Islamic teachings. Implement safeguards to handle ambiguous or sensitive queries responsibly.

**3.2.7 Deployment and User Feedback:**

Set up the chatbot on a web-based platform and monitor its operation in real time. Get feedback from users to improve the system, improve the user experience, and gradually add additional features.

**3.3 System Architecture**

The Quranic chatbot's system architecture is designed to provide functionality, scalability, and reliability. The User Interface Layer, which offers a web-based platform where users may enter questions or keywords and receive easily navigable responses, is the main point of contact. The Application Layer processes these inputs and generates relevant results, such as translations and verses, using a query processor, intent recognition module, and an AI model tailored for Quranic situations. With the use of a specialised search engine for keyword-based searches, the Data Layer stores and retrieves structured Quranic data using a knowledge base and database.

The middleware layer makes it easier for the application and data levels to communicate by using caching techniques. The Deployment Layer, which also comprises cloud hosting and monitoring tools, is used to provide the system in order to guarantee scalability and dependability. Finally, to facilitate continuous development and ensure the chatbot successfully meets user needs, a User Feedback and Monitoring Layer collects user input and keeps an eye on system performance.

**3.3.1 System Architecture Diagram**

Given architecture diagram is a visual representation of the system’s structure, showing its components, their relationships, and how data flows between them.

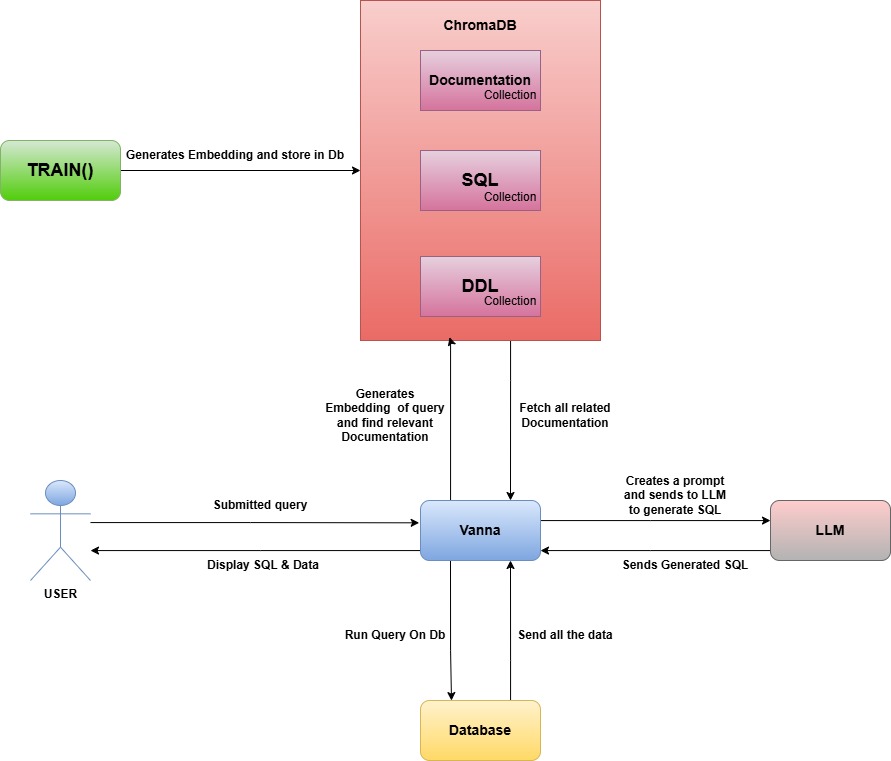


Fig 3.1: Architecture Diagram

* 1. **Use Cases**

Below is a description of how a user (*actor*) interacts with a system to achieve a specific goal. It defines the system’s behavior in response to a user's actions and helps capture functional requirements.

**3.4.1 Use Case Diagram (User)**

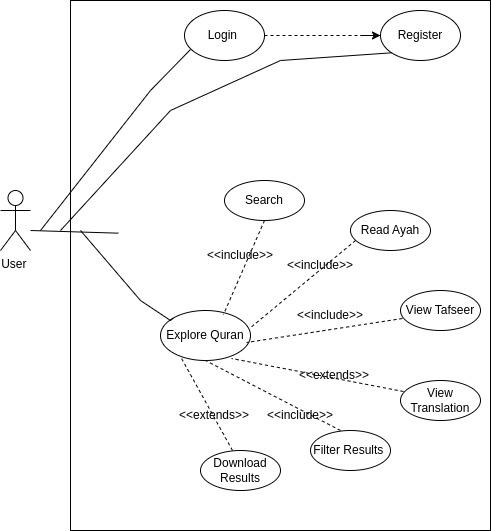
****

Fig 3.2: Use Case Diagram

**3.4.1.1 Use Case Description (UC-U01)**

Table 3.2 Use Case Description (UC-U01)

