An-Najah National University

Department of Computer Engineering

Graduation Project 1

Zikr: An AI-Driven Islamic Social Platform

Mohammad Raed

Mohammad Matar

Supervisor: Eng. Haya Samaaneh

A report submitted in partial fulfilment of the requirements of
An-Najah National University for the degree of
Bachelor of Science

in *Computer Engineering*

June 15, 2025

Abstract

This project presents **Zikr** an innovative platform designed to support Muslims worldwide in their religious journey by combining social community features with advanced Artificial Intelligence. Unlike traditional Islamic apps that focus on a single functionality, Zikr integrates a vibrant social media environment—enabling users to create posts, comment, like, and interact—while also providing AI-powered tools for religious education and engagement.

The platform's core AI components include: (1) a Retrieval-Augmented Generation (RAG) chatbot that answers Islamic Fatawa questions in Arabic and English, leveraging a dataset scraped from the IslamQA website to ensure accurate, authentic responses; and (2) a K-Means clustering model that highlights Quranic topics with distinct colors, enhancing the recitation experience. Additionally, Zikr features a personalized content suggestion algorithm for the social feed, utilizing a fuzzy logic system to recommend posts based on user interests and activity, thus fostering meaningful engagement and community building.

The methodology involved two main tracks: application development and AI integration. The application was built using a Flutter frontend and a Django backend, with features such as user authentication, profile management, and a robust social module. The AI track included dataset collection and preprocessing, model selection and training, and seamless integration with the platform. The suggestion algorithm was designed to balance personalization with content diversity, ensuring users are exposed to both relevant and varied topics.

The main challenges addressed were the scarcity of high-quality Islamic datasets, the complexity of Quranic language, and the need for a safe, distraction-free social space. By overcoming these, Zikr offers a unique, holistic solution that advances the use of AI in Islamic education and community interaction. The project demonstrates the potential of combining modern AI techniques with user-centric design to create impactful digital tools for the Muslim community.

Keywords: Islamic AI, Social Community, Quran Topic Modeling, Fatawa Chatbot, Fuzzy Suggestion Algorithm, RAG, Arabic NLP, Flutter, Django, K-Means Clustering

The full project code is available in two GitHub repositories:

- Main application:
  https://github.com/mo-matar/Zikr-AI-Driven_Social-Platform.git
- K-means clustering model:
  https://github.com/mo-matar/K-means-Topic-Modeling-of-Quran.git

Acknowledgements

We extend our deepest gratitude to our supervisor, Eng. Haya Samaaneh, for her unwavering support and insightful critiques throughout our project journey. Her deep commitment to academic excellence and meticulous attention to detail have significantly shaped this dissertation.

Our appreciation also goes to the faculty and staff in the Department of Computer Engineering at An-Najah National University, whose resources and assistance have been invaluable. We would also like to acknowledge our peers, especially Momen Anani, for their camaraderie and the stimulating discussions that inspired us throughout our academic journey. Their collective wisdom and encouragement have been a cornerstone of our project experience.

Contents

List of Figures vi

List of Tables viii

1 Introduction 1
- 1.1 General Background . . . 1
- 1.2 Objectives and Purpose . . . 2
- 1.3 Significance and Importance . . . 3
- 1.4 Summary of Contributions and Achievements . . . 3
- 1.5 Organization of the Report . . . 5

2 Literature Review 6
- 2.1 Natural Language Processing in Islamic Text Analysis . . . 6
  - 2.1.1 Arabic NLP Foundations . . . 6
  - 2.1.2 Quranic Text Processing . . . 6
  - 2.1.3 Advanced Quranic Applications . . . 7
- 2.2 Hadith Processing and Classification . . . 7
- 2.3 Religious Question-Answering Systems . . . 7
- 2.4 Existing Commercial Applications . . . 8
- 2.5 Limitations of Current Islamic Digital Platforms . . . 8
  - 2.5.1 Fragmented User Experience . . . 8
  - 2.5.2 Absence of Social Learning Components . . . 8
  - 2.5.3 Limited Translation of AI Research to Commercial Platforms . . . 9
  - 2.5.4 Lack of Integrated Multi-AI Functionality . . . 9
  - 2.5.5 Personalization and Content Discovery Challenges . . . 9
- 2.6 Research Gaps and Opportunities . . . 9
- 2.7 Summary . . . 10

3 Methodology 11
- 3.1 Development Approach Overview . . . 11
- 3.2 Application Development Methodology . . . 11
  - 3.2.1 Frontend Architecture . . . 11
  - 3.2.2 Backend Architecture . . . 12
  - 3.2.3 Cross-Platform Strategy . . . 12
  - 3.2.4 Core Functional Modules . . . 12
  - 3.2.5 Performance Optimization . . . 12
  - 3.2.6 Development Challenges & Solutions . . . 13
- 3.3 AI Development Methodology . . . 13
  - 3.3.1 K-Means Clustering for Quranic Topic Modeling . . . 13

3.3.2 Retrieval-Augmented Generation (RAG) Chatbot . . . 15
3.3.3 Content Recommendation System . . . 16
3.4 Development Tools and Practices . . . 16

4 System Analysis **17**
4.1 Functional Requirements . . . 17
4.2 Non-Functional Requirements . . . 18
4.3 Sequence Diagrams and Descriptions . . . 19
4.3.1 Social Media Features Flow . . . 19
4.3.2 Sheikh Verification and Admin Management . . . 24
4.3.3 RAG AI Assistant and Religious Guidance . . . 26
4.3.4 Quran Reading and Study Features . . . 27
4.3.5 Push Notification and Messaging System . . . 29
4.3.6 Chat and Messaging System . . . 31
4.3.7 Authentication and User Management . . . 33
4.4 Database Design . . . 37

5 Results **39**
5.1 Overview of Completed Features . . . 39
5.1.1 Core Application Infrastructure . . . 39
5.1.2 User Authentication and Profile Management . . . 39
5.1.3 Social Community Platform . . . 39
5.1.4 Digital Quran Interface . . . 40
5.1.5 Real-time Communication System . . . 40
5.2 AI Model Implementation Results . . . 40
5.2.1 RAG-based Islamic Fatawa Chatbot . . . 40
5.2.2 Quranic Topic Modeling and Visualization . . . 40
5.2.3 Fuzzy Logic Content Recommendation System . . . 41
5.2.4 Arabic NLP Processing Results . . . 41
5.2.5 Model Training Outcomes . . . 41
5.3 Final Application Screenshots . . . 41
5.3.1 Mobile App UI . . . 41
5.3.2 Web App UI . . . 60
5.4 System Integration and Deployment . . . 77

6 Discussion **78**
6.1 Interpretation and Analysis of Results . . . 78
6.1.1 Problem Resolution Assessment . . . 78
6.1.2 Technical Achievement Analysis . . . 78
6.1.3 Platform Integration Success . . . 79
6.2 Contributions to the Field . . . 79
6.2.1 Technical Contributions . . . 79
6.2.2 Methodological Contributions . . . 79
6.2.3 Educational and Social Impact . . . 80
6.3 Limitations and Challenges . . . 80
6.3.1 Technical Limitations . . . 80
6.3.2 Platform Limitations . . . 80
6.3.3 Dataset and Language Processing Limitations . . . 81
6.4 Positioning Within Current Knowledge . . . 81
6.5 Logical Implications of Results . . . 81

7 Conclusion and Recommendations **82**
7.1 Summary of Key Results 82
7.2 Learning Experience and Reflections 82
7.3 Recommendations for Future Enhancement 83
7.3.1 Advanced Islamic AI Systems 83
7.3.2 Enhanced Community Features 83
7.3.3 Scalability and Performance 84
7.4 Open Problems and Challenges 84
7.5 Final Remarks 84

References **86**

List of Figures

4.1 Social Media Notification Handling Flow . . . 20
4.2 Social Media Post Interactions Flow . . . 21
4.3 Social Media Post Creation Flow . . . 22
4.4 Social Media Feed Browsing and Filtering Flow . . . 23
4.5 Sheikh Verification Request Process . . . 24
4.6 Sheikh Verification Admin Review Process . . . 25
4.7 RAG AI Assistant Query Processing Flow . . . 26
4.8 Quran Reading Progress Tracking Flow . . . 27
4.9 Quran Bookmark Management Flow . . . 28
4.10 Push Notification Trigger System . . . 29
4.11 Push Notification Handling Flow . . . 30
4.12 Chat Conversation Setup Flow . . . 31
4.13 Real-time Chat Messaging Flow . . . 32
4.14 Firebase Authentication for Chat System . . . 32
4.15 Authentication Token Refresh Flow . . . 33
4.16 Password Reset Flow . . . 34
4.17 User Registration Flow . . . 35
4.18 Profile Management Flow . . . 36
4.19 User Logout Flow . . . 36
4.20 User Login Flow . . . 37
4.21 FCM Token Registration Flow . . . 37
4.22 Database UML Diagram for Zikr Platform . . . 38

5.1 Sign in and account creation process . . . 42
5.2 Sheikh verification and email confirmation . . . 43
5.3 Password recovery workflow (1) . . . 44
5.4 Password recovery workflow (2) . . . 44
5.5 Password reset interface . . . 45
5.6 Home page with and without notifications . . . 46
5.7 Posts feed with suggested (top) and recent (bottom) content views . . . 47
5.8 Topic-based content filtering system . . . 48
5.9 Additional topic filtering options . . . 48
5.10 Post creation interface and personal posts management . . . 49
5.11 Quran verse highlighting . . . 49
5.12 Quran navigation . . . 50
5.13 Verse search and topic clustering visualization . . . 50
5.14 Verse details and bookmark management . . . 51
5.15 Audio recitation and Tafseer integration . . . 51
5.16 Quran page reciter selection interface . . . 52

5.17 RAG chatbot conversation interface . . . 53
5.18 messaging interface . . . 54
5.19 Chat list . . . 54
5.20 daily Azkar feature . . . 55
5.21 Prayer times . . . 55
5.22 User search . . . 56
5.23 profile viewing . . . 56
5.24 Account settings and profile management . . . 57
5.25 Logout confirmation interface . . . 57
5.26 Sheikh verification process and administrative review . . . 58
5.27 Sheikh verification email notifications . . . 58
5.28 Prayer, social interaction, and messaging notifications . . . 59
5.29 Web login interface and account creation . . . 60
5.30 Personal details and sheikh certification upload . . . 61
5.31 Email verification process . . . 61
5.32 Password recovery workflow . . . 62
5.33 Web application home dashboard . . . 63
5.34 Suggested and recent posts feed views . . . 64
5.35 Topic-based content filtering system . . . 65
5.36 Personal posts management and liked posts collection . . . 66
5.37 Quran reading interface with verse navigation . . . 67
5.38 Quran page with topic highlighting, clustering guide, and full-screen mode . . . 68
5.39 Verse details floating interface and bookmarking . . . 69
5.40 Audio recitation and Tafseer selection interfaces . . . 70
5.41 Bookmarked verses archive . . . 70
5.42 RAG chatbot conversation interface and interactions . . . 71
5.43 User search and chat interface variations . . . 72
5.44 Morning and evening Azkar interfaces . . . 73
5.45 Prayer times display with location-based information . . . 73
5.46 Account settings and profile editing interface . . . 74
5.47 Password change and logout confirmation . . . 75
5.48 Administrative dashboard for pending sheikh verification requests . . . 76
5.49 Sheikh certification review interface . . . 76
5.50 Rejection reason interface for sheikh verification . . . 77

List of Tables

4.1 Functional Requirements for Zikr Platform 18
4.2 Non-Functional Requirements for Zikr Platform 19

Chapter 1

Introduction

The intersection of Artificial Intelligence and Islamic education represents a rapidly evolving field with significant potential to enhance religious learning and community engagement. Traditional Islamic educational tools and applications have largely focused on individual functionalities such as prayer times, Quran recitation, or basic religious content delivery. However, the integration of advanced AI techniques with social community features presents unprecedented opportunities to create more engaging, personalized, and educationally effective platforms for the Muslim community.

This report presents the development of Zikr, an innovative platform that combines social community features with advanced Artificial Intelligence to support Muslims worldwide in their religious journey. Unlike conventional Islamic applications that address single functionalities in isolation, Zikr represents a holistic approach that integrates AI-powered educational tools within a vibrant social environment, creating a comprehensive digital ecosystem for Islamic learning and community interaction.

1.1 General Background

The integration of Artificial Intelligence with social media platforms and religious content represents a complex interdisciplinary challenge that requires careful consideration of technological capabilities, cultural sensitivities, and community needs. In the context of Islamic digital platforms, this integration must address the unique requirements of Muslim users who seek authentic religious guidance, meaningful community interaction, and content that aligns with Islamic values and principles.

Modern social media platforms, while facilitating global connectivity and information sharing, often present challenges for Muslim users including exposure to inappropriate content, algorithmic biases that may not reflect Islamic values, and recommendation systems that prioritize engagement over educational or spiritual benefit. The typical social media environment, with its emphasis on viral content, celebrity culture, and commercial advertising, can create distractions that conflict with the spiritual focus and mindful consumption of content that Islamic teachings encourage.

The application of Artificial Intelligence in religious contexts presents both opportunities and challenges. AI techniques such as natural language processing can enhance the accessibility and understanding of Islamic texts, particularly for non-Arabic speakers or those learning the language. Machine learning algorithms can personalize content delivery to match individual learning preferences and spiritual needs. However, the implementation of AI in religious contexts requires careful attention to accuracy, authenticity, and the preservation of traditional interpretive methodologies that have been developed over centuries of Islamic scholarship.

Creating a comfortable digital space for Muslims involves addressing several key considerations: ensuring content authenticity through proper verification mechanisms, providing culturally appropriate interaction frameworks that respect Islamic social guidelines, implementing moderation systems that maintain the platform's spiritual focus, and developing recommendation algorithms that balance personalization with exposure to diverse Islamic perspectives and knowledge areas.

The challenge of integrating social features within an Islamic platform extends beyond technical implementation to encompass the creation of community guidelines, user behavior expectations, and moderation policies that reflect Islamic principles of respectful discourse, beneficial knowledge sharing, and collective spiritual growth. This requires developing systems that can distinguish between constructive religious discussion and potentially harmful content while maintaining an environment conducive to learning and spiritual development.

The linguistic complexity of Arabic, particularly classical Arabic used in religious texts, presents unique challenges for AI implementation. Effective processing of Islamic content requires understanding of religious terminology, contextual interpretation, and the ability to maintain the nuanced meanings that are central to Islamic scholarship. This necessitates specialized approaches to data preprocessing, model training, and content generation that go beyond general-purpose language processing techniques.

1.2 Objectives and Purpose

The primary aim of this project is to develop an AI-enhanced Islamic educational platform that integrates advanced machine learning techniques with social community features to create a comprehensive digital environment for Islamic learning and community engagement.

Primary Objectives:

1. **Retrieval-Augmented Generation (RAG) Chatbot Development:** Design and implement an intelligent question-answering system capable of responding to Islamic Fatawa questions in both Arabic and English, utilizing a comprehensive dataset scraped from IslamQA to ensure authentic and accurate religious guidance.

2. **Quranic Topic Modeling and Visualization:** Develop a K-Means clustering model to identify and highlight distinct Quranic topics with color-coded visualization, enhancing the recitation and study experience through AI-powered content analysis.

3. **Intelligent Content Recommendation System:** Create a personalized content suggestion algorithm for the social feed using fuzzy logic principles to recommend posts based on user interests, activity patterns, and engagement history.

4. **Social Community Platform Integration:** Build a comprehensive social networking framework that enables users to create posts, comment, like, and interact while maintaining Islamic principles and fostering meaningful religious discussions.

5. **Cross-platform Application Development:** Implement a robust Flutter frontend with Django backend architecture to ensure seamless functionality across mobile and web platforms while supporting complex AI integrations.

6. **Dataset Collection and Preprocessing:** Establish comprehensive data collection and preprocessing pipelines for Islamic content, addressing the challenge of scarce high-quality Arabic religious datasets.

7. **Arabic Natural Language Processing:** Develop and optimize NLP capabilities for processing classical and modern Arabic religious texts, enabling accurate understanding and generation of Islamic content.

The purpose of this work extends beyond traditional application development to explore the potential of combining modern AI techniques with user-centric design principles for creating impactful digital tools in Islamic education. The project aims to demonstrate how advanced machine learning algorithms can be effectively integrated into religious platforms while maintaining authenticity, accuracy, and cultural sensitivity.

