**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**

The **ChatWithQuran** project is a web-based platform designed to provide users with a comprehensive, accessible, and interactive experience in exploring Quranic text. While current digital platforms and AI-based Islamic chatbots offer basic search capabilities and short responses, they often lack in-depth interpretations and complete responses to complex queries. This project aims to address these gaps by incorporating advanced Natural Language Processing (NLP) techniques to deliver precise, contextually relevant, and detailed answers. By utilizing keyword-based and semantic search functionalities, **ChatWithQuran** will enable users to retrieve specific Quranic verses, translations, and extensive tafseer (interpretations) based on their queries. Through this approach, **ChatWithQuran** seeks to create a more engaging and educational experience, empowering users to gain a deeper understanding of the Quranic teachings.

The **ChatWithQuran** platform stands out by addressing specific limitations observed in existing Quranic search and AI-based Islamic chatbot solutions. Many current platforms provide limited or incomplete tafseer, lack detailed responses, or fail to understand complex, context-rich user queries. By integrating sophisticated NLP models, **ChatWithQuran** will enhance user experience with both simple keyword matching and reinforcement learning, allowing users to access Quranic content that is directly relevant to their questions. Additionally, the platform’s ability to deliver comprehensive, complete responses with full interpretations aims to bridge the knowledge gap, making Quranic teachings accessible to a broader audience. This project not only promises to improve accessibility to religious texts but also represents an innovative application of AI in the domain of Islamic scholarship.

Table of Contents

[Table of Contents i](#_gjdgxs)

[List of Tables iii](#_30j0zll)

[List of Figures iv](#_3znysh7)

[Abstract**.**](#_4d34og8)

[Chapter 1: Introduction](#_2s8eyo1) 1

[1.1 Goals and Objectives 3](#_17dp8vu)

[1.2 Scope of the Project 4](#_3rdcrjn)

[Chapter 2: Literature Review 5](#_26in1rg)

[2.1 Introduction 5](#_lnxbz9)

[2.2 Background and Problem Elaboration 5](#_35nkun2)

[2.3 Detailed Literature Review 7](#_1ksv4uv)

[2.3.1 Definitions 7](#_44sinio)

[2.3.2 Related Research Work 1 7](#_2jxsxqh)

[2.3.3 Related Research Work 2 9](#_z337ya)

[2.3.4 Related Research Work 3 10](#_z337ya)

[2.4 Literature Review Summary Table 12](#_3j2qqm3)

[2.5 Research Gap 13](#_1y810tw)

[2.6 Problem Statement 14](#_4i7ojhp)

[2.7 Conclusion 15](#_4i7ojhp)

[Chapter 3: Requirements and Design 16](#_2xcytpi)

[3.1 Requirements 17](#_1ci93xb)

[3.1.1 Functional Requirements 17](#_3whwml4)

[3.1.2 Non-Functional Requirements 17](#_2bn6wsx)

[3.1.3 Hardware and Software Requirements 18](#_qsh70q)

[3.2 Proposed Methodology 21](#_3as4poj)

[3.3 System Architecture 23](#_1pxezwc)

[3.4 Use Cases 24](#_49x2ik5)

[3.4.1 Sample Use Case Name Here](#_2p2csry) **.**

[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 27](#_1hmsyys)

[4.1 Implementation 27](#_41mghml)

[4.1.1 Implementation of First Component/Algorithm 27](#_2grqrue)

[4.2 Test case Design and description 27](#_vx1227)

[4.2.1 Sample Test case No.1 27](#_3fwokq0)

[4.2.2 Sample Test case No.2 27](#_1v1yuxt)

[4.3 Test Metrics 28](#_4f1mdlm)

[4.3.1 Sample Test case Matric.No.1 28](#_2u6wntf)

[4.3.2 Sample Test case Metric.No.2 28](#_19c6y18)

[4.3.3 Sample Test case Metric.No.3 28](#_3tbugp1)

[Chapter 5: Experimental Results and Analysis 29](#_28h4qwu)

[Chapter 6: Conclusion and Future Directions 30](#_nmf14n)

[References 31](#_37m2jsg)

[Appendix 32](#_1mrcu09)

[Appendix A: Guidelines 32](#_46r0co2)

[Appendix B: Heading of Sample Appendix B 32](#_2lwamvv)

[Formatting Guidelines 33](#_111kx3o)

[Chapter 1: Heading 1 33](#_3l18frh)

[1.1 Heading 2 33](#_206ipza)

[1.1.1 Heading 3 33](#_4k668n3)

[Tables and Figures 34](#_2zbgiuw)

[Equations 35](#_1egqt2p)