| UC ID | UC-U03 |
| --- | --- |
| UC Name | Search Queries |
| Primary Actor | User |
| Pre-Condition | User should have successfully login the website |
| Post-Condition | User’s search query will be successfully sent. |
| Main Flow | 1. The User will click on the Search Bar 2. They will write any type of Query in the Search Bar 3. Their previous Search history will be available just below the Search Bar 4. After writing the Query, they will press the search icon (or press Ctrl+enter). 5. This will send the User's search query for further processing 6. User will be navigated to the results page |
| Alternate Flow | 1.1 If a problem occurs with the UI, then the Search Bar will not open  1.2 The website will display a crash message, and it will ask the user to try again  3.1 User can click on their previous searched query from the available list  3.2 The query will appear on the Search Bar  3.3 They can then press the search icon to search for the query |

**3.4.1.2 Use Case Description (UC-U02)**

Table 3.3 : Use Case Description (UC-U02)

| UC ID | UC-U02 |
| --- | --- |
| UC Name | View Results |
| Primary Actor | User |
| Pre-Condition | User should have successfully written a Search Query |
| Post-Condition | User will have successfully opened a Search Result to see it's details |
| Main Flow | 1. User will be shown with list of search results retrieved from the server 2. These results will include Verses, Tafseer and Translation. 3. When an item is clicked, User will be navigated to the Details Page 4. All details of the item (Verse/ Translation/ Surah), that are available, will be displayed on the page 5. User can also go back to open another item |
| Alternate Flow | 1.1 If there are no results available, User will be shown a “No Results Available” message  1.2 User will be asked to try again(i.e. to write another search query and try searching again)  2.1 The results can have any mixture of items that are displayed  2.2 User can click on any type of Item, to see it's details  4.1 If a Google Search Result Item is clicked, the User will be redirected (with message) |

**3.4.1.3 Use Case Description (UC-U03)**

Table 3.4 : Use Case Description (UC-U03)

| UC ID | UC-U03 |
| --- | --- |
| UC Name | Read Ayah |
| Primary Actor | User |
| Pre-Condition | User should have clicked on the Search Bar |
| Post-Condition | User will have successfully manipulated the Query |
| Main Flow | 1. When User clicks on the Search Bar, they will be shown with their search results 2. If the user clicks on an item in the Ayah column, it will appear in the Ayah ,or User can search it |
| Alternate Flow | 1.1 The User can also go to the search bar and search the specific Ayah.  1.2 Users will click on the “Ayah column” and search ayah.  1.3. The User will be shown their Search Ayah in a chronological manner  1.4 If the User clicks on item, it will be searched, taking the user to its particular results |

**3.4.1.4 Use Case Description (UC-U04)**

Table 3.5 : Use Case Description (UC-U04)

| UC ID | UC-U04 |
| --- | --- |
| UC Name | Filter Results |
| Primary Actor | User |
| Pre-Condition | Search results based on user's search query should be available |
| Post-Condition | User will successfully see filtered items based on a filter value |
| Main Flow | 1. Filter options will be available on the top, just above the results 2. Its values will be All, Verses and Translations. 3. User can click on any value that they want 4. Based on the specific value, the search results will be filtered 5. And they will be shown again to the user |
| Alternate Flow | 1.1 If there are no results available, User not be shown any filter values  1.2 User will be asked to try again(i.e. to write another search query and try searching again)  2.1 The filter items can be more or less depending on the type of the results that are received  2.2 If the results do not contain verses or Translation, that particular filter value may not be shown  2.3 If the results contain external redirecting results, it may be an extra filter value that may appear as well |

**3.4.1.5 Use Case Description (UC-U05)**

Table 3.6: Use Case Description (UC-U05)

| UC ID | UC-U05 |
| --- | --- |
| UC Name | Filter All |
| Primary Actor | User |
| Pre-Condition | User should have successfully written a Search Query and have the results |
| Post-Condition | User will successfully see all the results that are available |
| Main Flow | 1. By default, ALL the search results will be shown on the screen 2. These search results may include Verses, and redirecting results 3. If the user is on any other filter option, he may click on the ‘All’ option 4. When user clicks on the ‘All’ option, the original list of results will be shown 5. This list will be stored in Cache, so they will not be called through Network |
| Alternate Flow | 3.1 If the user is on ‘All’ option, and they click on the ‘All’ option, there will be no change  3.2 User will have to click on any other filter option to see the effects of the All option |