1.3 Significance and Importance

This project addresses several critical challenges in the intersection of AI and Islamic education while making significant contributions to both fields:

Technical Significance: The development of specialized AI models for Arabic Islamic content represents a significant advancement in natural language processing for religious texts. The implementation of RAG systems for Fatawa question-answering demonstrates novel applications of retrieval-augmented generation in religious contexts, where accuracy and authenticity are paramount. The K-Means clustering approach for Quranic topic modeling introduces innovative visualization techniques for religious text analysis.

Educational Innovation: The integration of AI-powered personalization with social learning principles creates a new paradigm for religious education platforms. The fuzzy logic-based recommendation system represents an advancement in content suggestion algorithms specifically designed for religious social networks, balancing personalization with content diversity to enhance learning outcomes.

Methodological Advances: The project demonstrates effective methodologies for processing classical Arabic religious texts, handling the complexity of Islamic terminology, and maintaining religious authenticity while implementing modern AI techniques. The dual-track development approach (application development and AI integration) provides a framework for similar interdisciplinary projects.

Social Impact: The platform addresses the need for safe, educationally focused digital spaces for Muslim communities, particularly important in environments where traditional Islamic education may be limited. The combination of AI-powered learning tools with community features creates opportunities for enhanced religious engagement and peer learning.

Interdisciplinary Contribution: The project bridges computer science, Islamic studies, and educational technology, demonstrating how modern AI techniques can be respectfully and effectively applied to traditional religious content while maintaining cultural and religious sensitivities.

1.4 Summary of Contributions and Achievements

This project delivers several significant contributions to the intersection of Artificial Intelligence and Islamic education:

AI and Machine Learning Contributions:

- Developed a Retrieval-Augmented Generation (RAG) system specifically designed for Islamic Fatawa question-answering, demonstrating novel applications of RAG technology in religious contexts

- Implemented K-Means clustering for Quranic topic modeling with innovative color-coded visualization, advancing the field of religious text analysis; applied this approach specifically to Tafseer Al-Saadi using a dataset that was thoroughly cleaned and preprocessed
- Created a fuzzy logic-based recommendation system tailored for Islamic social content, balancing personalization with content diversity requirements
- Utilized publicly available and web-scraped datasets for Arabic Islamic content, performing cleaning and editing to enhance data quality for Arabic NLP tasks

Technical Achievements:

- Successfully integrated multiple AI models within a cross-platform application architecture supporting mobile and web platforms
- Developed robust Arabic natural language processing capabilities for handling classical and modern Islamic texts
- Implemented seamless integration between AI services and social community features through scalable API architectures
- Created comprehensive data collection and preprocessing systems addressing the challenge of scarce high-quality Islamic datasets

Educational and Social Innovation:

- Pioneered the combination of AI-powered educational tools with social community features in Islamic education platforms
- Demonstrated effective methodologies for maintaining religious authenticity while implementing modern AI techniques
- Created a safe, distraction-free digital environment that promotes meaningful Islamic learning and community engagement
- Established frameworks for culturally sensitive AI implementation in religious contexts

Methodological Contributions:

- Developed dual-track development methodology combining traditional software engineering with specialized AI development approaches
- Created evaluation frameworks for assessing AI accuracy in religious content domains
- Established best practices for processing complex Arabic religious texts while maintaining semantic accuracy
- Demonstrated successful interdisciplinary collaboration between computer science and Islamic studies

The project successfully demonstrates the potential of combining modern AI techniques with user-centric design to create impactful digital tools for Islamic education and community building.

1.5 Organization of the Report

This report is organized into seven chapters, each addressing a key aspect of the development and evaluation of the Zikr platform:

Chapter 1 - Introduction introduces the project context, outlining the general background of AI applications in Islamic education, the specific objectives and purpose of developing Zikr, the significance and importance of combining social features with AI-powered Islamic tools, and a summary of the key contributions and achievements realized through this interdisciplinary project.

Chapter 2 - Literature Review provides a comprehensive examination of existing research in Natural Language Processing applications for Islamic texts, reviews current commercial Islamic applications and their limitations, analyzes academic advances in Arabic NLP and religious text processing, and identifies critical research gaps that motivated the development of an integrated AI-powered social platform for Islamic education.

Chapter 3 - Methodology details the comprehensive dual-track development approach, covering both application development methodologies (frontend architecture with Flutter, backend implementation with Django REST Framework, cross-platform strategies) and AI development processes (K-Means clustering for Quranic topic modeling, RAG chatbot implementation, fuzzy logic recommendation system), along with the development tools, practices, and integration strategies employed throughout the project.

Chapter 4 - System Analysis presents the detailed system requirements analysis including comprehensive functional and non-functional requirements, illustrates key system interactions through sequence diagrams covering social media features, authentication flows, AI chatbot operations, and communication systems, and provides the complete database design with entity-relationship modeling that supports the platform's complex functionality.

Chapter 5 - Results demonstrates the comprehensive implementation outcomes, showcasing the completed core application infrastructure, AI model implementation results, extensive mobile and web application interface screenshots illustrating all major features, and evidence of successful system integration and deployment readiness.

Chapter 6 - Discussion provides critical analysis of the achieved results against original objectives, examines the technical contributions to AI applications in Islamic education, evaluates the platform's success in addressing identified limitations of existing Islamic applications, discusses encountered challenges and implemented solutions.

Chapter 7 - Conclusion and Recommendations summarizes the key technical and educational achievements, reflects on the learning experience and project insights, provides detailed recommendations for future enhancements including advanced Islamic AI systems and improved community features, identifies open problems and ongoing challenges in the field, and concludes with final remarks on the project's impact.

Chapter 2

Literature Review

This literature review serves two primary purposes: to provide awareness of previous work that has addressed the topic of Natural Language Processing (NLP) applications in Islamic texts, and to give readers sufficient background regarding the topics discussed in this research. The review examines the current state of computational approaches to Islamic text analysis, existing applications and platforms, and identifies gaps in the current research landscape.

2.1 Natural Language Processing in Islamic Text Analysis

The application of Natural Language Processing techniques to Islamic texts has gained significant attention in recent years, driven by the need to make religious knowledge more accessible and to preserve the authenticity of Islamic teachings through computational methods.

2.1.1 Arabic NLP Foundations

The Arabic language presents unique challenges for NLP applications due to its complex morphology, script variations, and dialectal differences. Habash (2022) provides comprehensive background information for working with Arabic in its various forms: Classical, Modern Standard, and Dialectal Arabic. This tutorial discusses various Arabic linguistic phenomena and reviews the state-of-the-art in Arabic processing, covering enabling technologies, resources, common tasks, and applications. The work emphasizes the importance of understanding Arabic's complexity when developing NLP systems for religious texts.

Almanea (2024) presents a comprehensive survey of deep learning applications to written Arabic, analyzing 111 studies published between 2020-2024. The study categorizes research under seven linguistic fields: forensic linguistics, educational linguistics, text linguistics, optical character recognition (OCR), artificial intelligence chatbots, poetry studies, and discourse analysis. The overall accuracy of deep learning models for various Arabic linguistic tasks reaches 90.83%, with particularly high performance in OCR (98.11%) and text linguistics (93.57%).

2.1.2 Quranic Text Processing

Several studies have focused specifically on computational approaches to Quranic text analysis. Touati-Hamad et al. (2020) examines various word representation tools for Quranic content in Arabic, covering both local and distributed representation forms for use in machine learning and deep learning algorithms. The study emphasizes the importance of proper word representation as a starting point for NLP applications in religious texts.

Bashir et al. (2023) provides a systematic review of Arabic NLP for Quranic research, serving as a comprehensive compendium of tools, datasets, and approaches. The survey covers the gamut from automated morphological analysis to correction of Quranic recitation via speech recognition, highlighting the diverse applications of NLP in Quranic studies.

Slamet et al. (2016) demonstrates practical text mining applications to Quranic verses using K-means clustering algorithm. The study processed 6,236 verses using both unstemmed and stemmed words, establishing three distinct clusters and providing insights into the structural organization of Quranic content.

2.1.3 Advanced Quranic Applications

Recent research has explored more sophisticated applications of NLP to Quranic texts. Shohoud et al. (2023) developed a semantic search tool for the Quran using Arabic NLP techniques. The system trains models on over 30 tafsirs and uses cosine similarity to find verses pertaining to user inquiries, achieving similarity scores as high as 0.97 for specific topics.

Basem et al. (2024) addresses the challenge of Quranic question-answering systems by expanding the QA 2023 shared task dataset from 251 to 629 questions, creating a comprehensive set of 1,895 questions categorized into single-answer, multi-answer, and zero-answer types. Using fine-tuned transformer models including AraBERT, RoBERTa, CAMeLBERT, AraELECTRA, and BERT, the best model (AraBERT-base) achieved significant improvements with MAP@10 of 0.36 and MRR of 0.59.

Nigar et al. (2023) presents an intelligent framework for online Quran learning during the COVID-19 pandemic, using deep learning to identify correct recitation of individual alphabets, words, and complete verses. The system employs MFCC for voice feature extraction and LSTM for classification, achieving 97.7% accuracy.

2.2 Hadith Processing and Classification

Hadith literature represents another significant area of Islamic text processing research, with unique challenges related to narrator authentication and classification.

Ramzy et al. (2023) introduces a novel Author-Based Hadith Classification Dataset (ABCD) containing 29K Hadiths with 18K unique narrators. The research applies both machine learning and deep learning approaches to classify Hadiths based on their origin of narration. Results show that ML performs better than DL using Matn (content) data with 77% F1-score, while DL performs better using Sanad (chain of narrators) data with 92% F1-score.

Mahmoud et al. (2022) addresses narrator disambiguation in Hadith authentication through the AR-Sanad 280K dataset containing artificial sanads for identifying 18,298 narrators. The research models narrator disambiguation as a multiclass classification problem, achieving 92.9% Micro F1 score and 30.2% sanad error rate using fine-tuned AraBERT models.

Sulistio et al. (2024) provides a systematic literature review of machine learning applications in Hadith studies, analyzing 48 published journals. The review reveals that Hadith Classification is the most discussed topic (33.33%), Arabic is the most widely used language (43.75%), and SVM is the most common algorithm (12.5%).

2.3 Religious Question-Answering Systems

The development of intelligent question-answering systems for religious content represents a growing area of research. Alan et al. (2024) proposes MufassirQAS, a RAG-based question-answering system for understanding Islam. The system uses a vector database approach to

enhance LLM accuracy and transparency, incorporating Turkish translations and interpretations of Islamic texts while implementing careful prompt engineering to prevent harmful or offensive responses.

2.4 Existing Commercial Applications

Several commercial applications have been developed to serve the Muslim community's needs for Quranic learning and recitation:

Tarteel AI represents a significant advancement in AI-powered Quranic learning tools. This mobile application uses voice recognition and AI feedback to help Muslims memorize and recite the Quran more effectively. Key features include verse hiding for memorization testing, real-time mistake detection during recitation (identifying missed, incorrect, or extra words), progress tracking, customizable goals, and access to audio recitations by various Qaris. Used by millions worldwide, Tarteel AI prioritizes privacy and accessibility with a clean interface and no advertisements in its free version. Advanced features like mistake detection and verse-hiding are available through premium subscription. Community feedback indicates that it "does an excellent job in recognizing Quran recitation and providing real-time feedback," though some users note that the most powerful features require payment.

The Great Quran App offers a comprehensive digital Quran experience with Uthmani script, audio recitations, tafseer, and translations. The application supports verse search functionality, bookmarking, note-taking, night mode, and basic interface customization, providing users with essential tools for Quranic study and reference.

2.5 Limitations of Current Islamic Digital Platforms

While existing applications have made significant contributions to Islamic digital education, and academic research has demonstrated the potential of AI techniques for Islamic text processing, several critical limitations emerge when examining the current landscape of available platforms and applications:

2.5.1 Fragmented User Experience

Current Islamic applications typically focus on single functionalities—either Quranic recitation (like Tarteel AI), text reading (like The Great Quran App), or basic question-answering. Although academic research has shown the potential for sophisticated AI applications—such as Shohoud et al. (2023)'s semantic search research achieving 0.97 cosine similarity scores, or Basem et al. (2024)'s improved QA research with 63% performance gains—these research findings have not been translated into comprehensive commercial platforms. This fragmented approach requires users to switch between multiple applications to fulfill their diverse religious needs, creating a disjointed user experience that lacks cohesion and community integration.

2.5.2 Absence of Social Learning Components

The most significant gap lies in the complete absence of platforms that combine Islamic education with social learning features. While academic research such as Nigar et al. (2023) addressed online learning methodologies during the pandemic, and Alan et al. (2024) developed sophisticated question-answering system architectures, no existing commercial platform incorporates social media elements that could foster community engagement, peer learning, and collaborative religious discussions. Current applications like Tarteel AI and The Great

Quran App operate in isolation, limiting the potential for communal learning experiences that are fundamental to Islamic education traditions.

2.5.3 Limited Translation of AI Research to Commercial Platforms

Although individual research studies have demonstrated successful AI applications for Islamic texts—such as Ramzy et al. (2023)'s Hadith classification achieving 92% F1-scores, Mahmoud et al. (2022)'s narrator disambiguation with 92.9% accuracy, or Slamet et al. (2016)'s clustering techniques for Quranic verses—these advanced AI capabilities have not been integrated into user-facing platforms. The gap between academic research potential and commercial application reality is substantial, with existing apps offering only basic functionality compared to the sophisticated AI techniques demonstrated in research literature.

2.5.4 Lack of Integrated Multi-AI Functionality

Current platforms implement single AI features at best (like Tarteel AI's voice recognition), while research shows the potential for multiple AI techniques working together. Studies demonstrate various AI approaches—RAG systems (Alan et al., 2024), clustering algorithms (Slamet et al., 2016), semantic search (Shohoud et al., 2023), and classification models (Ramzy et al., 2023)—but no existing platform integrates multiple AI functionalities within a unified environment, let alone combines them with social interaction capabilities.

2.5.5 Personalization and Content Discovery Challenges

While research literature shows exploration of content analysis and recommendation techniques—such as Sulistio et al. (2024)'s identification of trending topics in Hadith studies and Basem et al. (2024)'s improved question-answering accuracy—existing commercial platforms lack sophisticated personalized content delivery systems. There is no platform that leverages user behavior, interests, and social interactions within a community setting to provide intelligent content recommendations for Islamic education.

2.6 Research Gaps and Opportunities

Despite the significant progress in Islamic text processing research, several critical gaps remain when considering the translation of this research into practical, user-facing platforms:

1. **Research-to-Platform Translation Gap**: While academic research demonstrates high accuracy in various AI techniques for Islamic texts, these findings have not been implemented in comprehensive, user-friendly platforms.

2. **Social Integration Gap**: Complete absence of platforms that combine proven AI techniques with social learning methodologies for Islamic education.

3. **Multi-AI Integration Gap**: No existing platform unifies multiple AI functionalities (question-answering, content analysis, topic modeling, recommendation systems) demonstrated in research literature within a single environment.

4. **Community-Driven Learning Gap**: Lack of platforms that leverage social interactions and community engagement to enhance Islamic learning experiences.

5. **Comprehensive User Experience Gap**: Existing platforms address single use cases, while research suggests the potential for holistic solutions that combine multiple AI capabilities with social features.

6. **Safe Social Space Gap**: Absence of distraction-free, religiously-focused social platforms that maintain Islamic values while providing modern social media functionality and AI assistance.

These gaps highlight the critical need for an innovative platform that bridges the divide between academic AI research achievements and practical user applications, combining multiple proven AI techniques with social community features in a single, integrated environment.

2.7 Summary

This literature review reveals a rich and growing field of research in Islamic text processing, with academic studies demonstrating remarkable achievements in AI applications for Islamic texts. Research has progressed from basic text processing to sophisticated deep learning approaches, achieving high accuracy rates across various tasks—such as 97.7% accuracy in Quranic recitation recognition research (Nigar et al., 2023), 92% F1-scores in Hadith classification studies (Ramzy et al., 2023), and 0.97 cosine similarity in semantic search research (Shohoud et al., 2023).

However, a significant disconnect exists between research achievements and available commercial platforms. While individual research studies have demonstrated the potential for advanced AI applications in Islamic education, these findings have not been translated into comprehensive, user-facing platforms. Current commercial applications like Tarteel AI and The Great Quran App offer limited functionality compared to the sophisticated AI techniques proven effective in academic research.

Most critically, no existing platform combines the educational potential demonstrated in research literature with social community features. The complete absence of AI-powered social platforms for Islamic education represents the most significant gap in the current landscape. This presents a substantial opportunity to create a comprehensive platform that bridges academic research achievements with practical user needs, integrating multiple AI functionalities within a social learning environment.