[Header/Footer 35](#_3ygebqi)

[Other Formatting Guidelines 35](#_2dlolyb)

[References 35](#_sqyw64)

List of Tables

[Table 1: This is Sample table caption 34](#_3cqmetx)

[Table 2: This is Sample table caption 34](#_1rvwp1q)

List of Figures

Figure 1: List of Styles 34

Figure 2: IEEE Reference style 35

**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.

Millions of people around the world rely on the Quran as their primary source of wisdom and direction. But it can be difficult to find certain verses, comprehend what they represent, and investigate the related tafseer (interpretation), especially for people who are unfamiliar with the Quran. Our project, ChatWithQuran, aims to create a web-based platform that will allow users to easily query Quranic verses and obtain precise translations and tafseer using an intuitive search interface. The platform's integration of sophisticated Natural Language Processing (NLP) algorithms will improve the search experience by enabling users to locate pertinent verses even when their queries are imprecise or lacking. This project seeks to increase the Quran's readability and accessibility in order to promote a closer relationship and comprehension of the book.

**1.1 Goals and Objectives:**

Enhance Accessibility of Quranic Knowledge to Provide a user-friendly platform that makes Quranic verses, translations, and tafseer available to a large audience. It provides Accurate and Context-Rich Responses to Make sure users receive thorough, accurate, and pertinent responses to their questions by utilizing sophisticated natural language processing techniques. Bridge Gaps in Existing Islamic Chatbot Platforms: Provide thorough, reputable, and in-depth tafseer for a deeper comprehension in order to overcome the shortcomings of the existing Quranic chatbots.

**1.1.1 Objectives:**

Analyze and interpret user inquiries by utilizing Natural Language Processing (NLP) techniques.

Ensure 100% accuracy by training a specialized large language model (LLM) to map

queries to a structured database.

To increase answer quality and relevance, NLP algorithms should be continuously improved and tested.

Provide translations and tafsir from two renowned scholars to support diverse user

needs and ensure credibility.

Assure effective Quranic verses, translations, and tafseer processing, retrieval, and storage from the dataset.

Create an interface that is easy to use, accessible, and intuitive so that users can enter queries and view results with ease.

Work along with the NLP and backend teams to guarantee seamless operation, accurate result display, and general user happiness.

1.2 Scope of the Project

Provide a well-structured database with Quranic verses, translations, and comprehensive tafseer so that users can get precise Islamic information for every verse. Employ cutting-edge natural language processing (NLP) approaches to precisely analyze user searches, such as keyword extraction and thematic understanding, to deliver pertinent verse, translation, and tafseer results. Employ thematic search functions (such as LLM) to enable users to locate verses and tafseer that are contextually relevant to their queries, even in the absence of specific keywords. Give users choices for both in-depth tafseer and flexible answers according to user preferences and the difficulty of the query. Design an intuitive web interface that offers simple, clear navigation, allowing users to search, view, and interact with Quranic content effortlessly. To preserve clarity and comprehension depth, make sure that answers to user inquiries are comprehensive, contextually correct, and consistent across question types.

**Chapter 2: Literature Review**

**2.1 Introduction**

Processing methods (NLP). Through the website, users can submit queries or keywords and get precise, contextually relevant answers, such as translations, Quranic verses, and tafseer (interpretations). By using a vast collection of Quranic knowledge, ChatWithQuran seeks to deliver thorough and extensive replies, in contrast to many other solutions that only provide restricted and frequently insufficient answers. The platform's capabilities, which include keyword matching, semantic search, and summarized tafseer alternatives, are designed to be a dependable resource for anybody looking to get a greater understanding of the Quran. 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.

ChatWithQuran hopes 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, comprehensive responses, the ChatWithQuran project seeks to remove these constraints and meet the demands of users who want to comprehend the Quran.

2.2 Background and Problem Elaboration

By bridging the gap between traditional Quranic study and modern digital accessibility with an intuitive design and multilingual support, ChatWithQuran aims to enhance the user experience for individuals worldwide. The ChatWithQuran initiative aims to eliminate these limitations and satisfy the needs of users who wish to get a deeper understanding of the Quran by providing a platform that offers comprehensive tafseer and contextually rich, in-depth responses.   
It can be difficult to explain tafseer in a way that respects and clarifies the various ways that different schools of thought read the same verses in the Quran. A user-friendly, interactive platform that can adjust to each learner's unique needs and offer tailored advice via Quranic content is required. The use of conversational AI tools for self-study is growing in many domains, but Islamic studies more especially, the interpretation of the Quran has not yet completely profited from these developments. The majority of Quranic systems currently in use utilize simple keyword matching, which restricts their capacity to handle complex queries where the user might not be aware of the precise terms to search for.