**3.4.1.6 Use Case Description (UC-U06)**

Table 3.7: Use Case Description (UC-U06)

| UC ID | UC-U08 |
| --- | --- |
| UC Name | Filter Verses |
| Primary Actor | User |
| Pre-Condition | User should have successfully written a Search Query and have the results |
| Post-Condition | User will successfully see the Verses results that are available |
| Main Flow | 1. By default, All the search results will be shown 2. User will click on the Verse option from the filter values 3. All the results, that are verses, will be shown 4. Only the verses will be shown to the user, no matter how many they are 5. User will be able to go back to see All the results, by clicking on the All option 6. By clicking on any Verse Search Result, the item will be opened to show it's details |
| Alternate Flow | 3.1 The original (all) results will be filtered  3.2 If the user wants to choose any other filter option, the original list will be used for this purpose again |

**3.4.1.7 Use Case Description (UC-U07)**

Table 3.8: Use Case Description (UC-U07)

| UC ID | UC-U07 |
| --- | --- |
| UC Name | Filter Translations |
| Primary Actor | User |
| Pre-Condition | User should have successfully written a Search Query and have the results |
| Post-Condition | User will successfully see the Translation results that are available in database |
| Main Flow | 1. By default, All the search results will be shown 2. Users can filter the Translations by clicking on the search button. |
| Alternate Flow | 3.1 The original (all) results will be filtered  3.2 The original results will still remain in the Cache. They will not be deleted  3.3 If the user wants to choose any other filter option, the original list will be used for this purpose again |

**3.4.1.8 Use Case Description (UC-U08)**

Table 3.9: Use Case Description (UC-U08)

| UC ID | UC-U10 |
| --- | --- |
| UC Name | Download Results |
| Primary Actor | User |
| Pre-Condition | User should have successfully got the results. |
| Post-Condition | User will have successfully clicked on the download button. |
| Main Flow | 1. User will click on the Download option that will be available top of the original text 2. Users can download any item (Verse, Surah, Tafseer, etc.) 3. When they click on the Download option, they will be promoted to select the desired platform 4. The original text (Arabic Text) and the English Text with some supporting details, will be download on the user selected platform |
| Alternate Flow | 3.1 If there are internet issues, the selected platform will not open  3.2 User will be shown with an appropriate message that the item cannot be shared at the moment  3.3 The user will be asked to try again  4.1 The Arabic Text will be stored in Unicode  4.2 If the certain platform doesn't accept Unicode, user will be shown an error message  5.1 The link will be appended with the texts at the very top |

**3.4.1.9 Use Case Description (UC-U09)**

Table 3.10: Use Case Description (UC-U09)

| UC ID | UC-U11 |
| --- | --- |
| UC Name | Bookmark |
| Primary Actor | User |
| Pre-Condition | User should have successfully opened an item |
| Post-Condition | User will successfully bookmark the opened item |
| Main Flow | 1. The User will open their desired item (Verse, Tafseer, etc) 2. They will then click on the Bookmark button just below the Original text 3. When the user clicks on the button, they will be shown a message that the item has been successfully Bookmarked 4. If the user clicks on the bookmark button again 5. That particular item will be removed from Bookmarks, and user will be shown a message that the item has been successfully removed from Bookmarks |
| Alternate Flow | 3.1 If there is Internet Connectivity Issue, the bookmark feature will not work  3.2 The user will be shown a message there is No Internet Connection  3.3 The User will be asked to try again |

**3.4.1.10 Use Case Description (UC-U10)**

Table 3.11: Use Case Description (UC-U10)

| UC ID | UC-U10 |
| --- | --- |
| UC Name | Delete Bookmarks |
| Primary Actor | User |
| Pre-Condition | User should have successfully opened the Profile Page |
| Post-Condition | User will have successfully deleted a Bookmark or all Bookmarks |
| Main Flow | 1. User will open the Profile Page, and click on Bookmarks 2. A list of all the Bookmarked Verses, Surah, Tafseer, etc. will be shown to the user 3. User can press the Bookmark, and it will open the item to show it's details 4. User can press the trash icon next to the item to delete that particular Bookmark 5. User can also press “Delete All” to delete all the items that are bookmarked by User |
| Alternate Flow | 3.1 When User clicks on a Bookmarked Item, it will be opened to show it's details  3.2 If there are connectivity issues, User will be shown a message to Try Again  3.3 If the Bookmarked Item is an external redirecting search item, User will be prompted to select their preferred browser to open the result |

**3.4.2 Use Case Diagram (Admin)**

To provide control and flexibility in managing Quranic data and user access, ensuring the content is accurate, updated, and structured properly for NLP or AI queries.