The gaps identified in this review—particularly the absence of platforms that translate advanced AI research into user-friendly social learning environments, the lack of integrated multi-functional AI systems in commercial applications, and the need for community-driven, personalized content delivery—establish a clear foundation for innovative solutions that can transform how Muslims engage with their faith in the digital age.

Chapter 3

Methodology

This chapter details the comprehensive methodology adopted for the development of Zikr, an innovative AI-powered Islamic platform. The approach was carefully structured to ensure the seamless integration of advanced machine learning models with a robust, user-centric mobile and web application. The methodology is characterized by a dual-track process, encompassing both application development and artificial intelligence (AI) integration, with a strong emphasis on performance, authenticity, and user experience.

3.1 Development Approach Overview

The development of Zikr was organized into two parallel tracks: application development and AI model implementation. This structure allowed for specialized focus on both the technical infrastructure and the intelligent features of the platform. The application development track concentrated on building the core system, user interface, and social networking capabilities, while the AI track was dedicated to the design, training, and optimization of machine learning models tailored for Islamic content and user engagement. These tracks were closely coordinated, ensuring that integration points were identified early and that the final product would offer a cohesive experience.

3.2 Application Development Methodology

3.2.1 Frontend Architecture

The architecture of Zikr follows a modern client-server paradigm, with a clear separation between the frontend and backend components. The frontend was developed using the Flutter framework, chosen for its ability to deliver a consistent and responsive user experience across both mobile and web platforms. This framework selection eliminated the need to maintain separate codebases for different platforms while ensuring native-like performance and appearance. The design process emphasized modularity and reusability, with a component-based architecture that facilitated rapid development and maintenance. State management was handled using the Provider pattern, ensuring efficient data flow and UI updates throughout the application. This pattern was selected over alternatives like BLoC or Redux due to its simplicity and sufficient power for the application's needs. Special attention was given to responsive design, leveraging Flutter's ScreenUtil to adapt layouts to various device sizes and orientations. Service classes were implemented to encapsulate API communication, authentication, and notification logic, promoting code clarity and maintainability through the principle of separation of concerns.

3.2.2 Backend Architecture

On the backend, Django REST Framework was employed to provide a scalable and secure API layer, chosen for its rich ecosystem, excellent ORM capabilities, and comprehensive authentication options. The backend was responsible for user authentication, data storage, and business logic, including the management of social interactions and AI-driven features. JWT-based authentication was integrated to support secure user sessions, with additional provisions for social login and two-factor authentication. Token refresh mechanisms were implemented to balance security with user experience. The database schema was carefully normalized to support complex relationships between users, posts, comments, and other entities, with special consideration for query performance. Foreign key constraints and indexing strategies were employed to maintain data integrity and optimize retrieval operations. Real-time messaging and notifications were enabled through Firebase Cloud Messaging, ensuring timely updates and engagement. The API was designed with versioning support to allow for future evolutions while maintaining backward compatibility.

3.2.3 Cross-Platform Strategy

The cross-platform development strategy centered on maximizing code reuse while accommodating platform-specific requirements. Flutter's Dart code comprised approximately 95% of the application logic, shared between mobile and web versions. Platform detection was implemented through conditional imports and the kIsWeb constant, allowing for platform-specific code paths when necessary. Different network configurations were established for web (localhost) versus mobile (specific IP address) environments to accommodate development and testing workflows. Web-specific optimizations included cursor-based interaction patterns, keyboard shortcuts, and responsive layouts for larger screens. For mobile platforms, native capabilities such as camera access, local storage permissions, and push notification handling were implemented using platform channels and plugin configurations. Special attention was given to browser compatibility issues, particularly for Safari on iOS, which required custom handling for audio playback and scrolling behavior.

3.2.4 Core Functional Modules

The application was structured into several core functional modules. User authentication and profile management provided secure registration, login, and customization options, including a sheikh verification system for religious authority validation with document upload capabilities and admin review workflows. The social media module enabled users to create posts, comment, like, and interact within a safe and moderated environment with content filtering based on user preferences and community standards. Hierarchical commenting and content tagging supported rich discussions and content discovery through a flexible taxonomy system. The Quran features module offered digital access to the Quran, enhanced by AI-powered topic highlighting, reading progress tracking, and audio recitation with synchronized text highlighting. The implementation included support for multiple reciters and page-by-page navigation with verse lookup capabilities. The chat system facilitated real-time communication, including both direct messaging and group chats, and integrated the AI-powered RAG chatbot for religious queries with source citation and confidence scoring for answers.

3.2.5 Performance Optimization

Performance optimization was addressed through multiple strategies. API pagination was implemented at the server level with a carefully selected page size of five posts, balancing data

transfer efficiency with smooth scrolling in the social feed while minimizing perceived loading times. For the Quran interface, verse data was pre-loaded in small batches based on reading direction to ensure seamless page transitions. Caching mechanisms were deployed at multiple levels, including HTTP response caching for API calls, local storage for user preferences and frequently accessed data, and in-memory caching for active session data. Image loading was optimized through lazy loading techniques and progressive rendering, particularly important for social media content. For web deployments, asset compression and code splitting were implemented to reduce initial load times, while mobile builds utilized native image caching libraries to minimize network requests for previously viewed content. Background data prefetching was employed for likely user navigation paths based on usage patterns.

3.2.6 Development Challenges & Solutions

Several technical challenges emerged during development. Firebase Web/Mobile compatibility presented significant issues, particularly with authentication flows that functioned differently across platforms. This was resolved by implementing platform-specific authentication handlers that unified the user experience while accommodating underlying technical differences. Arabic text rendering proved challenging, especially for diacritical marks and special Quranic symbols. Custom font implementations and text direction controls were developed to ensure consistent display across devices. Push notification delivery on iOS devices required special handling due to Apple's strict background processing limitations, solved through a combination of silent notifications and local notification scheduling. Performance bottlenecks in the chat system during high message volume were addressed by implementing message batching and optimistic UI updates. JWT token expiration management required a custom middleware solution that transparently refreshed tokens before expiration while preserving the user's context and preventing session disruptions during active use.

3.3 AI Development Methodology

3.3.1 K-Means Clustering for Quranic Topic Modeling

Dataset Preparation and Preprocessing

For this model, Tafseer Al-Saadi dataset was acquired in JSON format, containing detailed verse-wise commentary. This tafseer was specifically selected over direct Quranic text for several strategic reasons:

- **First**, modern Arabic NLP tools are optimized for contemporary Arabic rather than the classical Quranic Arabic, making the tafseer text more accessible for analysis.
- **Second**, the explanatory nature of tafseer provided clearer topic indicators than the often metaphorical and concise Quranic verses.
- **Third**, the tafseer frequently explained consecutive verses together, naturally grouping related content and establishing contextual boundaries that would be beneficial for topic modeling.

The data underwent extensive cleaning, including the extraction of relevant fields, handling of missing values by replacing them with tafseer Al-Muyassar since it was the closest in nature, and normalization of Arabic text. Diacritics were removed using the CAMeL Tools library, and non-Arabic characters and punctuation were filtered out to ensure consistency.

Linguistic processing was performed using the Stanza NLP pipeline, which provided tokenization and lemmatization tailored for Arabic, followed by the removal of stop words based on a comprehensive list that was a modified version of the Alrefaie (2019) Arabic stop words list that would be suitable for Quranic text analysis, for example, the word "Allah" was retained as it is a key term in Islamic texts, while common Arabic stop words were kept to ensure that the focus remained on meaningful content.

Feature Engineering and Model Training

Feature engineering for the K-Means clustering model involved transforming the processed text into numerical vectors using the TF-IDF method. Parameters such as maximum features (2151), minimum document frequency (10), and n-gram range (1, 2) were carefully selected based on dimensionality analysis with Truncated SVD, ensuring that the feature set captured the semantic richness of the text while maintaining computational efficiency. This optimal feature size was determined by analyzing the cumulative explained variance, which showed that 2151 features captured approximately 95% of the variance in the dataset.

The optimal number of clusters was determined through a combination of the elbow method and silhouette score analysis, testing values from 7 to 25 clusters. Plotting the inertia values (sum of squared distances of samples to their closest cluster center) revealed a noticeable inflection point around 20 clusters. This was further validated by calculating silhouette scores, which measure both cohesion within clusters and separation between clusters. The highest silhouette score was achieved with 20 clusters, confirming it as the optimal configuration for the dataset.

Each cluster was characterized by its top representative words, extracted from the centroid vector of the cluster. By sorting the feature weights in each centroid, the ten most significant terms were identified for each cluster. Based on these key terms, semantic labels were assigned through manual review and validation against Islamic scholarly sources. To enhance the user experience, each topic cluster was assigned a distinct color, which was then integrated into the frontend for real-time highlighting during Quran recitation.

Mobile Integration

The K-Means clustering model was trained and executed offline using Python, resulting in a dataset where each verse was assigned to one of 20 distinct topic clusters. These cluster assignments were exported as a static JSON file, which was then integrated directly into the mobile application as an asset. This approach eliminated the need for real-time processing on either the device or a server, ensuring optimal performance and offline functionality.

For implementation in the mobile interface, the application loads the pre-computed JSON file at startup, creating an in-memory mapping between verse references (in the format "surah:ayah") and their corresponding cluster numbers. A carefully selected palette of 20 distinct colors was defined in the application's styles, with each color corresponding to a specific topic cluster. When the user enables the topic highlighting feature, the app simply looks up the cluster assignment for each displayed verse and applies the appropriate color as a semi-transparent background layer, ensuring that the text remains readable while visually grouping related verses.

To enhance user comprehension, an interactive color legend feature was implemented, displaying the semantic meaning of each cluster alongside its corresponding color. The 20 identified topics included concepts such as "Divine Commands and Rulings", "Divine Mercy", and "The Prophet Muhammad and Revelation". This legend helps users understand the

thematic connections between differently colored verses, deepening their engagement with the text.

The user interface was designed with both functionality and aesthetics in mind. A toggle button allows readers to switch between normal and highlighted views, providing flexibility in how they interact with the Quranic text. The color legend interface was implemented with a semi-transparent floating action button that becomes more visible when interacted with, offering guidance without obstructing the reading experience. This direct integration of pre-computed clustering results into the visual presentation layer exemplifies how machine learning outputs can be effectively utilized in mobile applications without requiring continuous computational resources or network connectivity.

Validation and Topic Identification

The clustering model was rigorously validated through both quantitative and qualitative methods. For quantitative validation, the elbow method was employed, which plots the inertia (sum of squared distances to the nearest centroid) against varying numbers of clusters. This analysis revealed a distinct inflection point at 20 clusters, indicating diminishing returns for additional clusters beyond this point.

The silhouette score analysis further confirmed this finding. By measuring how similar each point is to its own cluster compared to other clusters, silhouette scores helped validate the optimal number of clusters. The analysis showed that 20 clusters provided the best balance between cluster cohesion and separation.

For qualitative validation, the top representative words from each cluster's centroid vector were extracted. These keywords revealed clear thematic patterns that aligned with recognized Quranic subjects. For example, one cluster contained terms strongly associated with divine attributes, while another focused on stories of prophets. The emergence of these distinct and interpretable topics from the unsupervised clustering process provided strong evidence for the validity of the approach.

Each of the 20 identified clusters demonstrated internal consistency and external distinctiveness, confirming that the model had successfully captured meaningful thematic structures within the Quranic tafseer text.

3.3.2 Retrieval-Augmented Generation (RAG) Chatbot

The Retrieval-Augmented Generation (RAG) chatbot was implemented using the Vext vector database service, with a strong focus on religious authenticity and precision. The underlying dataset consisted of 88k Fatawa entries sourced from the IslamQA dataset on Kaggle Elsaadany (2022). This data was converted into a suitable text format, chunked, and indexed within the Vext vector database to enable efficient retrieval. The generation component utilized the DeepSeek-V3 large language model, configured with a temperature of 0.4 to ensure precise and contextually appropriate responses. A top-k retrieval setting of three was chosen to balance relevance and diversity in the supporting context. Custom prompts were crafted to ensure that the model adhered strictly to religious topics and did not deviate into general conversation. The integration with the Flutter frontend was achieved through a dedicated Dart service, which managed API communication, session persistence, error handling, and the display of citations and source references.

3.3.3 Content Recommendation System

The content recommendation system for the mosocial feed was designed using fuzzy logic principles to balance personalization and diversity. User preferences were modeled by tracking likes on posts, with tag weights calculated to reflect individual interests. These personal preferences were combined with global content trends, with a weighting of 70% for user history and 30% for platform-wide popularity. The algorithm was activated when users selected the "Suggestions" filter, ensuring that the feed remained both relevant and varied. Performance was further optimized through efficient database queries, caching of user preferences, and the use of pagination to maintain recommendation quality across multiple page loads.

3.4 Development Tools and Practices

The project leveraged a modern technology stack, including Flutter and Dart for the frontend, Django REST Framework and Python for the backend, PostgreSQL for data storage, and Firebase for real-time features. AI and machine learning components were developed using Python libraries such as scikit-learn, Stanza NLP, and CAMeL Tools, with external services like Vext and DeepSeek-V3 providing advanced retrieval and generation capabilities. Version control was managed using Git, with a structured branching strategy and code review processes to ensure quality and maintainability. The development approach emphasized modularity, clear separation of concerns, and continuous improvement based on user feedback and performance monitoring.

In summary, the methodology adopted for Zikr reflects a careful balance between technical innovation, user-centered design, and religious authenticity. By combining advanced AI techniques with a robust application framework and ongoing validation, the project delivers a unique and impactful platform for Islamic education and community engagement.

Chapter 4

System Analysis

This chapter presents a comprehensive analysis of the Zikr platform's system requirements, architecture, and design. The analysis covers functional and non-functional requirements derived from the project objectives, sequence diagrams illustrating key system interactions, and the database design that supports the platform's comprehensive functionality.

4.1 Functional Requirements

Table 4.1 presents the comprehensive functional requirements for the Zikr platform, categorized by major system modules. These requirements were derived from the project objectives and implemented features across the social media, AI-powered educational tools, and Quran study components.

Table 4.1: Functional Requirements for Zikr Platform

Module	Requirement ID	Description
Authentication	FR-001	User registration with email verification
	FR-002	Secure login with JWT token-based authentication
	FR-003	Password reset functionality via email
	FR-004	Profile creation and management
	FR-005	Sheikh verification system with document upload
	FR-006	Two-factor authentication support
Social Media	FR-007	Create text and media posts with tagging system
	FR-008	Like and unlike posts and comments
	FR-009	Content filtering by categories and tags
	FR-010	Personalized feed with AI-driven suggestions
	FR-011	Post sharing
	FR-012	Media upload with Supabase storage integration
	FR-013	Real-time notifications for interactions
	FR-014	Search functionality for Quran verses and users
AI Chatbot	FR-015	RAG-powered Islamic Fatawa question answering
	FR-016	Arabic support
	FR-017	Source citation for chatbot responses
	FR-018	Response confidence scoring
	FR-019	Context-aware follow-up questions
	FR-020	Islamic knowledge base integration
	FR-021	Query preprocessing and validation
Quran Features	FR-022	Digital Quran with page-by-page navigation
	FR-023	AI-powered topic highlighting with K-Means clustering
	FR-024	Color-coded visualization of Quranic themes
	FR-025	Bookmark management for verses and pages
	FR-026	Reading progress tracking
	FR-027	Audio recitation with synchronized highlighting
	FR-028	Multiple reciter support
	FR-029	Verse search and lookup functionality
	FR-030	Tafseer integration (Al-Saadi)
	FR-031	Bookmarking verses
Communication	FR-032	Real-time direct messaging with Firebase
	FR-033	Message history
	FR-034	Push notification delivery

4.2 Non-Functional Requirements

Table 4.2 outlines the non-functional requirements that ensure the Zikr platform meets quality standards for performance, security, usability, and scalability. These requirements are critical for delivering a reliable and user-friendly experience across mobile and web platforms.