**2.3 Detailed Literature Review**

In an effort to improve comprehension and interpretation of religious texts, the application of Natural Language Processing (NLP) to Quranic studies has been increasingly popular in recent years. In order to categorize and evaluate Quranic verses and their translations, this research aims to improve a LLM or BERT-based model that was created especially for multilingual jobs. The study aims to tackle the intricacy of semantic differences across languages and interpretations by utilizing a dataset that comprises Arabic text, several English translations, and tafsir (interpretation). Tokenization, stopword removal, and diacritical removal are examples of preprocessing methods that guarantee the data is ready 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**

Creating an AI-based system to categorize and evaluate Quranic verses, their translations, and tafsir (interpretations) is the main goal of ChatWithQuran. It seeks to offer precise classification and more profound insights into Quranic texts for improved comprehension and accessibility by utilizing cutting-edge machine learning techniques such as multilingual BERT or LLM.

Using cutting-edge natural language processing techniques, ChatWithQuran builds an AI-powered model to categorize and evaluate Quranic verses and their translations. The system seeks to facilitate user exploration and efficient classification of Quranic content by training a multilingual LLM or BERT-based model on Quranic text and translations.

**2.3.2 Related Research Work 1**

This web-based Quran platform aims to give users easy access to Quranic texts, translations, and interpretations. The platform's multilingual query capability, which incorporates cutting-edge natural language processing algorithms, allows users to search and study Quranic text in Arabic, English, and other languages. It makes use of cutting-edge machine learning models, such LLM or BERT and its multilingual variations, to improve the accuracy and relevance of search results, guaranteeing that users can quickly locate particular verses or topics. To accommodate a range of user preferences and academic requirements, the site also includes numerous translations and tafsir (interpretations). The platform is a useful resource for scholars, students, and public audiences looking to gain a deeper grasp of the Quran because of its intuitive interface and sophisticated backend.

**2.3.3 Related Research Work 2**

This study offers a thorough investigation of a web-based Quranic platform intended to give users easy access to Quranic texts, interpretations, and translations. To maximize keyword search and guarantee precise retrieval of Quranic text across several languages, the platform uses sophisticated natural language processing (NLP) and machine learning techniques, including multilingual models like BERT or LLM. The system serves a broad audience, including scholars, students, and everyday users, by incorporating several translations and tafsir resources, encouraging a deeper comprehension of the Quran. In order to facilitate effective exploration of themes, keywords, and situations within the Quran, the platform's architecture which blends user-friendly interface with advanced backend processing is highlighted in the study. This study highlights the value of technology in supporting Quranic studies and shows how AI-powered solutions can improve religious text accessibility.

**2.3.4 Related Research Work 3**

This paper presents IslamandAI, a beta-version AI-powered chatbot that can provide succinct and precise answers to common Quranic questions. The chatbot uses cutting-edge natural language processing (NLP) models to interpret user queries or keywords and retrieve pertinent Quranic verses, translations, or explanations to enable interaction. By providing users with a rapid and effective method of learning about the Quran, this early-stage system seeks to increase accessibility to Islamic knowledge. The development is appropriate for consumers looking for a basic comprehension or clarification on frequent Quranic themes since it prioritizes accuracy, simplicity, and usability. The study demonstrates how AI can help people connect with Islamic teachings more deeply while tackling issues like maintaining theological truth and efficiently managing a variety of linguistic inputs.

**2.3.5 Related Research Work 4**

The article highlights the evolution of machine language capabilities, starting from statistical models to the powerful Large Language Models (LLMs) of today. While humans naturally develop language skills, machines require advanced AI algorithms to understand and generate language. The rise of LLMs powered by deep learning, large datasets, and vast computational resources has significantly improved machines' ability to perform complex language tasks.

The paper emphasizes a focus on LLMs, categorizing them into four types and highlighting their role in both NLP and Computer Vision. It also introduces ChatGPT (based on GPT-3.5), known for generating human-like responses, as an example of a cutting-edge application in this domain.

**2.3.6 Related Research Work 5**

This research paper provides a comprehensive overview of the Text-to-SQL task, which aims to bridge the gap between non-technical users and complex relational databases by converting natural language (NL) queries into accurate SQL statements. It highlights the challenges of ambiguity, schema understanding, and cross-domain generalization in current systems. The paper introduces Retrieval-Augmented Generation (RAG) as a promising solution that enhances SQL generation by combining a retrieval module with a generative module to dynamically fetch and integrate relevant schema or knowledge. Additionally, it traces the evolution of text-to-SQL systems from rule-based methods to deep learning and pre-trained language models, showcasing significant progress in understanding and generating SQL from natural queries. The paper concludes with a taxonomy of current techniques and emphasizes RAG and Graph RAG as state-of-the-art approaches to improve accuracy and adaptability in real-world applications.