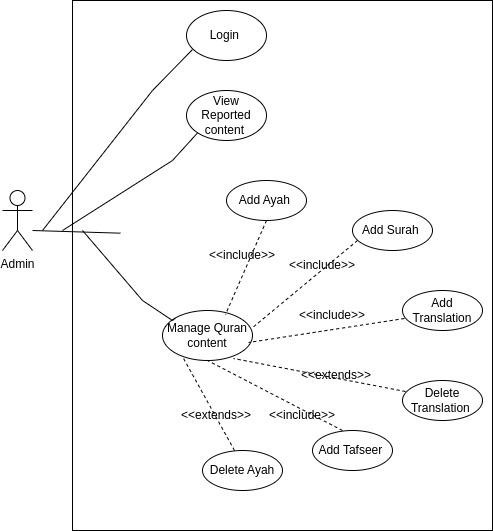


Fig 3.2: Use Case Diagram(Admin)

**Chapter 4:**

**Implementation and Test Cases**

**4.1 Introduction**

This chapter's main goal is to develop a full stack web application that facilitates user comprehension and access to relational database data. Although LLMs already help generate SQL queries, they frequently make mistakes because they don't have the database schema information. The difficulty is in figuring out how to incorporate the database schema knowledge of a particular database into the LLM so that it can provide more sophisticated and accurate help than it could if it didn't know the database schema. By investigating several options for incorporating said domain information, including few-shot learning, fine-tuning an LLM, and other paradigms, this study seeks to address this issue. Finally, a proof of concept must be implemented as a full stack web application.

**4.2 Implementation**

Building an effective system that can comprehend the user's query and collect pertinent data is the main goal of this project's implementation. The main elements of the system, the algorithms employed, the technologies put into use, and the system's general architecture are all covered in detail below.

**4.2.1 Implementation Details:**

**4.2.1.1 Platform:**

Streamlit, a well-known open-source Python framework for creating interactive web apps specifically for data science, machine learning, and AI-powered tools, served as the platform for this project.

**4.2.1.2 Functionality:**

The goal of the Streamlit-based Quran AI Assistant project is to give users an intelligent, interactive platform where they can ask questions about the Quran and get knowledgeable, contextually relevant responses.

**4.2.2 Model Training:**

Using domain-specific Quranic data, the software trains a pre-built model (Vanna AI) and refines it.Vanna is a tool that uses LLM to retrieve data from a database in natural language.

**4.2.4 Text Extraction**

Your Quran AI Assistant project uses a SQLite database (Quran translation) to extract text from the dataset. Quranic verses, translations, and tafsir are all included in this dataset and are kept in a table called "Quran translation."

**4.2.5 Text to SQL queries:**

When a user inputs a question in natural language, such as "What does the Quran say about forgiveness?" In this project Vanna and GPT3(LLM) are AI models, is used by the system to interpret the query. After that, LLM automatically generates a SQL query that looks for pertinent verses, translations, and tafsir in the Quranic database.

**4.2.5.1 Implementation Details:**

**4.2.5.2 Platform:**

The primary platform on which the web application was developed. It makes the user interface (display, authentication, chat input, etc.).

**4.2.5.3 Libraries :**

**Vanna AI** Retrieve the data from database its a retrieval augmentation generation model   
**SQLite** runs the query on Quranic data.  
**Pandas** handles the results.  
**Streamlit** displays the formatted response to the user.

**4.2.6 Frontend Functionalities:**

**Objective**: Provide an intuitive interface for users to interact with the system.

**4.2.6.1 User Authentication Module:**

This file uses a straightforward flat file structure to manage user registration, login, and sessions.

**4.2.6.2 Result Search :**

Users can search the specific verse or tafseer from a given output which is extracted from the database.

**4.2.6.4 Download Results :**

Once a query is processed Users can download the data in text file or csv file.

**4.3 Test Case Design and Description**

Test cases were created for this project in order to verify the Quran AI Assistant's frontend and backend features, such as user authentication, natural language processing, database querying, and user interface behaviour. Every test case describes a particular situation with specified inputs, anticipated results, and the relevant module in charge.