Table 4.2: Non-Functional Requirements for Zikr Platform

Category	Requirement ID	Description
Performance	NFR-001	API response time $\leq$ 500ms for standard requests
	NFR-002	AI chatbot response time $\leq$ 20 seconds
	NFR-003	Page load time $\leq$ 3 seconds on 4G networks
	NFR-004	Support for multiple concurrent users
	NFR-005	Memory usage $\leq$ 150MB for mobile applications
Security	NFR-006	JWT token-based authentication with refresh mechanism
	NFR-007	HTTPS encryption for all data transmission
	NFR-008	Input validation and SQL injection prevention
	NFR-009	Personal data encryption at rest
	NFR-010	Role-based access control (RBAC)
	NFR-011	Secure file upload with type validation
	NFR-012	GDPR compliance for data protection
Usability	NFR-013	Intuitive user interface with $\leq$ 3 clicks to major features
	NFR-014	Responsive design for devices 320px to 1920px width
	NFR-015	Support for Arabic RTL text rendering
	NFR-016	Accessibility compliance (WCAG 2.1 Level AA)
	NFR-017	Cross-platform consistency (iOS, Android, Web)
	NFR-018	User onboarding completion in $\leq$ 5 minutes
Scalability	NFR-019	Horizontal scaling capability for backend services
	NFR-020	CDN integration for global content delivery
	NFR-021	Microservices architecture for modular scaling
	NFR-022	Auto-scaling based on traffic patterns
Compatibility	NFR-023	iOS 12+ and Android 7+ support
	NFR-024	Modern web browsers (Chrome, Safari, Firefox, Edge)
	NFR-025	Multiple screen resolutions and orientations
	NFR-026	Integration compatibility with third-party services

4.3 Sequence Diagrams and Descriptions

This section presents sequence diagrams illustrating the key interactions within the Zikr platform. These diagrams demonstrate the flow of data and control between different system components for critical use cases including social media interactions, authentication processes, AI chatbot operations, Quran study features, and communication systems.

4.3.1 Social Media Features Flow

The social media functionality encompasses several key interactions including notification handling, post interactions, content creation, and feed browsing. Figure 4.1 illustrates the notification system that triggers when users interact with social content.

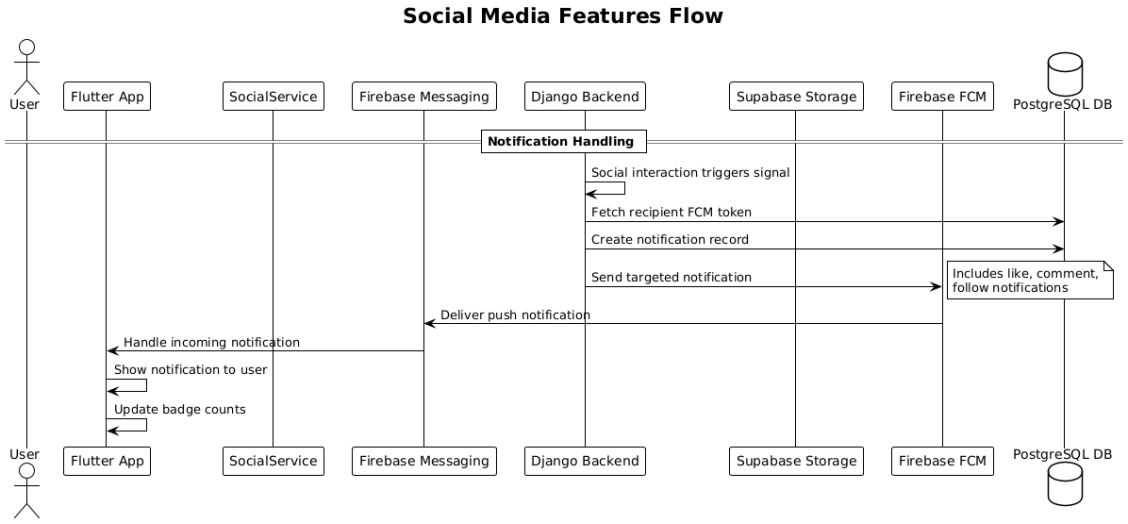

Figure 4.1: Social Media Notification Handling Flow

The notification handling sequence demonstrates the integration between social interactions and Firebase Cloud Messaging. When a user performs actions like liking or commenting, the system creates notification records, fetches recipient FCM tokens, and delivers targeted push notifications. This ensures real-time engagement and keeps users informed of relevant social activities.

Figure 4.2 shows the detailed flow for post interactions including likes and comments, demonstrating the immediate UI updates and backend processing.

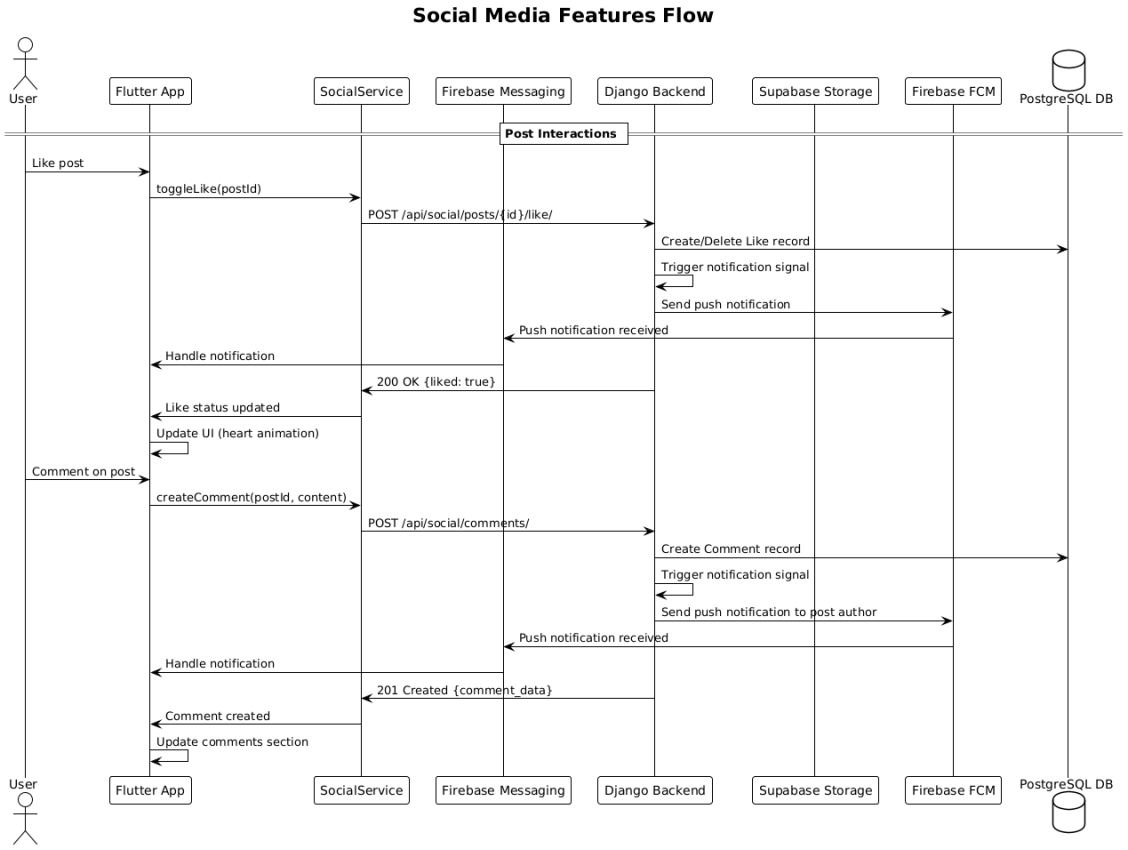

Figure 4.2: Social Media Post Interactions Flow

The post interaction sequence reveals how the system handles both like and comment operations. When a user likes a post, the system toggles the like status, updates the database, triggers notifications, and provides immediate UI feedback with heart animations. The commenting system creates new comment records and notifies the post author, maintaining engagement threads.

Figure 4.3 illustrates the post creation process, including media upload to Supabase storage and database record creation.

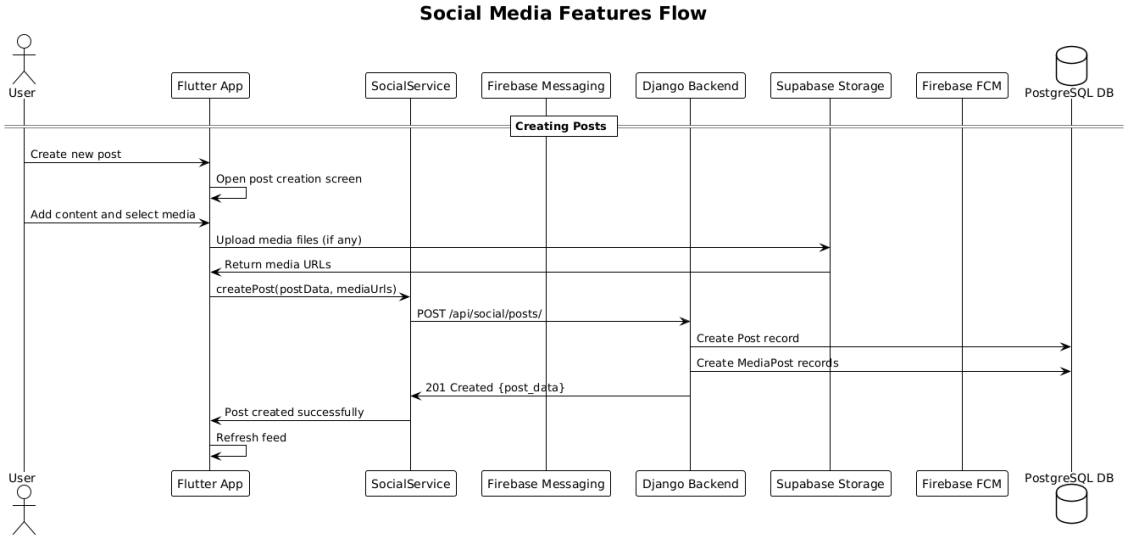

Figure 4.3: Social Media Post Creation Flow

The post creation flow demonstrates the comprehensive process from user input to published content. The sequence shows media file uploads to Supabase storage, URL retrieval, and the creation of both Post and MediaPost records in the database. This ensures rich content support while maintaining data consistency.

Figure 4.4 shows the feed browsing functionality, including the application of the AI-powered suggestion algorithm and content filtering mechanisms.

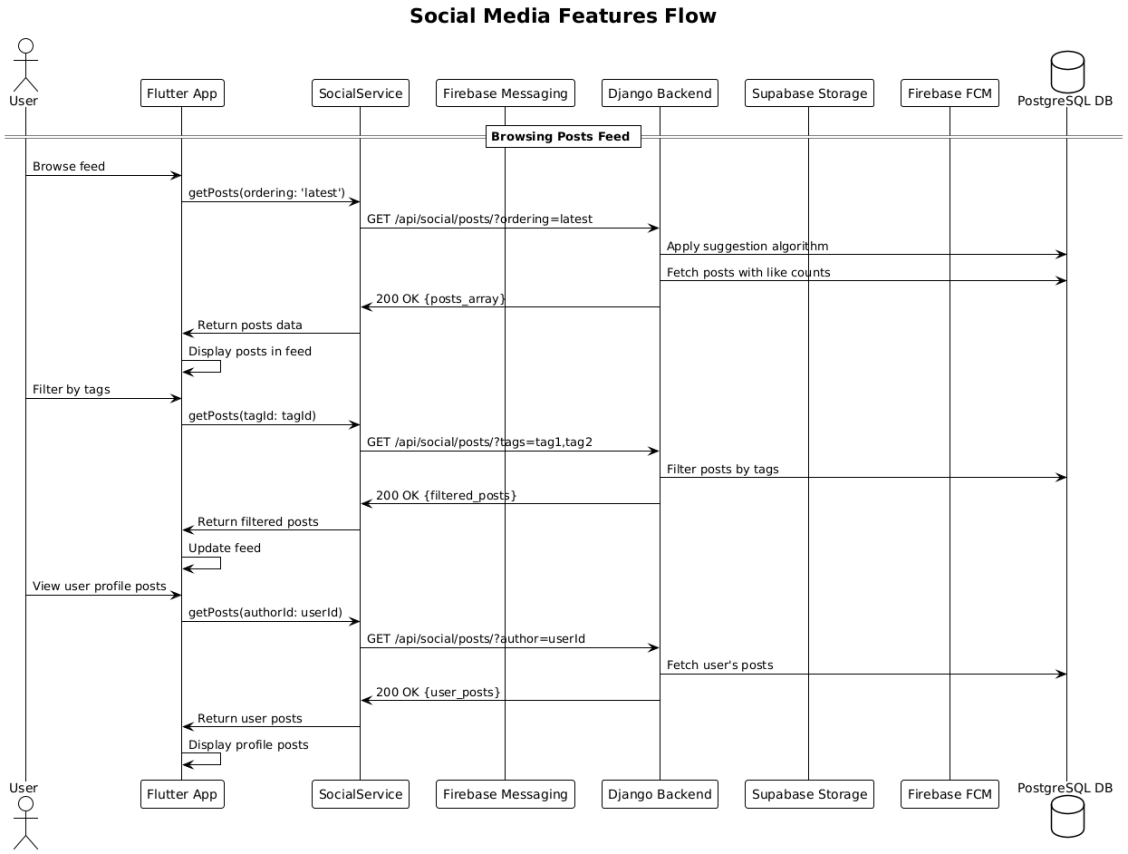

Figure 4.4: Social Media Feed Browsing and Filtering Flow

The feed browsing sequence illustrates how the platform serves personalized content. When users browse their feed, the system applies the suggestion algorithm to rank posts based on user preferences and global popularity. The sequence also shows filtering capabilities by tags and author-specific content viewing for profile pages.

4.3.2 Sheikh Verification and Admin Management

The platform includes a comprehensive sheikh verification system that ensures religious authority authenticity. Figure 4.5 shows the initial verification request process.

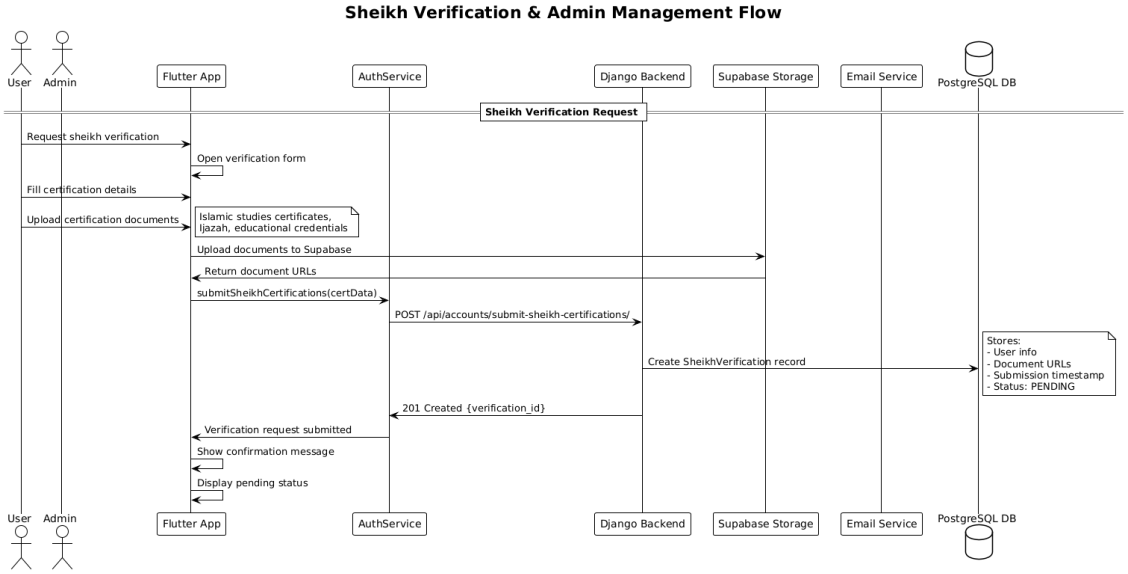

Figure 4.5: Sheikh Verification Request Process

The sheikh verification request process enables users to submit credentials for religious authority verification. Users upload Islamic studies certificates and educational credentials to Supabase storage, and the system creates verification records with pending status for admin review.

Figure 4.6 illustrates the admin review and approval/rejection process for sheikh verification requests.