**2.4 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 accurate responses, including comprehensive 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 or keywords, 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 ground-breaking 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

| **ID** | **Category** | **Requirement**  **Description** |
| --- | --- | --- |
| FR1 | User Query Processing | Allow users to input queries and retrieve relevant Quranic verses, translations, and tafseer. Support basic and semantic 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 multilingual, 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**:

Verify if the platform's Quranic verses, translations, and tafseer are true, genuine, and derived from reliable Islamic scholars and sources.

**3.1.2.2 Performance**:

For the best user experience, search results should load on the platform in two to three seconds.

**3.1.2.3 Scalability**:

A rising user base and an increasing query load should be supported by the system without causing performance issues.

**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**:

Respect data privacy guidelines by making sure that, if accounts are supported, user data is encrypted and maintained safely.

### 3.1.3 Hardware and Software Requirements

The size of the dataset, the intricacy of the model architecture, and the training methodology can all affect the hardware needed to train the model on the dataset. Nonetheless, the following specifications are deemed sufficient for datasets of a moderate size.

**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**:

Nginx or Apache to serve the 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, use Transformers, SpaCy, NLTK, Streamlit, vanna.ai, ollama 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:**

Compile an extensive collection of Quranic passages, translations (such as several English translations), and Tafsir from reliable sources.While maintaining theological accuracy, clean and preprocess the data by eliminating duplicates, unnecessary text, and inconsistencies.

**3.2.2 Text Representation:**

The Quranic text and translations should be tokenized using cutting-edge multilingual tokenizers such as LLM.Using the proper NLP approaches, handle linguistic variety by making sure Arabic, English, and other languages are supported.

**3.2.3 Model Selection and Training:**

Adjust transformer-based 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:**

Create a backend system with an intuitive user interface that incorporates the learned AI model. Give the chatbot the ability to take user input (question or keyword) and provide succinct, precise Quranic answers.

**3.2.5 Evaluation and Optimization:**

To assess the chatbot's functionality, precision, and applicability, test it with a variety of questions. Adjust the model for improved contextual knowledge, handle edge cases, and optimize it based on 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:**

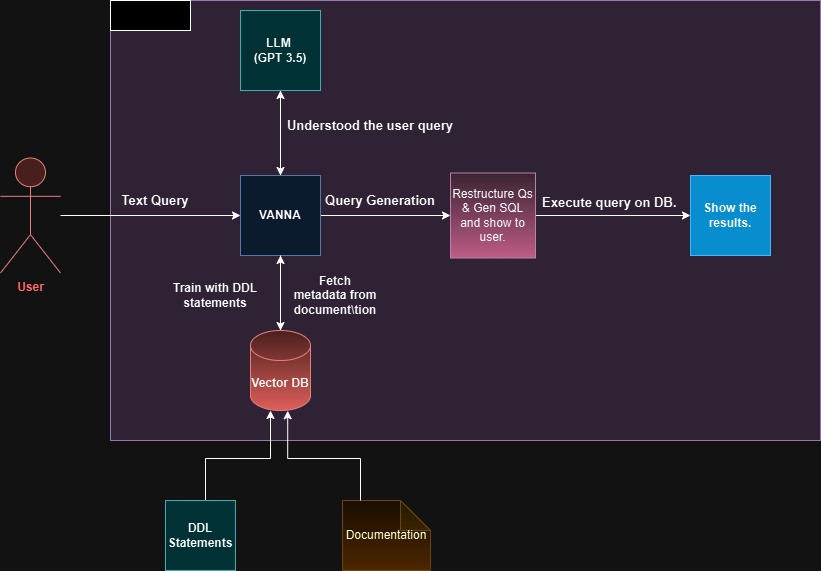
Install the chatbot on an online platform and keep an eye on its functionality in real time.To make the system better, enhance the user experience, and add more functionality over time, get customer input.

**3.3 System Architecture**

The system architecture for the Quranic chatbot is made to guarantee dependability, scalability, and functionality. The major point of engagement is the User Interface Layer, which provides a web-based platform where users may enter queries or keywords and get user-friendly answers. A query processor, intent recognition module, and an AI model optimized for Quranic contexts are used by the Application Layer to process these inputs and produce pertinent results, including translations and verses. The Data Layer uses a knowledge base and database to store and retrieve structured Quranic data, with the help of a dedicated search engine for keyword-based searches.