**4.3.1 Test Case 1**

Table 4.1: Test Case 1

| User registration with valid details | |
| --- | --- |
| Test case ID | TC-001 |
| Test case version | v1.0 |
| objective | Verify that a user can register successfully with valid information. |
| comments |  |
| Passed / Failed | passed |

**4.3.2 Test Case 2**

Table 4.2: Test Case 2

| Submit Quran Question | |
| --- | --- |
| Test case ID | TC-002 |
| Test case version | v2.0 |
| objective | Ensure the system can process a user’s question and return a valid Quranic response. |
| comments | need to do proper Tokenization |
| Passed / Failed | Passed |

**4.3.3 Test Case 3**

Table 4.3: Test Case 3

| Display Results in Grid | |
| --- | --- |
| Test case ID | TC-003 |
| Test case version | v3.0 |
| objective | Check if results appear in a styled, searchable, and scrollable table. |
| comments |  |
| Passed / Failed | Passed |

**4.3.4 Test Case 4**

Table 4.4: Test Case 4

| Data Download Button | |
| --- | --- |
| Test case ID | TC-004 |
| Test case version | v4.0 |
| objective | Validate CSV download functionality from grid view. |
| comments |  |
| Passed / Failed | Passed |

## 

## **4.4 Test Metrics**

Test metrics are essential for evaluating the effectiveness, quality, and completeness of the software testing process. They provide measurable insights into how well the system performs under test conditions and help identify areas for improvement. This section outlines the key testing metrics such as the total number of test cases developed, passed, or failed, and evaluates performance using metrics like Test Case Defect Density and Test Case Effectiveness. These indicators guide quality assurance teams in assessing the reliability and robustness of the system, ensuring that the final product meets its functional requirements.

### 

### 

### 

### 

### 4.4.1 Sample Test case Metrics

Table 4.5: Test Matrix 1

| **Metric:** | **Calculation** | **Result** |
| --- | --- | --- |
| **Number of Test Cases:** | Total test cases developed : 4 | 4 |
| **Number of Test Cases Passed:** | Number of passed: TBD after execution. | 4 |
| **Number of Test Cases Failed:** | Number of failed: TBD after execution. | 0 |
| **Test Case Defect Density:** | (No of test cases failed \* 100)  No of test cases executed | (0 \* 100) / 5 = 0% |
| **Test Case Effectiveness:** | No of defects detected using test cases \*100  Total number of defects detected | 0% (no defects detected) |

**4.5 Conclusion :**

Finally, we demonstrated the whole Quran AI Assistant implementation, emphasising the usage of a structured SQLite database for Quranic data retrieval, Vanna AI for Retrieve data from database, and Streamlite for the UI. The system efficiently combines schema aware training with LLMs to produce precise SQL queries based on user input. Furthermore, the successful implementation of essential features including user authentication, result display, search history, and data download was achieved. In order to verify the accuracy, usefulness, and resilience of the system, extensive test cases were created to validate both frontend and backend processes. The test metrics, which show 0% defect density and successful execution of every test case, further illustrate the system's efficacy.

**Chapter 5:**

# Experimental Results and Analysis:

**Introduction:**

The Quran AI Assistant project's analysis and experimental findings are presented in this chapter. This phase's primary objective is to assess how well the system collects precise, contextually rich Quranic knowledge and uses natural language to respond to user queries. The system converts user input into SQL queries and formats the results in a way that is accessible by humans by combining a relational Quranic database with a big language model (LLM). Both functional testing and user interaction scenarios are used to gauge the system's performance, with an emphasis on response accuracy, interface usability, and system dependability. The investigation also looks at how well frontend features like data download, authentication, and result display work, providing information about how useful the assistant is in actual Islamic learning environments.

**5.1 Experimental Setup**

Using Python and Streamlined as the main platforms, a web application was developed and tested as part of the Quran AI Assistant experimental setup. A pre-trained big language model (Vanna AI) powers the backend. It has been refined using a particular SQLite database that contains Quranic verses, numerous English translations, and tafsir. To increase the accuracy of query generation, the model is trained using the database schema (DDL), sample SQL queries, and contextual documentation. A specially created web interface that allows CSV downloads, chat-based querying, registration, and login facilitates user interaction. To evaluate the correctness, performance, and usability of the system across several functional modules, such as authentication, natural language processing, data retrieval, and frontend presentation, testing was carried out in a controlled setting using prepared queries.