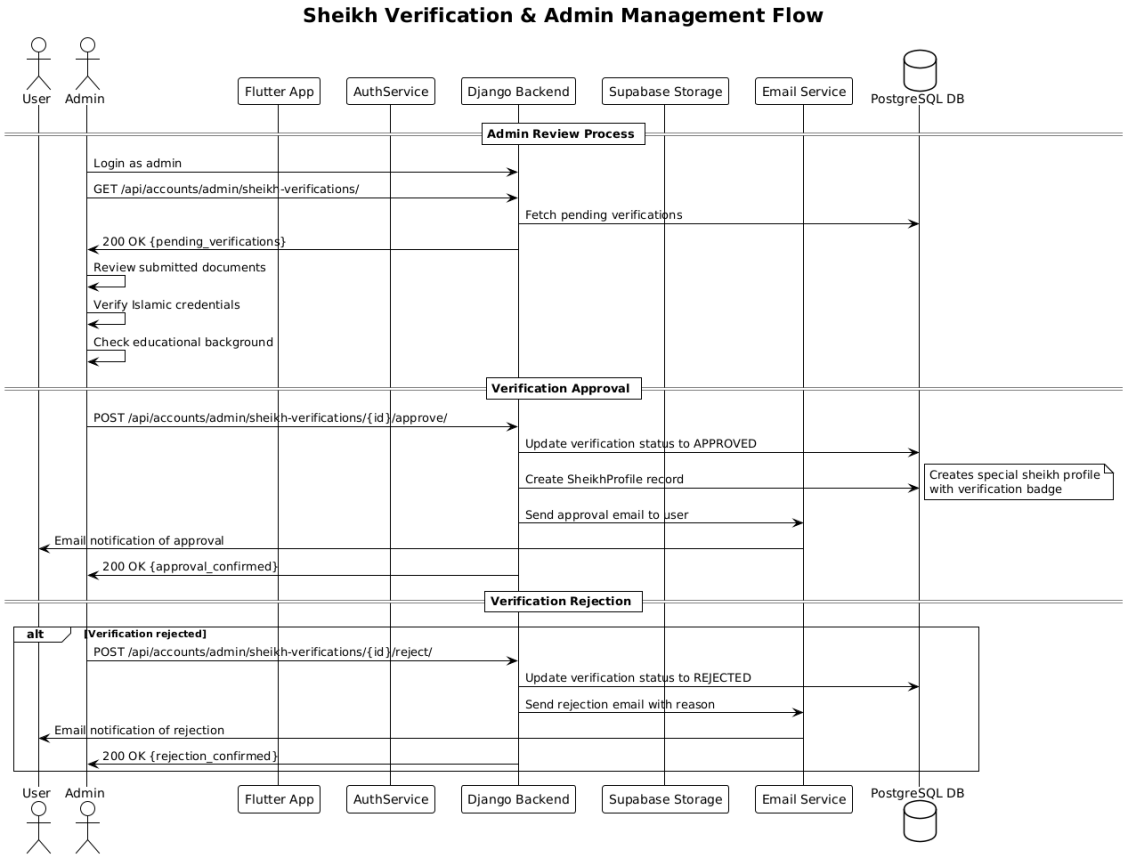

Figure 4.6: Sheikh Verification Admin Review Process

The admin review process shows how administrators can approve or reject verification requests. Upon approval, the system creates special sheikh profiles with verification badges and sends email notifications. Rejected requests include explanatory messages to help users understand the requirements.

4.3.3 RAG AI Assistant and Religious Guidance

The AI-powered chatbot provides Islamic guidance through a sophisticated RAG (Retrieval-Augmented Generation) system. Figure 4.7 demonstrates the query processing workflow.

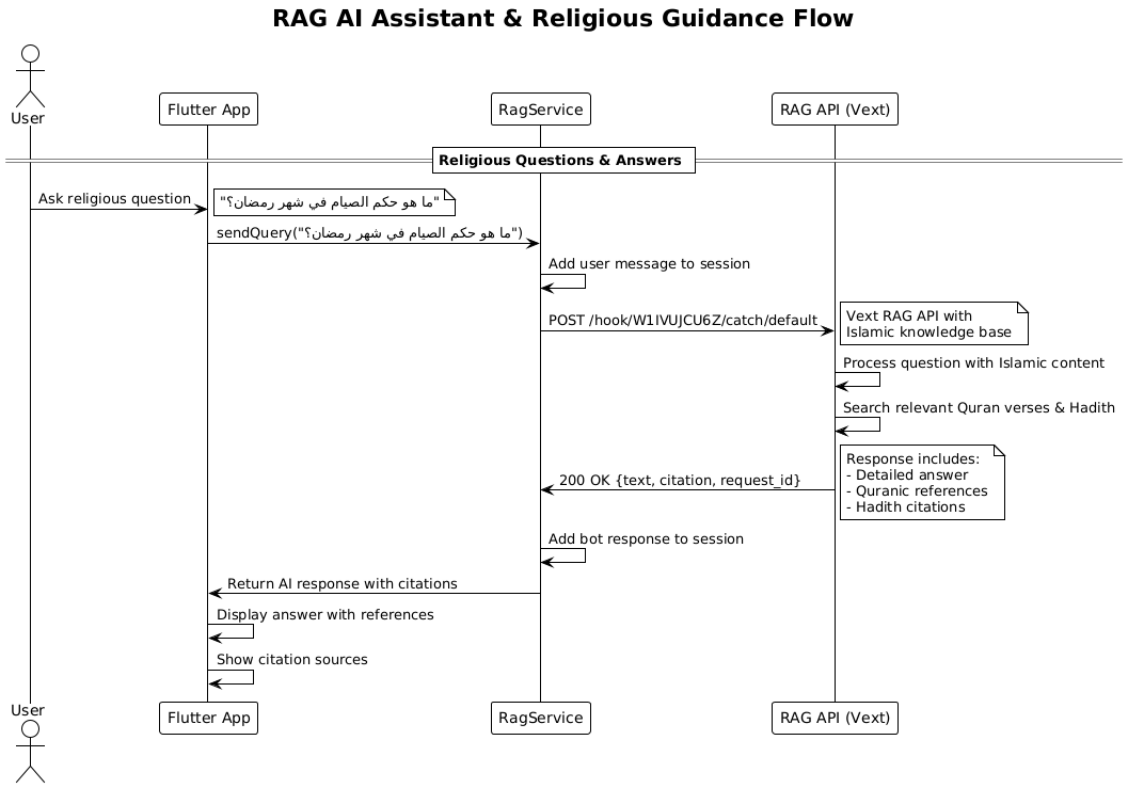

Figure 4.7: RAG AI Assistant Query Processing Flow

The RAG assistant sequence shows the comprehensive AI pipeline for processing Islamic queries. When users ask religious questions in Arabic or English, the system processes the query through the Vext RAG API, which searches relevant Quran verses and Hadith citations. The response includes detailed answers with Quranic references and Hadith citations, ensuring authentic religious guidance with proper source attribution.

4.3.4 Quran Reading and Study Features

The Quran study features provide enhanced reading experiences with progress tracking and bookmark management. Figure 4.8 shows the reading progress tracking system.

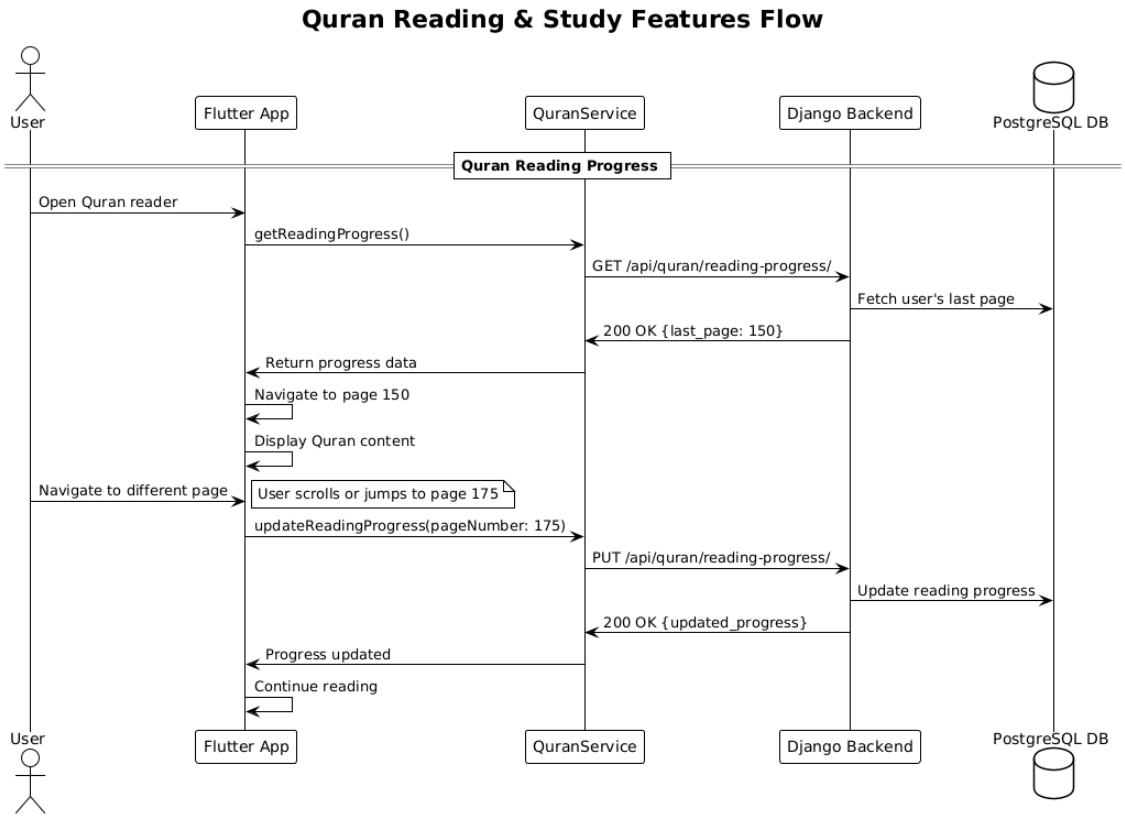

Figure 4.8: Quran Reading Progress Tracking Flow

The reading progress sequence demonstrates how the system tracks user reading sessions. When users open the Quran reader, the system retrieves their last read page and enables seamless continuation. As users navigate through pages, the system updates their progress, allowing for consistent reading experiences across sessions and devices.

Figure 4.9 illustrates the bookmark management system that allows users to save and organize their favorite verses.

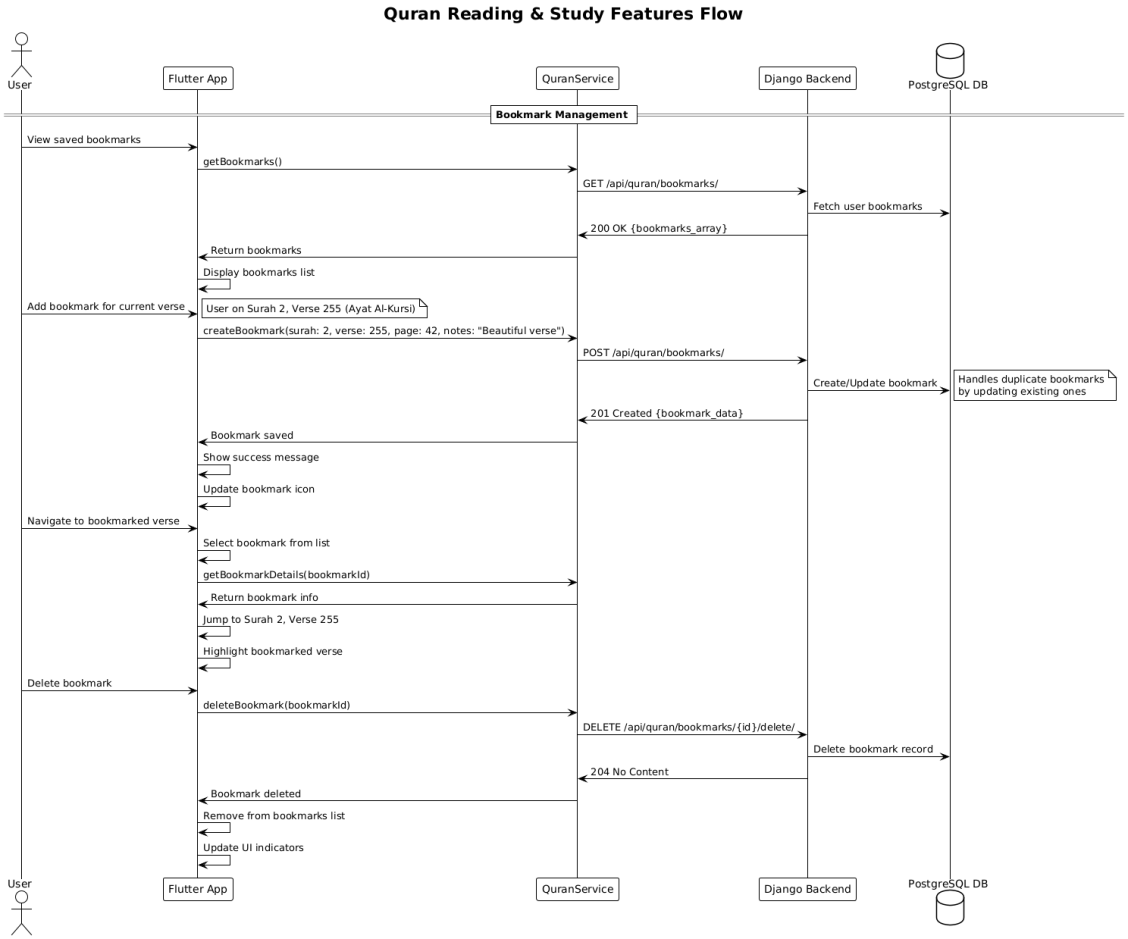

Figure 4.9: Quran Bookmark Management Flow

The bookmark management sequence shows the comprehensive bookmark system. Users can view saved bookmarks, add new bookmarks with personal notes, navigate to bookmarked verses, and delete unwanted bookmarks. The system handles duplicate bookmarks by updating existing ones and provides detailed bookmark information including Surah, verse, and page references.

4.3.5 Push Notification and Messaging System

The platform implements a comprehensive notification system using Firebase Cloud Messaging. Figure 4.10 shows the notification trigger mechanisms.

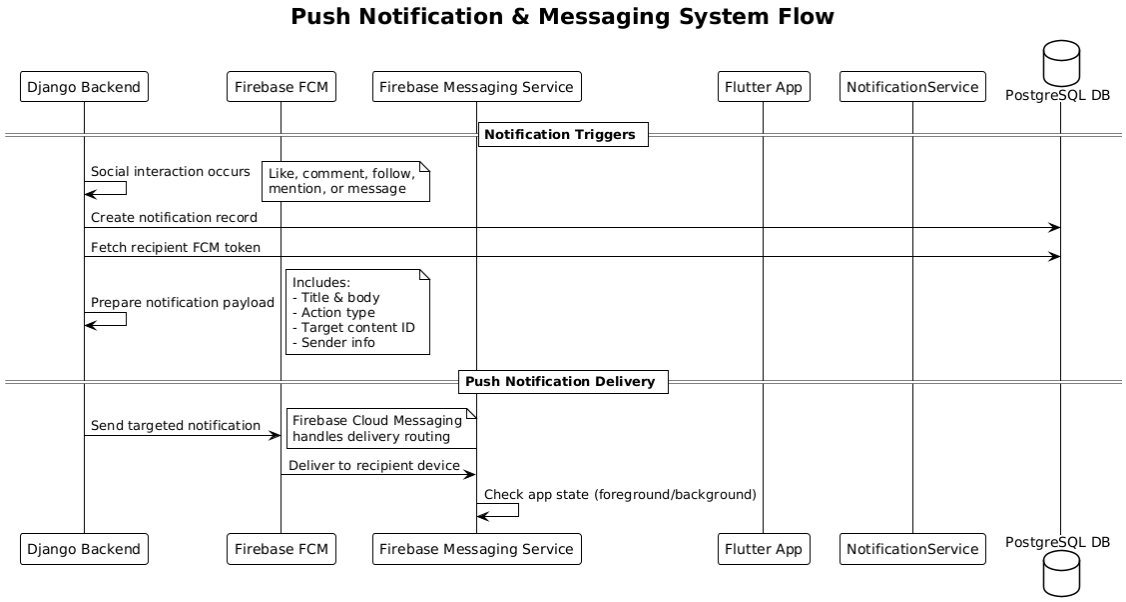

Figure 4.10: Push Notification Trigger System

The notification trigger system demonstrates how social interactions generate push notifications. The system creates notification records, fetches recipient FCM tokens, prepares notification payloads with titles, bodies, and action types, and delivers targeted notifications through Firebase Cloud Messaging.

Figure 4.11 illustrates how the application handles incoming notifications in both foreground and background states.