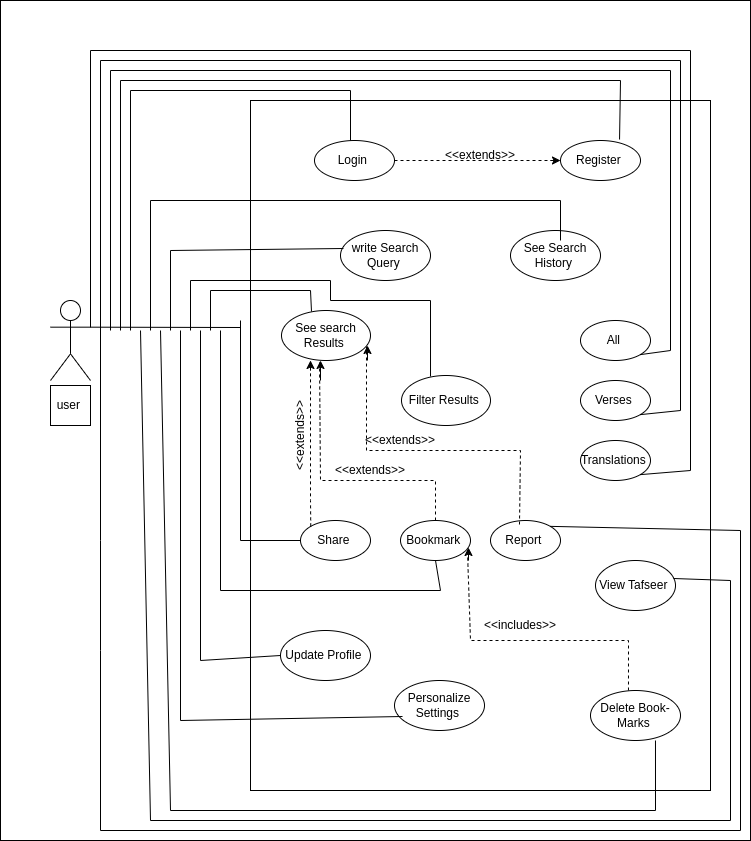
The use of caching techniques, the middleware layer facilitates smooth communication between the application and data levels. To ensure scalability and dependability, the system is delivered using the Deployment Layer, which also includes cloud hosting and monitoring tools. Last but not least, a User Feedback and Monitoring Layer gathers user input and monitors system performance to support ongoing development and guarantee the chatbot successfully satisfies user needs.

**3.3.1 Architecture Diagram**



* 1. **Use Cases**

**3.4.1 Use Case Diagram (User)**

****

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

| UC ID | UC-U03 |
| --- | --- |
| UC Name | Write 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 Search Query in the Search Bar 3. Their previous Search history will be available just below the Search Bar 4. After writing the Search Query, they will press the search icon (or press 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)**

| UC ID | UC-U02 |
| --- | --- |
| UC Name | See Search 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 may or may not contain all the mentioned type of items  2.2 The results can have any mixture of items that are displayed  2.3 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)**

| UC ID | UC-U03 |
| --- | --- |
| UC Name | See Search History |
| Primary Actor | User |
| Pre-Condition | User should have clicked on the Search Bar/ or they should have opene Search History Page |
| Post-Condition | User will have successfully manipulated the Search History |
| Main Flow | 1. When User clicks on the Search Bar, they will be shown with their recent search history 2. If the user clicks on an item in the search history, it will appear in the Search Bar, and User can search it 3. If the User long presses the item, they will be prompted with a message “This Item will be Deleted from History” 4. If user clicks “Delete”, the item will be removed from their history |
| Alternate Flow | 1.1 The User can also go to the Profile Page to see their Search History  1.2 User will click on the “See History” button to open the Search History Page  1.3. The User will be shown their Search History in a chronological manner  1.4 If the User clicks on item, it will be searched, taking the user to its particular results  1.5 The user can also click on “Delete All” to clear their Search History |

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

| UC ID | UC-U04 |
| --- | --- |
| UC Name | Filter Search 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 hadiths, 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  6.1 The origins search query results will be stored in the Phone Cache  6.2 Internet Connectivity will not be an issue |

**3.4.1.5 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, Hadiths, 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)**

| 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)**

| 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. User will click on the Hadith option from the filter values 3. All the results, that are hadiths, will be shown 4. Only the hadiths 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 Hadith 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 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)**

| UC ID | UC-U10 |
| --- | --- |
| UC Name | Share |
| Primary Actor | User |
| Pre-Condition | User should have successfully opened an item |
| Post-Condition | User will have successfully shared the opened item's content |
| Main Flow | 1. User will click on the share option that will be available below the original text 2. Users can share any item (Verse, Surah, Tafseer, etc.) 3. When they click on the share 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 shared 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 |