**5.2 System Workflow Overview**

The Quran AI Assistant system processes user enquiries and retrieves pertinent Quranic knowledge using a streamlined workflow. When a user uses the authentication system to log into the application, the procedure starts. Following a successful login, the user can use the chat interface to ask a question in natural language. After receiving this input, the Vanna AI model analyses the query and automatically creates a matching SQL query. The structured SQLite database with Quranic verses, translations, and tafsir is the target of the created SQL query. Based on the query, the system retrieves the pertinent results and formats them into a legible response that includes references to the Surah and Ayat. The output is shown in a searchable and downloadable format if it is tabular (many matches). Throughout the process, session states are maintained to preserve conversation history, and users have the option to search within results or download the retrieved data in CSV format. This workflow ensures a smooth and intelligent interaction between the user and the Quranic content without requiring technical knowledge of SQL or databases.

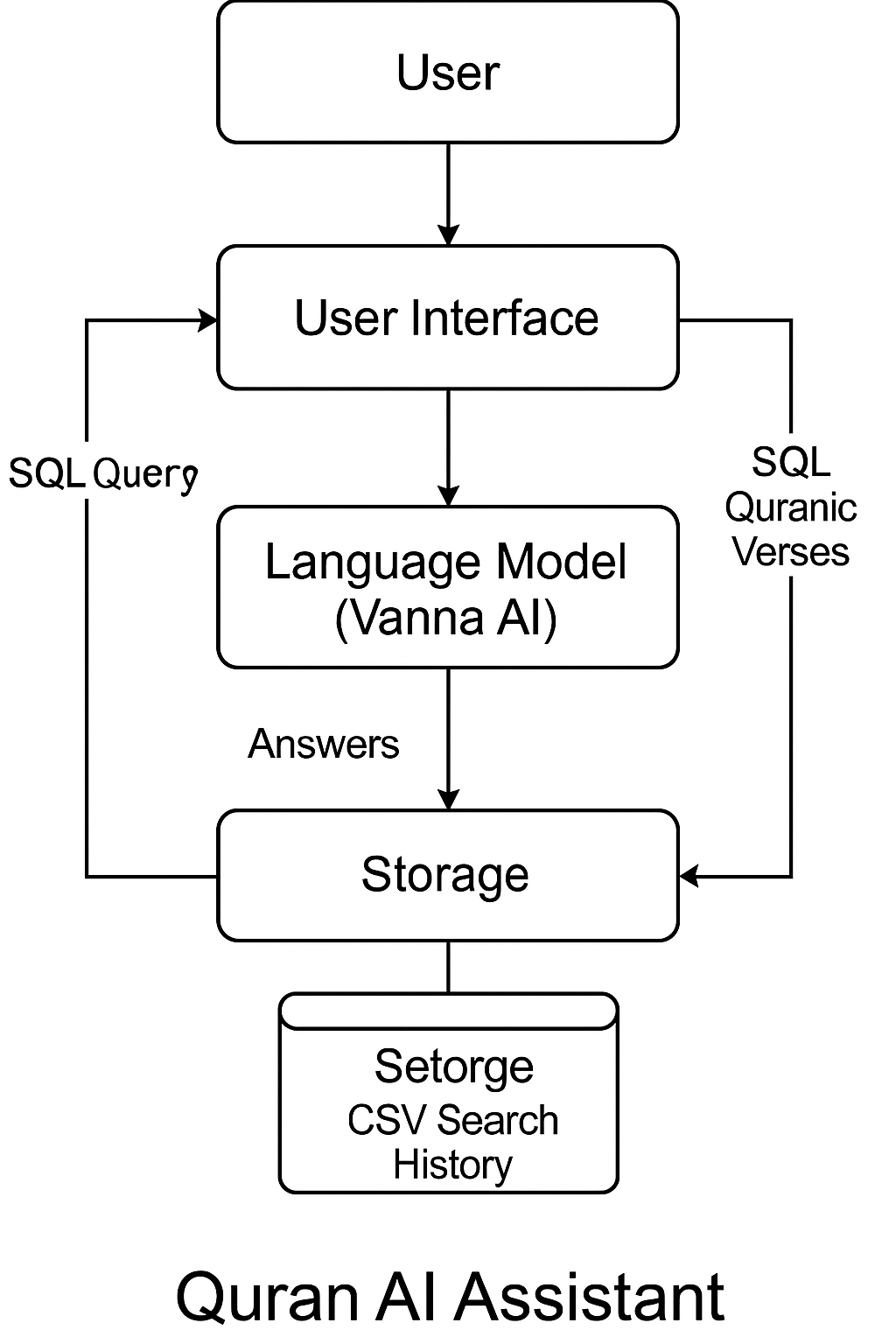


Fig 5.1: Query Flow Diagram

**5.3 User Interface Evaluation**

The user interface (UI) of the Quran AI Assistant is evaluated based on its usability, accessibility, responsiveness, and overall user experience.