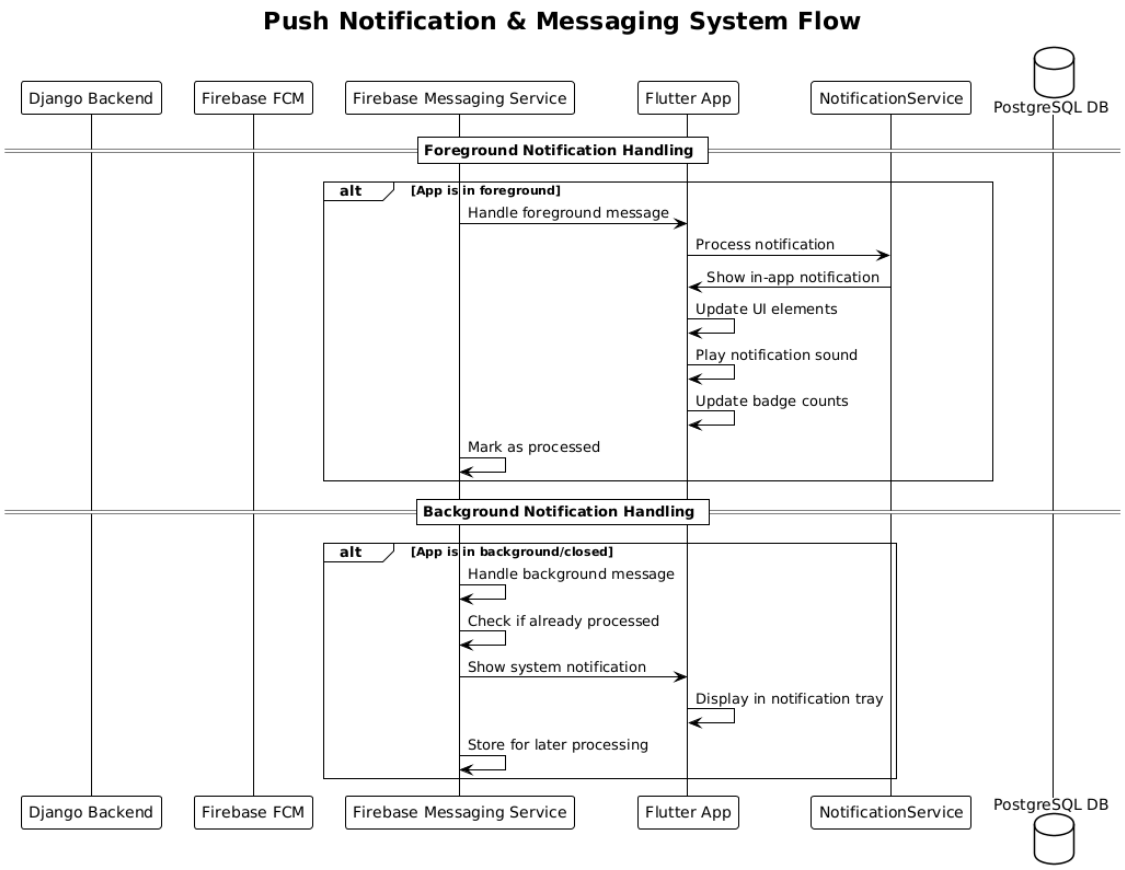

Figure 4.11: Push Notification Handling Flow

The notification handling sequence shows the platform's response to incoming notifications. In foreground mode, the system processes notifications immediately, shows in-app notifications, updates UI elements, and plays notification sounds. In background mode, the system displays system notifications in the notification tray and stores them for later processing when the app becomes active.

4.3.6 Chat and Messaging System

The real-time messaging system enables direct communication between users. Figure 4.12 shows the conversation setup process.

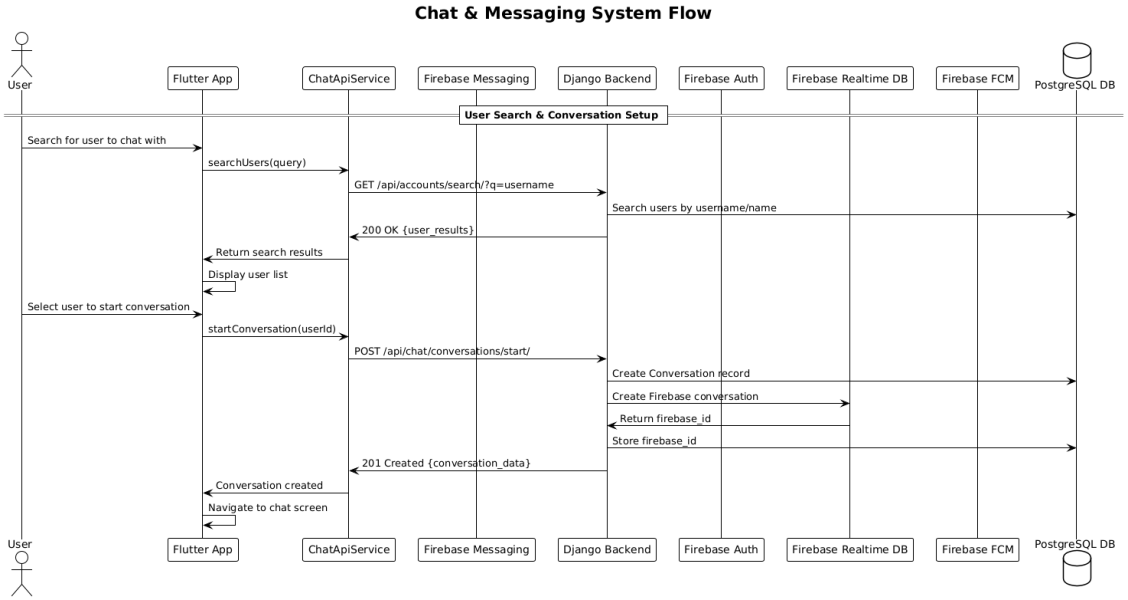

Figure 4.12: Chat Conversation Setup Flow

The chat setup sequence demonstrates how users initiate conversations. Users search for other users, select recipients, and the system creates conversation records with corresponding Firebase realtime database entries. This establishes the foundation for real-time messaging between users.

Figure 4.13 illustrates the real-time messaging flow and message status management.

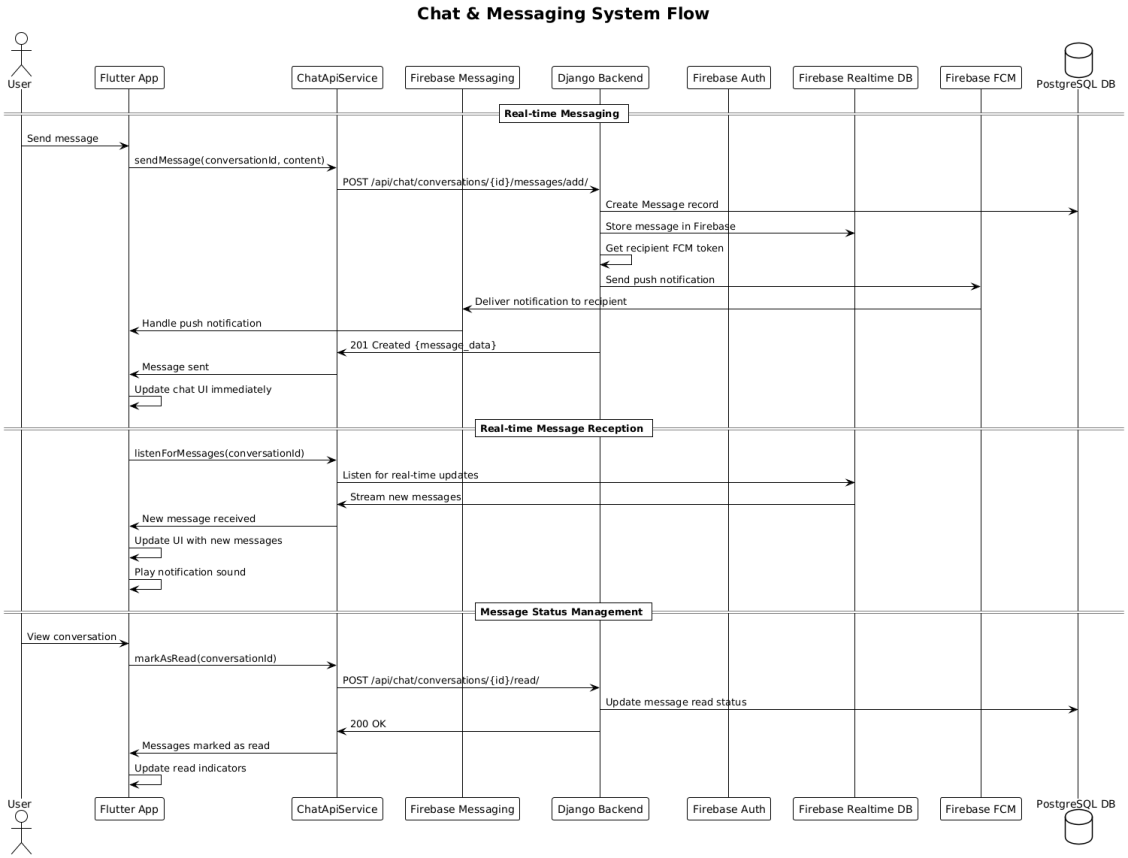

Figure 4.13: Real-time Chat Messaging Flow

The real-time messaging sequence shows the comprehensive messaging system. When users send messages, the system creates message records, stores them in Firebase, retrieves recipient FCM tokens, and sends push notifications. The system also handles real-time message reception through Firebase listeners and manages message read status tracking.

Figure 4.14 demonstrates the Firebase authentication process required for real-time chat functionality.

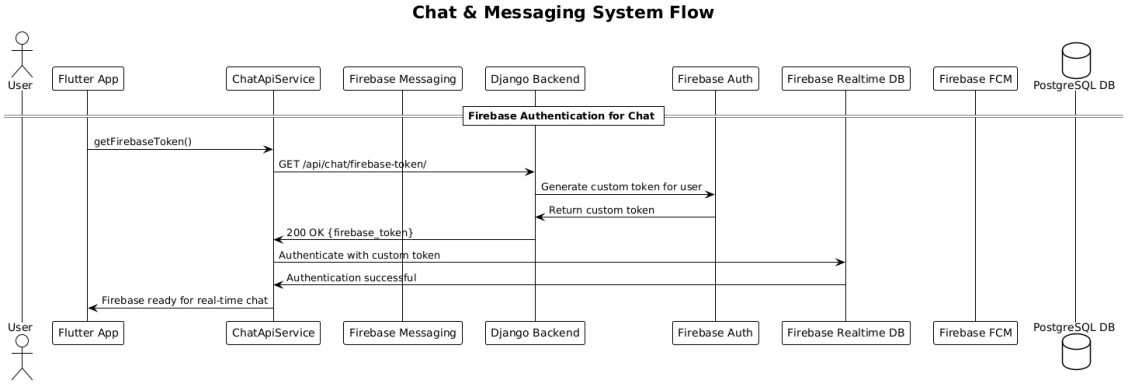

Figure 4.14: Firebase Authentication for Chat System

The Firebase authentication sequence shows how the system generates custom tokens for

chat authentication. This enables secure access to Firebase Realtime Database for messaging while maintaining user identity and permissions.

4.3.7 Authentication and User Management

The authentication system provides comprehensive user management with JWT token-based security. Figure 4.15 shows the token refresh mechanism.

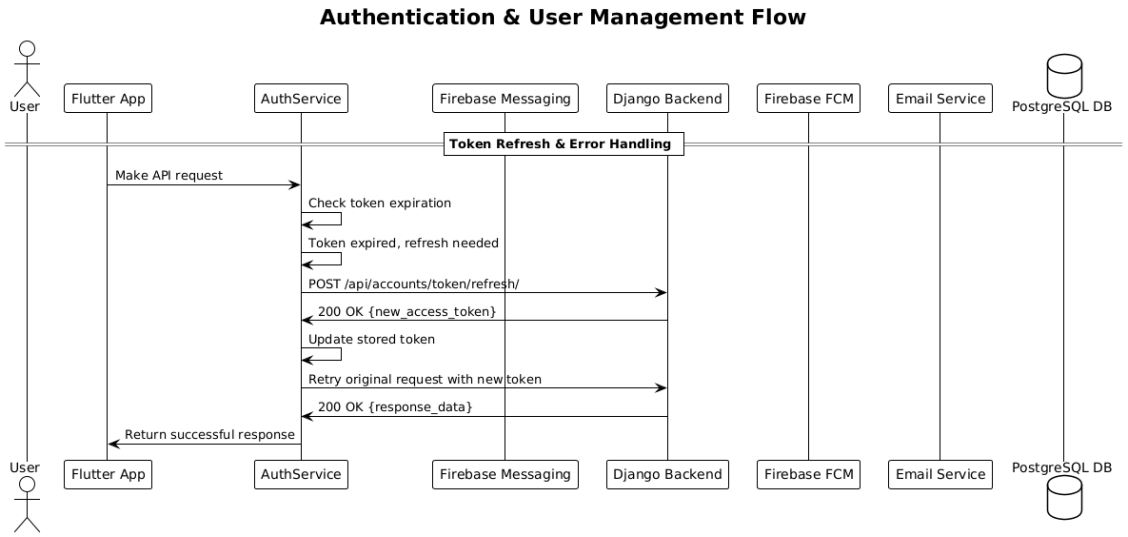

Figure 4.15: Authentication Token Refresh Flow

The token refresh sequence demonstrates the system's automatic token management. When API requests detect expired tokens, the system automatically refreshes them using refresh tokens, updates stored credentials, and retries the original request seamlessly, maintaining user sessions without interruption.

Figure 4.16 illustrates the password reset functionality with email verification.

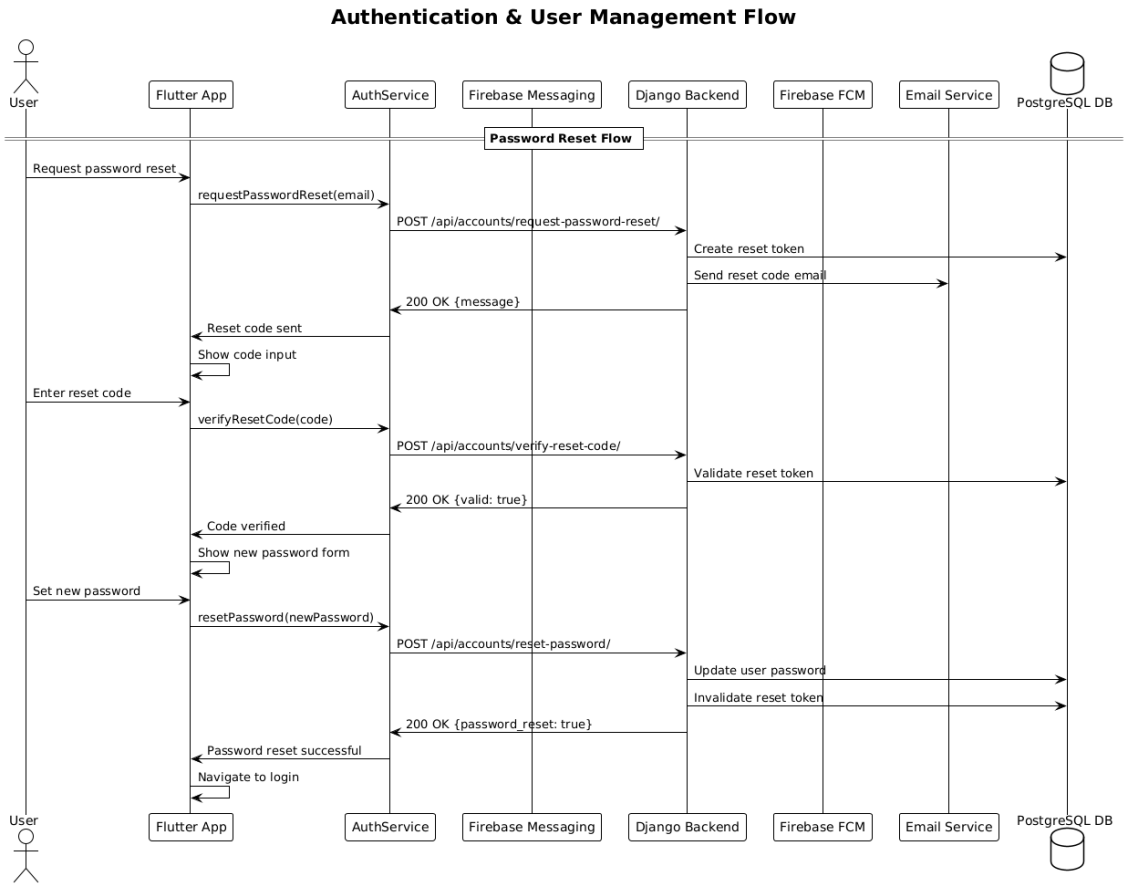

Figure 4.16: Password Reset Flow

The password reset sequence shows the secure password recovery process. Users request password resets, receive verification codes via email, validate the codes, and set new passwords. The system invalidates reset tokens after use to maintain security.

Figure 4.17 demonstrates the user registration process with email verification.