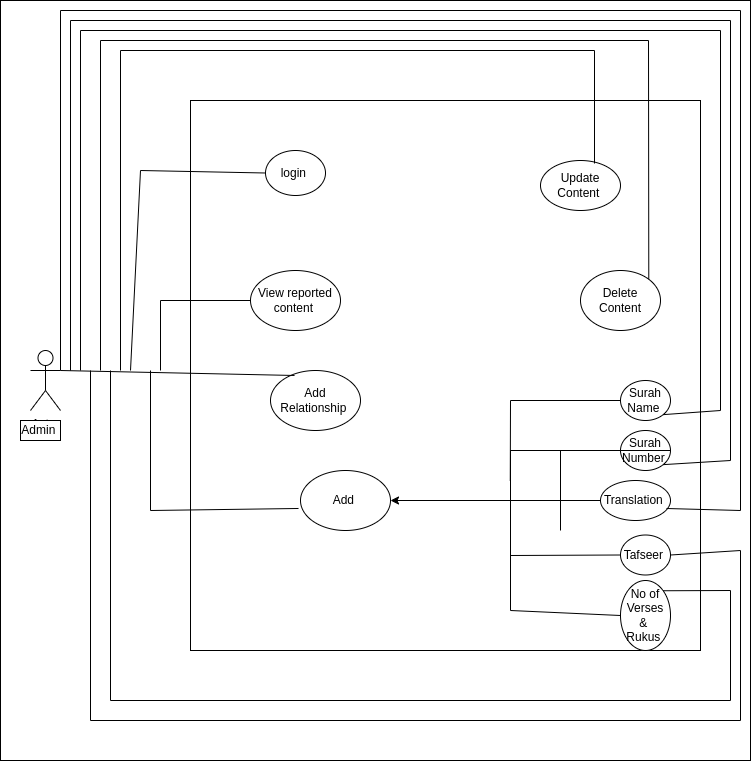
**3.4.1.9 Use Case Description (UC-U09)**

| 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)**

| 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  5.1 Once User presses “Delete All”, they will be prompted if they are sure they want to Delete All Bookmarks  5.2 If User presses Ok, all their bookmarks will be deleted. This action cannot be undone |

**3.4.2 Use Case Diagram (Admin)**

****

**Chapter 4:** **Implementation and Test Cases**

**4.1 Introduction**

The primary objective of this chapter is to create a full stack web application which helps the user understand and access data which lies in a relational database. LLMs already assist in generating SQL queries; they are however often faulty because they lack the information of the database schema. The challenge lies hereby in ﬁnding ways of injecting the database schema knowledge of a speciﬁc database into the LLM such that the LLM can give more accurate, as well as more complex assistance than without knowledge of the database schema. This project aims to resolve this problem by exploring various possibilities of injecting said domain knowledge such as fine-tuning an LLM, few-shot learning and other paradigms. Finally, a proof of concept must be implemented as a full stack web application.

**4.2 Implementation**

The implementation of this project is centered around building an efficient system capable of understanding the user’s query and extracting relevant data. Below, we detail the major components of the system, the algorithms used, the technologies implemented, and the overall architecture of the system.

**4.2.1 Implementation Details:**

**4.2.1.1 Platform:**

The platform used for this project is **Streamlit**, a popular open-source Python library for building interactive web applications especially for data science, machine learning, and AI-powered tools.

**4.2.1.2 Functionality:**

Streamlit-based Quran AI Assistant project is to provide users with an interactive, intelligent platform to ask questions about the Quran and receive contextually rich, referenced answers.

**4.2.2 Model Training:**

The app uses and fine-tunes a pre-built model (Vanna AI) by training it on domain specific data related to the Quran.

Vanna is an LLM powered tool designed to generate SQL from natural language questions.

**4.2.4 Text Extraction**

To extract text from the dataset in your Quran AI Assistant project, you’re working with a SQLite database (quran translation). This dataset includes Quranic verses, translations, and tafsir stored in a table named quran translation.

**4.2.5 Text to SQL queries:**

In this project, when a user types a question in natural language like “What does the Quran say about forgiveness?” The system uses an AI model called Vanna to understand the meaning of the question. Vanna then automatically creates an SQL query that searches the Quranic database for relevant verses, translations, and tafsir.

**4.2.5.1 Implementation Details:**

**4.2.5.2 Platform:**

The main platform used to build the web application. It creates the user interface (chat input, display, authentication, etc.).