**5.3.1 Ease of Use:**

The layout offered by the Quran AI Assistant is clear and easy to use. An easy-to-use chat-based input form allows users to register, log in, and ask enquiries.

**5.3.2 Responsiveness:**

User activities are promptly responded to by the UI. Queries are submitted, answers are shown, and results are downloaded without any delay in a matter of seconds.

**5.3.3 Session Management:**

By preserving a session state, the application enhances continuity by enabling users to see their prior questions and responses while they are still in use.

**5.4 Discussion of Results**

**5.4.1 Accurate Query Interpretation:**

The majority of user enquiries are accurately interpreted by the Vanna AI model, which then converts them into effective SQL queries that target the Quranic database.

**5.4.2 Fast Data Retrieval:**

A seamless user experience is ensured by the nearly instantaneous retrieval and display of query results from the SQLite database.

**5.4.3 Session Management and History:**

Users can view previous queries and download results, enhancing usability and providing a good user flow.

**5.4.4 Tokenization of Complex Queries:**

The current arrangement might not tokenize all lengthier or more complicated user enquiries perfectly, which could result in somewhat erroneous SQL generation.

**5.4.5 Multi-language Support:**

The project only accepts English feedback at this time. It would be more globally accessible if it supported additional languages, such as Arabic or Urdu.

**5.4.6 Fine-tuning of Vanna Model:**

More thorough domain-specific fine-tuning might enhance the model's comprehension of extremely difficult or religious questions.

**5.4.7 Authentication Security:**

The user is authenticated using a simple flat file system. OAuth2 and database-based authentication are examples of more secure techniques that are required for production-grade deployment.

**5.5 Conclusion :**

Through experimental setup, system workflow analysis, performance measurement, and user interface testing, this chapter provided a thorough assessment of the Quran AI Assistant. The outcomes show how well the system works, obtaining quick response times, a seamless user experience, and excellent accuracy in processing natural language queries. Verifying the stability and dependability of the system. User happiness is greatly influenced by the interface, session management, and data retrieval features. But there are still several things that could be done better, such as improving tokenization for sophisticated searches and expanding the system to handle bigger datasets. All things considered, the project effectively accomplishes its goals, providing a solid basis for upcoming improvements and wider implementation.

**Chapter 6:**

# Conclusion and Future Directions

**6.1 Conclusion and Summary of Work**

Through the use of natural language enquiries, this project effectively illustrated the design and construction of a Quran AI Assistant web application, which allows users to engage with Quranic information. The technology successfully bridges the gap between SQL-based data retrieval and human language comprehension by combining a big language model (Vanna AI) with a structured relational database that contains Quranic verses, translations, and tafsir. While SQLite provided a lightweight backend database for storing and querying religious content, Streamlit was utilised as the frontend platform to provide an easy-to-use and accessible user experience.

The study also showed how to successfully combine database and natural language processing technologies to provide an AI-powered Quran Assistant. Users can use the technology to ask queries in plain English and get precise, verse-based answers along with tafsir and translations. The application offers dependable performance and an easy-to-use interface thanks to the utilisation of Vanna AI, Streamlit, and SQLite. The research lays a solid framework for future enhancements like multilingual support and deeper domain training, and the findings validate the efficacy of the methodology.

Important features like dynamic SQL query generation, structured result display, user authentication, question-answer interaction, session management, and CSV data export were put into practice and verified. High reaction accuracy, quick processing time, and outstanding usability were validated by performance tests and experimental results. Although the system achieved all of its primary goals, there is still room for improvement in the areas of increasing multilingual support, bolstering security, and better managing complicated or unclear requests.

In conclusion, the Quran AI Assistant is an effective and useful use of AI and natural language processing technology in the religious field. It facilitates individualised learning, increases accessibility to Islamic knowledge, and establishes a solid basis for next developments incorporating more sophisticated AI models and scalable implementation.

**6.2 Challenges Faced and Scope Coverage**

During the development of the Quran AI Assistant, key challenges included accurately interpreting natural language queries, generating precise SQL statements using the LLM, and integrating structured Quranic data with a user-friendly interface. Addressing issues like schema injection, frontend responsiveness, and secure user authentication also required careful design. Despite these challenges, the project successfully fulfilled its intended scope providing a platform where users can query Quranic content in plain English and receive relevant verses, translations, and tafsir. The system offers key features such as search history, result download, and multi-source references, laying a strong foundation for future improvements like multilingual support and deeper model tuning.