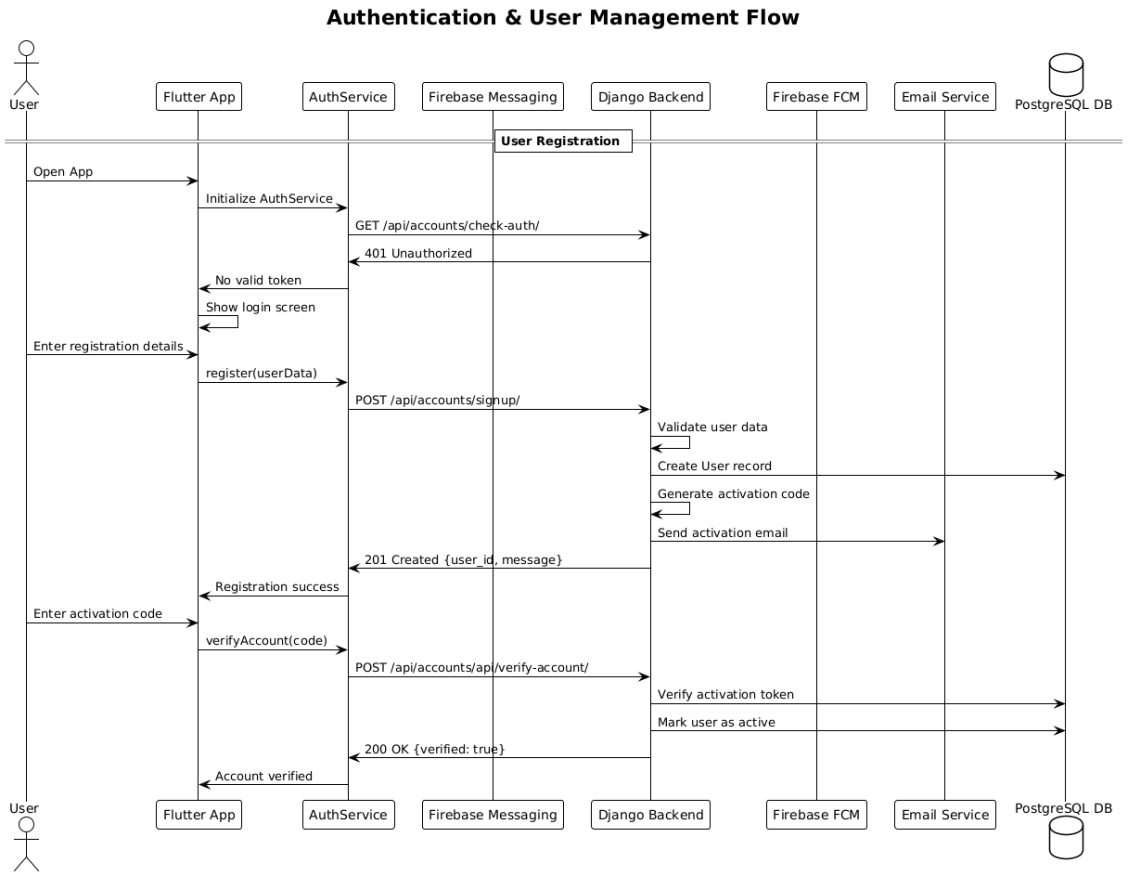

Figure 4.17: User Registration Flow

The registration sequence shows the comprehensive user onboarding process. New users provide registration details, receive activation codes via email, verify their accounts, and gain access to the platform. This ensures email validity and prevents unauthorized account creation.

Figure 4.18 illustrates profile viewing and updating functionality.

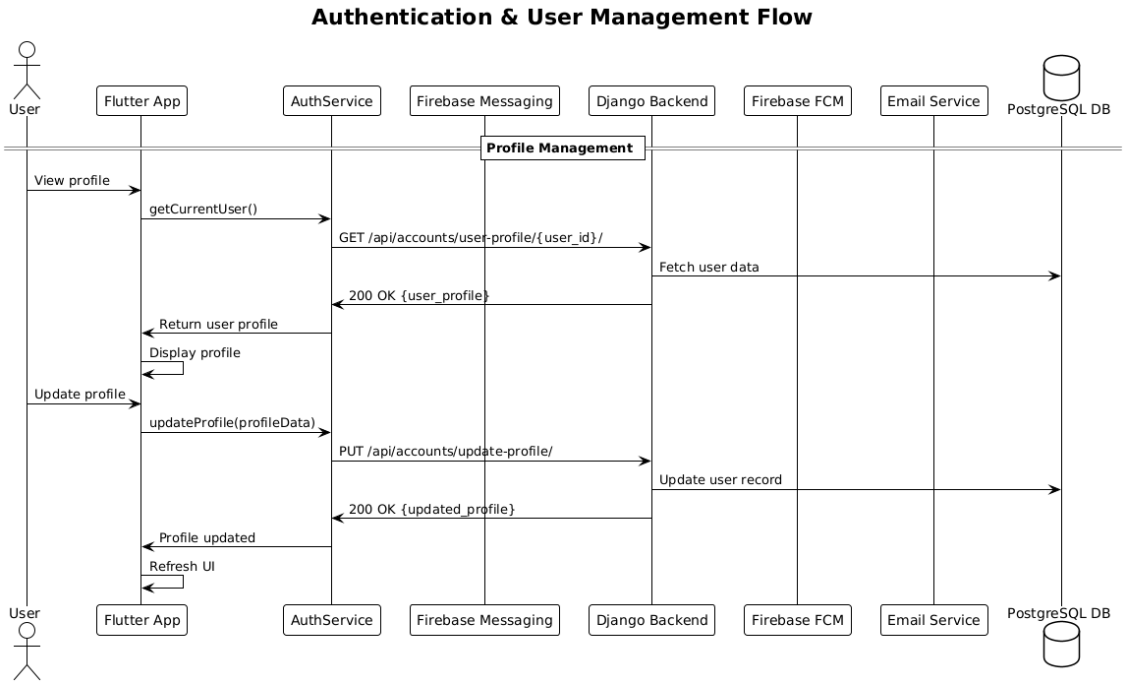

Figure 4.18: Profile Management Flow

The profile management sequence demonstrates how users can view and update their profile information. The system fetches current user data, displays profile information, and processes profile updates with immediate UI refresh to reflect changes.

Figure 4.19 shows the logout process with token management.

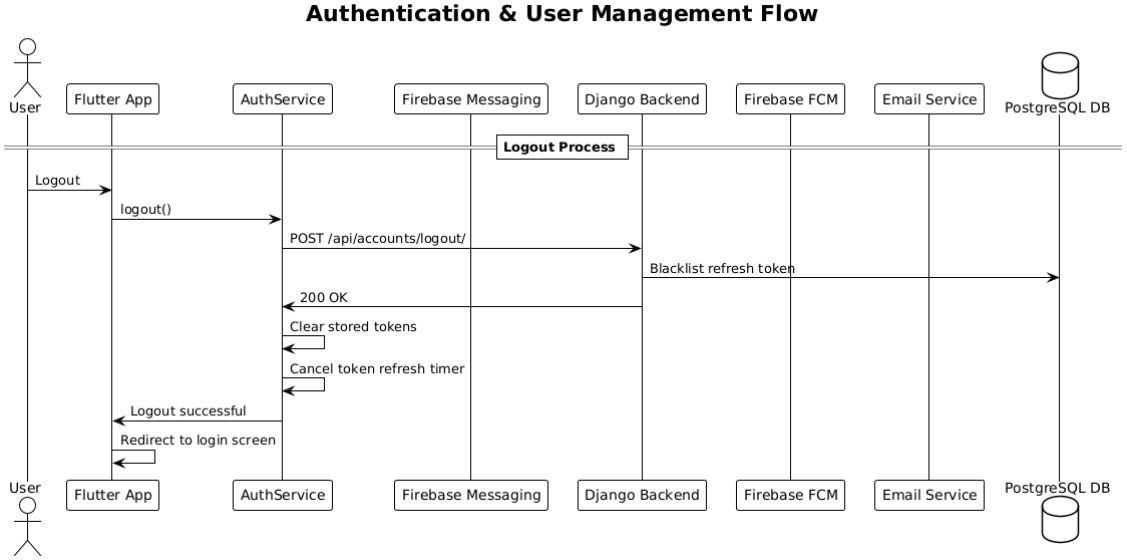

Figure 4.19: User Logout Flow

The logout sequence illustrates the secure session termination process. The system blacklists refresh tokens, clears stored credentials, cancels token refresh timers, and redirects users

to the login screen, ensuring complete session cleanup.

Figure 4.20 demonstrates the login process with JWT token generation.

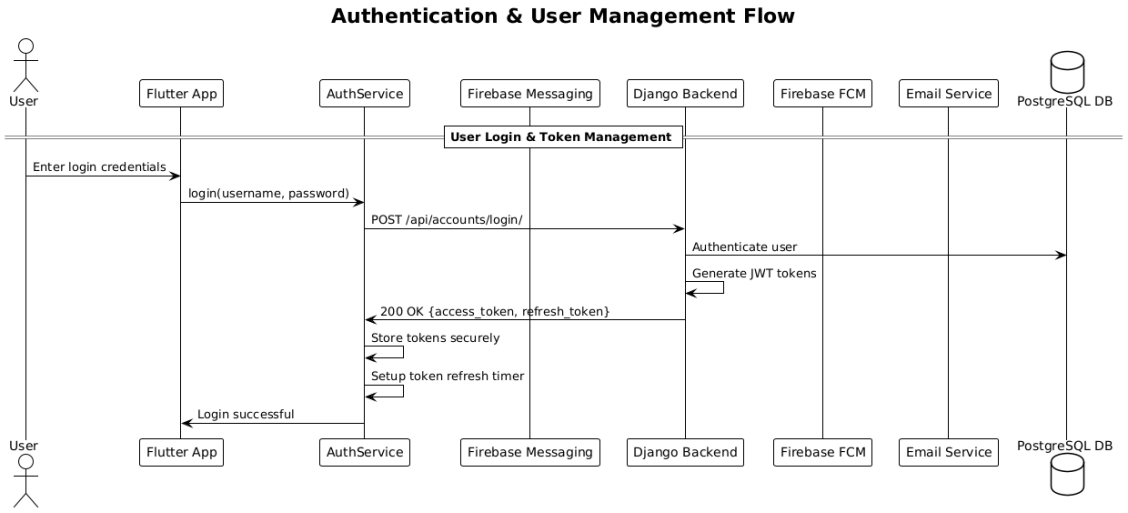

Figure 4.20: User Login Flow

The login sequence shows the authentication process where users provide credentials, the system authenticates them, generates JWT tokens, and establishes secure sessions with automatic token refresh mechanisms.

Figure 4.21 illustrates FCM token registration for push notifications.

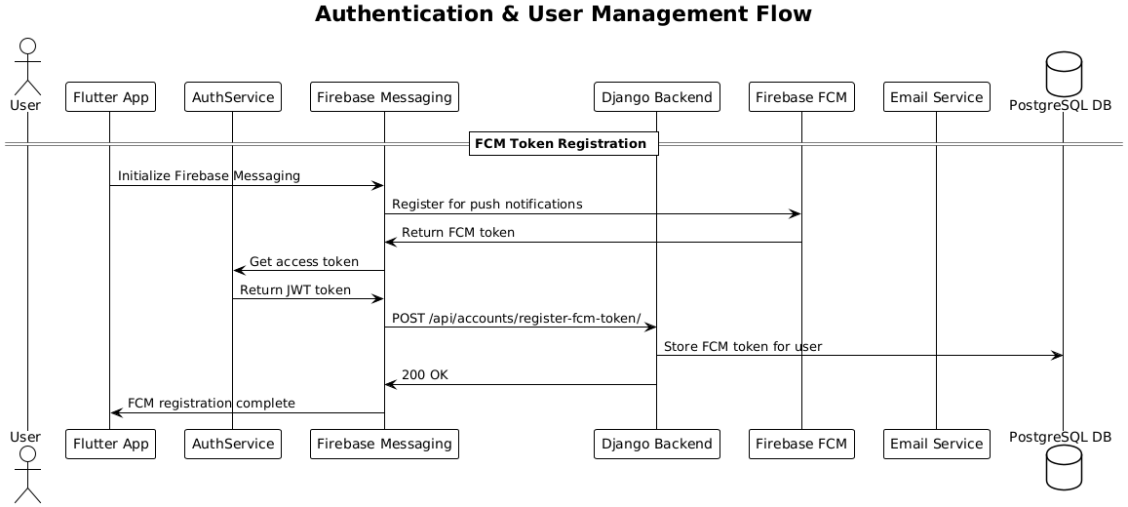

Figure 4.21: FCM Token Registration Flow

4.4 Database Design

The database design for the Zikr platform follows a normalized relational model that supports the complex relationships between users, social content, AI-generated data, and Islamic educational materials. Figure 4.22 presents the complete Entity-Relationship Diagram (ERD) that illustrates the database structure and relationships.

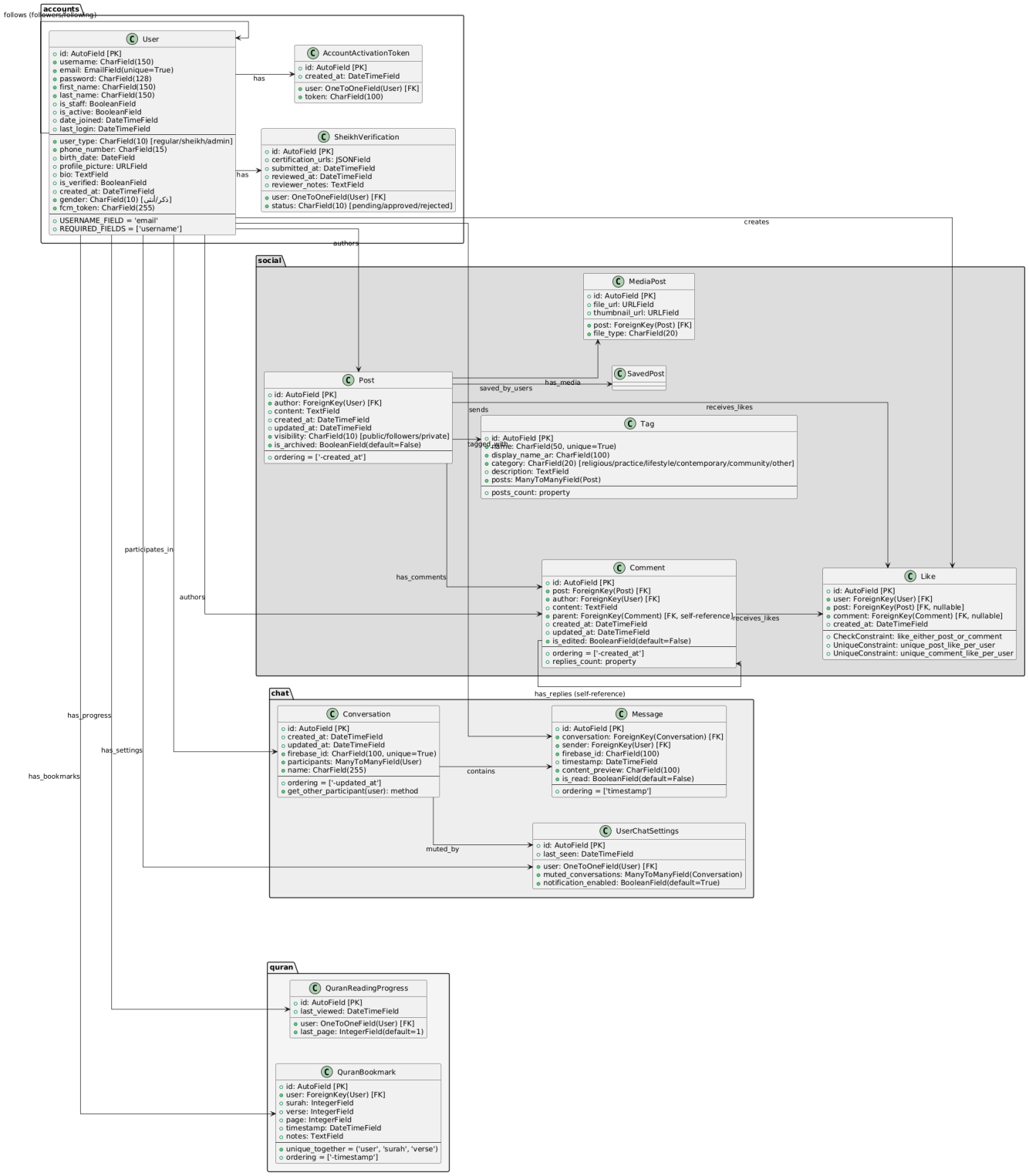

Figure 4.22: Database UML Diagram for Zikr Platform

Chapter 5

Results

This chapter presents the comprehensive results achieved in the development of Zikr, an innovative AI-powered Islamic platform. The results demonstrate the successful integration of advanced machine learning techniques with social community features, creating a holistic digital environment for Islamic learning and community engagement. The platform combines three core AI components: a Retrieval-Augmented Generation (RAG) chatbot for Islamic Fatawa questions, a K-Means clustering model for Quranic topic highlighting, and a fuzzy logic-based content recommendation system for personalized social feeds.