**4.2.5.3 Libraries :**

**Vanna AI** converts the question to SQL.  
**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 handles user registration, login, and session management using a simple flat file system.

**4.2.6.2 Result Search :**

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

**4.2.6.3 Search History :**

Users can see the search history which was searched before.

**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**

In this project, test cases were designed to validate both the frontend and backend functionalities of the Quran AI Assistant, including user authentication, natural language processing, database querying, and user interface behavior. Each test case outlines a specific scenario with defined inputs, expected outputs, and the corresponding module responsible.

**4.3.1 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**

| 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**

| 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**

| 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**

The Coverage and effectiveness of our system is shown below.

### 4.4.1 Sample Test case Metrics

| **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 :**

In conclusion we presented the complete implementation of the Quran AI Assistant, highlighting the use of Streamlit for the frontend, Vanna AI for natural language to SQL conversion, and a structured SQLite database for Quranic data retrieval. The system effectively integrates LLMs with schema-aware training to generate accurate SQL queries from user input. Additionally, core functionalities such as user authentication, result display, search history, and data download were successfully implemented. Comprehensive test cases were designed to validate both frontend and backend operations, confirming the system's accuracy, usability, and robustness. The test metrics further demonstrate the system's effectiveness, with all test cases executed successfully and zero defect density.

**Chapter 5:**

# Experimental Results and Analysis:

**Introduction:**

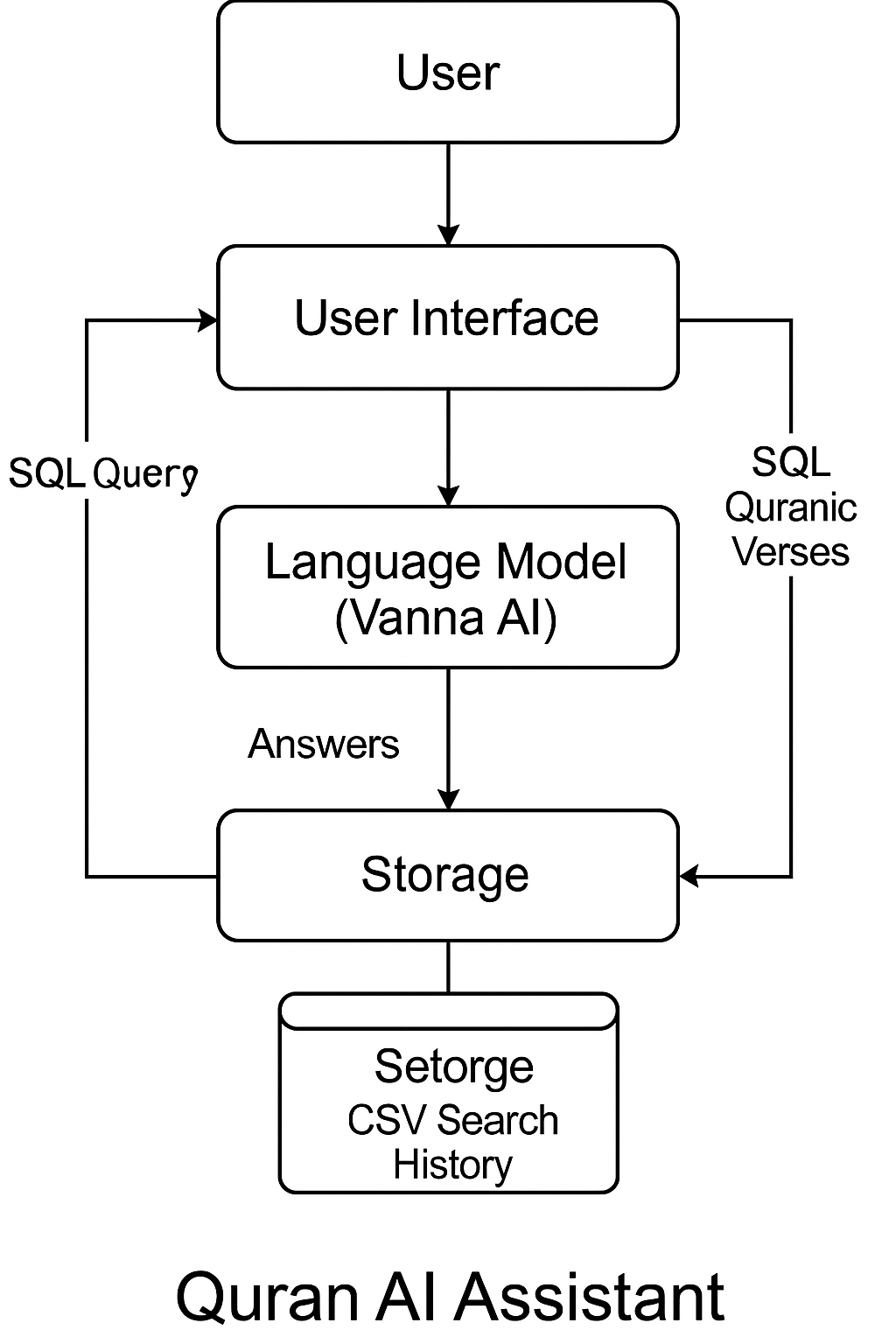
This chapter presents the experimental results and analysis of the Quran AI Assistant project. The main goal of this phase is to evaluate how effectively the system responds to user queries using natural language and retrieves accurate, context-rich Quranic information. By integrating a large language model (Vanna AI) with a relational Quranic database, the system translates user input into SQL queries and formats results in a human-readable manner. The performance of the system is measured through both functional testing and user interaction scenarios, focusing on the accuracy of responses, interface usability, and system reliability. The analysis also examines the effectiveness of frontend components such as result display, authentication, and data download, offering insights into the practical applicability of the assistant in real-world Islamic learning contexts.