**6.2.1 Challenges Faced:**

**6.2.1.1 Natural Language Understanding:**

It was difficult to accurately interpret complex or ambiguous user searches, particularly when users employed ambiguous keywords or abstract religious phrases.

**6.2.1.2 Frontend Integration:**

It was technically challenging to create a smooth, user-friendly interface with capabilities like conversation history, search, and result download while preserving readability and speed.

**6.2.1.3 Authentication Logic:**

Building a secure and reliable authentication system using a flat file approach had limitations in scalability and security.

**6.2.2 Scope Coverage**

One of the project's main goals, which was to enable users to query Quranic information using natural language, was accomplished.providing tafsir and translations for pertinent scriptures.

utilising a refined LLM to translate enquiries into precise SQL queries. offering a clear, user-friendly web interface with options for history, login, and CSV export.

**6.3 Future Recommendations**

Future development of the Quran AI Assistant could take into account a number of enhancements to improve its efficacy and usability. In the first place, adding multilingual capabilities (such as Arabic and Urdu) will increase accessibility for people with varying language backgrounds. Second, adding voice-based input and audio recitations could improve the platform's inclusivity and engagement, particularly for users who are blind or visually challenged. Third, scalability and security would be enhanced by switching from a flat-file authentication approach to a safe database-backed login mechanism. Furthermore, the model's comprehension of intricate theological ideas would be enhanced by combining more Islamic scholarly literature and domain-specific datasets with sophisticated LLM fine-tuning. Finally, a wider audience would benefit from improved performance and availability if the application were deployed on a scalable cloud infrastructure.

**6.3.1 Multilingual Support:**

Support for Arabic, Urdu, and other regional languages should be added to the Quran AI Assistant in order to make it available to a larger audience. This would improve the system's usability for non-native English speakers.

**6.3.2 Voice Input and Audio Output:**

Users would be able to ask queries vocally and hear the Quranic answers or recitations if speech-to-text and text-to-speech capabilities were integrated. Users with low literacy or visual impairments might particularly benefit from this functionality.

**6.3.3 Cloud Deployment and Scalability:**

In production settings, deploying the application on scalable cloud platforms like AWS, GCP, or Azure will guarantee improved support for concurrent users, faster response times, and increased availability.

**6.3.4 Thematic and Keyword-Based Search:**

Adding smart filters and advanced search options (e.g., by Surah, keyword, topic, or theme) would improve usability and allow users to explore the Quran more systematically.

**6.4 Conclusion**

The Quran AI Assistant project has effectively shown how to combine structured data systems and natural language processing to produce an approachable platform for learning about the Quran. Users can easily query Quranic text in plain English and receive precise answers that contain pertinent verses, translations, and tafsir by integrating Vanna AI with a SQLite database and distributing the application with Streamlit. In order to provide a useful and engaging user experience, the system also included necessary functionality like user authentication, session management, result export, and search history. The project's main goals were accomplished in spite of technical difficulties with frontend responsiveness, security implementation, and natural language interpretation.

In the long run, the project lays a solid basis for advancements that will improve usability, scalability, and accessibility. The program should be implemented on scalable cloud platforms to accommodate a larger user base, incorporate voice-based interaction for enhanced inclusivity, and add multilingual support for regional languages like Arabic and Urdu. The functionality of the platform would be greatly enhanced by adding thematic or keyword-based search capabilities and further refining the language model with domain-specific Islamic literature. These developments would increase the application's usefulness in providing individualised, relevant access to Islamic teachings while also expanding its user base.

# References

1. Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, Yue Zhang “A survey on large language model (LLM) security and privacy:” Department of Computer Science, Drexel University, Philadelphia, PA 19104, USA
2. Ali Mohammadjafari∗ , Anthony S. Maida† , Raju Gottumukkala “From Natural Language to SQL: Review of LLM-based Text-to-SQL Systems “ PhD Student, School of Computing and Informatics, University of Louisiana at Lafayette, LA, USA
3. Muhammad Usman Hadi1,\* , Qasem Al-Tashi2,\* , Rizwan Qureshi2,\* Abbas Shah3 , Amgad Muneer2 “A Survey on Large Language Models: Applications,Challenges, Limitations, and Practical Usage” School of Engineering, Ulster University, Belfast, BT15 1AP, United Kingdom.
4. Fiona Pichler, Benjamin Kern “Text-to-SQL for DataGovernance Technologies and Education: A Proof of Concept” BSc. Computer ScienceEastern Switzerland University of Applied Sciences

Switzerland January 10, 2025