**5.1 Experimental Setup**

The experimental setup for the Quran AI Assistant involved the development and testing of a web application using Python and Streamlit as the primary platform. The backend is powered by a pre-trained large language model (Vanna AI), which is fine-tuned using a specific SQLite database containing Quranic verses, multiple English translations, and tafsir. The model is trained with the database schema (DDL), sample SQL queries, and contextual documentation to improve query generation accuracy. User interaction is facilitated through a custom designed web interface that supports login, registration, chat-based querying, and CSV downloads. Testing was performed in a controlled environment using predefined queries to assess the system’s accuracy, performance, and usability across multiple functional modules including authentication, natural language processing, data retrieval, and frontend display.

**5.2 System Workflow Overview**

The Quran AI Assistant system follows a streamlined workflow to process user queries and retrieve relevant Quranic information. The workflow begins when a user logs into the application using the authentication system. After successful login, the user can enter a question in natural language through the chat interface. This input is then passed to the Vanna AI model, which interprets the question and automatically generates a corresponding SQL query. The generated SQL query is executed against the structured SQLite database containing Quranic verses, translations, and tafsir. The system retrieves the relevant results based on the query and formats them into a readable response, including Surah and Ayat references. If the output is tabular (multiple matches), it is displayed in a searchable and downloadable grid format using Streamlit components. 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.



**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 Quran AI Assistant provides a clean and intuitive layout. Users can easily register, log in, and submit questions through a simple chat-based input form.

**5.3.2 Responsiveness:**

The interface responds quickly to user actions. Submitting queries, displaying answers, and downloading results happen within a few seconds without lag.

**5.3.3 Session Management:**

The app maintains a session state, allowing users to view their previous queries and answers during their active session, improving continuity.

**5.4 Discussion of Results**

**5.4.1 Accurate Query Interpretation:**

The Vanna AI model correctly interprets most user questions and translates them into efficient SQL queries targeting the Quranic database.

**5.4.2 Fast Data Retrieval:**

Query results from the SQLite database are fetched and displayed almost instantly, ensuring a smooth user experience.

**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:**

Some longer or more complex user questions may not be tokenized perfectly by the current setup, potentially causing slightly inaccurate SQL generation.

**5.4.5 Multi-language Support:**

Currently, the project handles English input only. Supporting other languages like Arabic or Urdu would make it more globally accessible.

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

More extensive domain-specific fine-tuning could further improve the model’s ability to understand highly theological or complex queries.

**5.4.7 Authentication Security:**

User authentication is handled by a basic flat file system. For production grade deployment, more secure methods (like OAuth2 or database-based authentication) are needed.

**5.5 Conclusion :**

This chapter presented a detailed evaluation of the Quran AI Assistant through experimental setup, system workflow analysis, performance measurement, and user interface testing. The results demonstrate that the system performs effectively, achieving high accuracy in natural language query processing, fast response times, and a smooth user experience. Confirming the system’s stability and reliability. The interface, session management, and data retrieval features contribute significantly to user satisfaction. However, areas for improvement were also identified, including enhancing tokenization for complex queries, and scaling the system for larger datasets. Overall, the project successfully meets its objectives, laying a strong foundation for future enhancements and broader deployment.

**Chapter 6:**

# Conclusion and Future Directions

**6.1 Conclusion and Summary of Work**