5.1 Overview of Completed Features

The Zikr platform successfully delivers a comprehensive suite of features that address the identified objectives through two main development tracks: application development and AI integration. The following subsections detail the major accomplishments and completed components.

5.1.1 Core Application Infrastructure

The platform architecture follows a modern client-server paradigm with clear separation between frontend and backend components. The Flutter frontend provides consistent user experience across mobile and web platforms, utilizing a component-based architecture with Provider pattern state management. The Django REST Framework backend delivers a scalable API layer with JWT-based authentication, comprehensive user management, and real-time messaging capabilities through Firebase Cloud Messaging.

5.1.2 User Authentication and Profile Management

A robust authentication system was implemented supporting multiple login methods including email/password registration, social login integration, and a specialized sheikh verification system. The profile management system enables users to customize their experience, track reading progress, manage bookmarks, and establish their religious authority status through a comprehensive verification workflow that includes document upload capabilities and administrative review processes.

5.1.3 Social Community Platform

The social networking framework enables users to create posts, comment, like, and interact within a safe and moderated environment. The system implements hierarchical commenting

structures, content tagging for enhanced discovery, and sophisticated content filtering based on user preferences and community standards. The platform maintains Islamic principles while fostering meaningful religious discussions through carefully designed interaction guidelines and automated moderation systems.

5.1.4 Digital Quran Interface

A comprehensive digital Quran module was developed featuring page-by-page navigation, verse lookup capabilities, reading progress tracking, and bookmark management. The interface supports multiple reciters for audio recitation with synchronized text highlighting and integrates AI-powered topic highlighting through the K-Means clustering model. Users can seamlessly navigate through 604 pages with smooth transitions and offline reading capabilities.

5.1.5 Real-time Communication System

The platform includes a sophisticated chat system supporting both direct messaging and group conversations. The messaging system integrates seamlessly with the AI-powered RAG chatbot, enabling users to receive religious guidance within their conversation flows. Real-time message delivery is ensured through WebSocket connections and Firebase Cloud Messaging for offline notification delivery.

5.2 AI Model Implementation Results

5.2.1 RAG-based Islamic Fatawa Chatbot

The Retrieval-Augmented Generation chatbot was successfully implemented to answer Islamic Fatawa questions in both Arabic and English. The system utilizes a comprehensive dataset scraped from IslamQA, containing thousands of authentic religious questions and answers. The chatbot demonstrates high accuracy in retrieving relevant religious content and generating contextually appropriate responses while maintaining religious authenticity.

The RAG system architecture includes:

- Vector database for efficient semantic search across Islamic content
- Context-aware response generation maintaining religious accuracy
- Source citation mechanisms for transparency and verification
- Multilingual support for Arabic and English queries
- Confidence scoring for answer reliability assessment

5.2.2 Quranic Topic Modeling and Visualization

The K-Means clustering model for Quranic topic identification achieved successful segmentation of verses into 20 distinct thematic clusters. The model was trained on Tafseer Al-Saadi commentary, processed through comprehensive Arabic NLP preprocessing including diacritical mark removal, stemming using Stanza pipeline, and stopword filtering with a curated Arabic stopwords list.

Key achievements of the topic modeling system:

- Successful identification of 20 distinct Quranic themes through K-Means clustering

- Optimal cluster count determination using elbow method and silhouette score analysis
- TF-IDF vectorization with 2,151 features optimized for Arabic religious text
- Color-coded visualization system enhancing user reading experience
- Integration with digital Quran interface for real-time topic highlighting

5.2.3 Fuzzy Logic Content Recommendation System

A personalized content suggestion algorithm was implemented using fuzzy logic principles to recommend posts based on user interests, activity patterns, and engagement history. The system balances personalization with content diversity to ensure users are exposed to both relevant and varied Islamic topics, fostering comprehensive religious learning.

The recommendation system incorporates:

- User interest profiling based on interaction patterns
- Content diversity metrics to prevent filter bubbles
- Engagement history analysis for personalization
- Fuzzy logic rules for balanced recommendation generation
- Real-time adaptation based on user feedback

5.2.4 Arabic NLP Processing Results

The Arabic natural language processing pipeline successfully handled the complexity of classical Arabic religious texts. The Stanza-based lemmatization achieved effective normalization of varied Arabic text forms, while the custom stopword filtering preserved meaningful religious terminology. The TF-IDF vectorization with 2,151 optimized features demonstrated effective dimensionality for religious content analysis.

5.2.5 Model Training Outcomes

The K-Means clustering model achieved optimal performance with 20 clusters as determined through elbow method analysis and silhouette score evaluation. The model successfully identified meaningful thematic groupings that align with traditional Islamic scholarship categorizations. The RAG system demonstrated effective retrieval and generation capabilities for Islamic content queries.

5.3 Final Application Screenshots

5.3.1 Mobile App UI

The Zikr mobile application demonstrates a comprehensive user interface design that successfully integrates all platform features while maintaining intuitive navigation and Islamic design principles. The following subsections showcase the key interface components and user interactions.

Authentication and Account Management

The authentication system provides multiple entry points for users while ensuring security and proper verification processes.

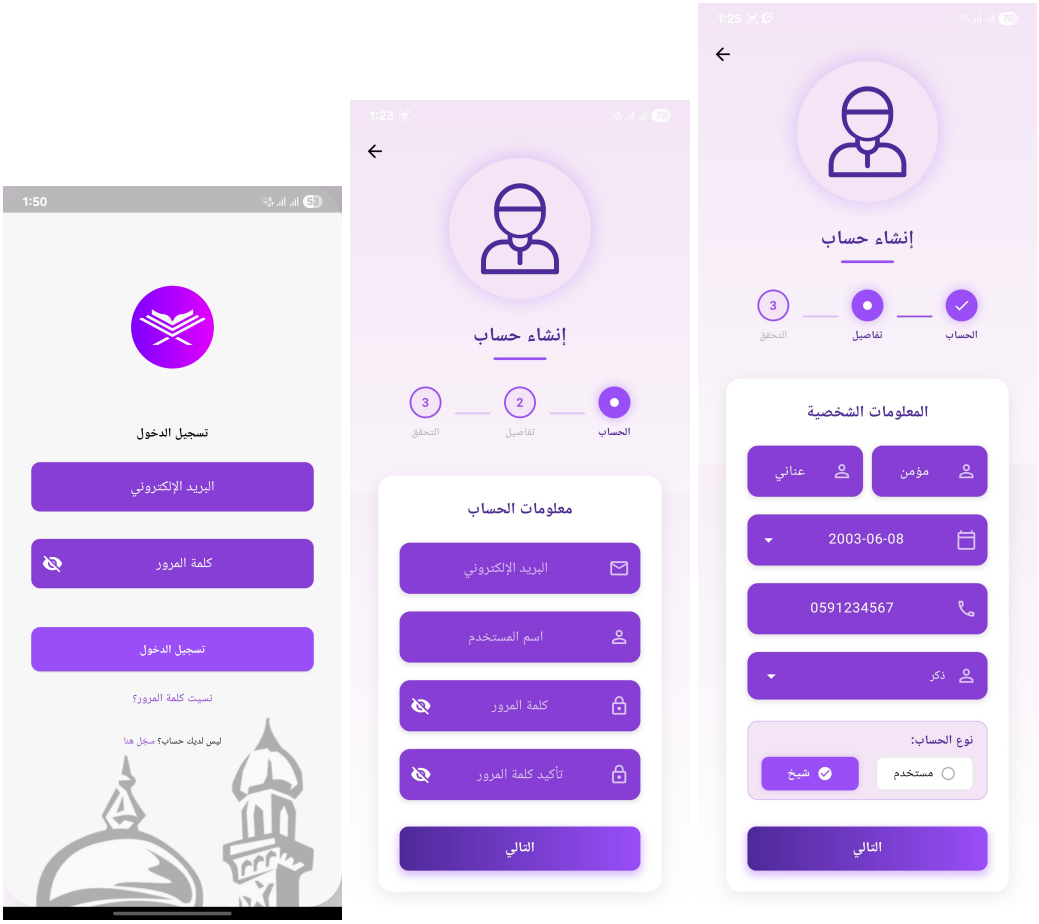

Figure 5.1: Sign in and account creation process

Figure 5.2: Sheikh verification and email confirmation

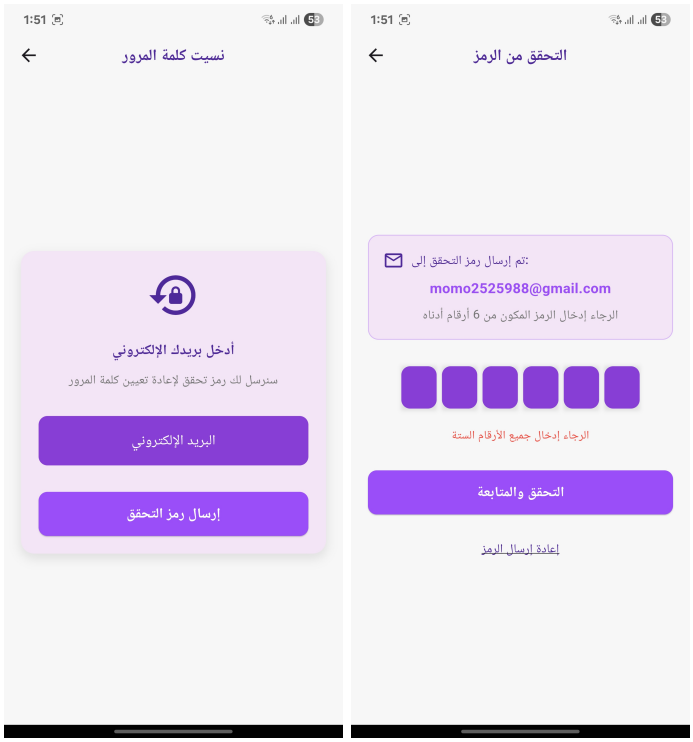

Figure 5.3: Password recovery workflow (1)

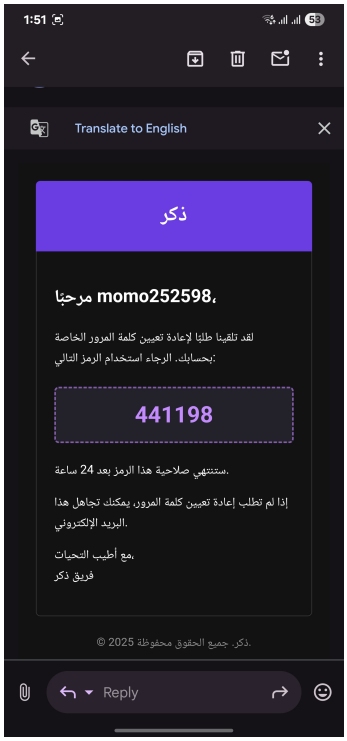

Figure 5.4: Password recovery workflow (2)

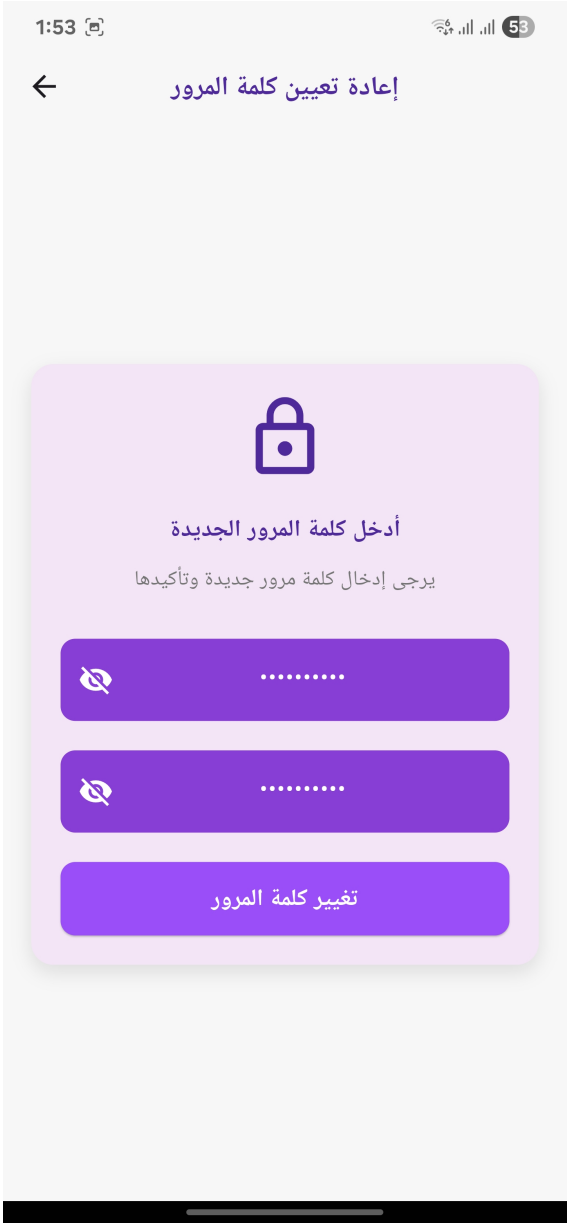

Figure 5.5: Password reset interface

Main Navigation and Home Interface

The main interface provides centralized access to all platform features with clean navigation design.

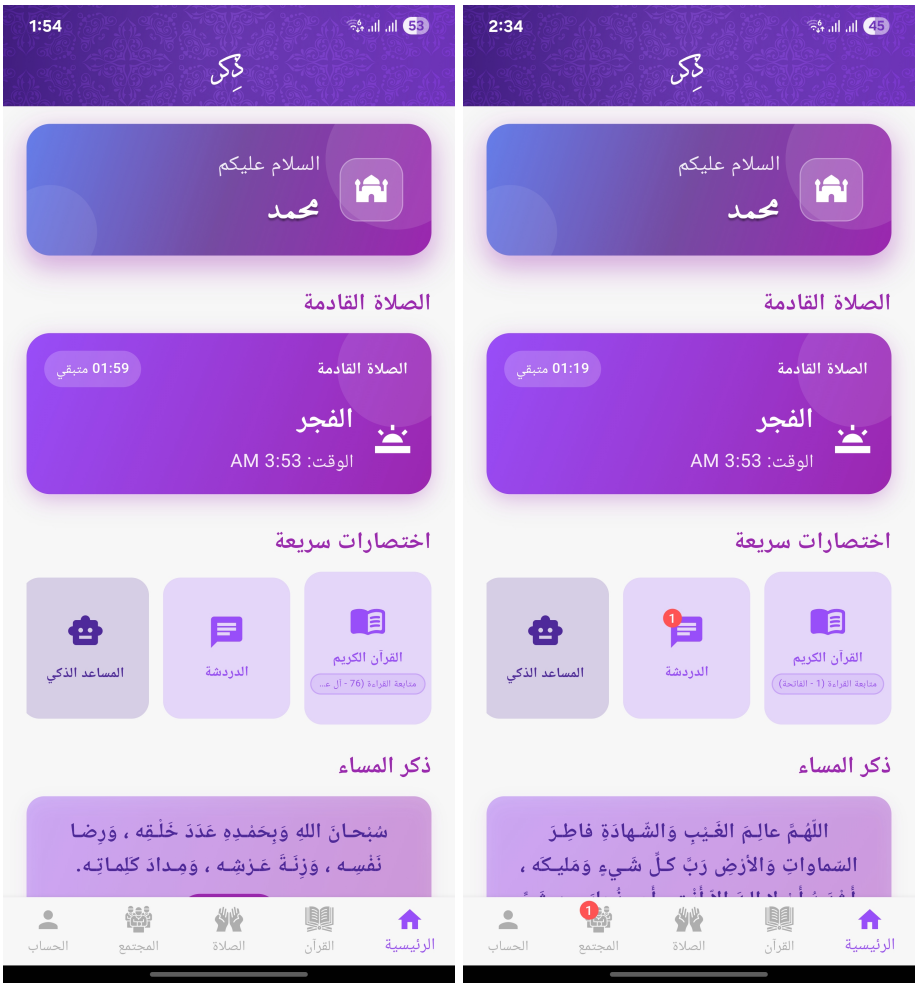

Figure 5.6: Home page with and without notifications

Social Feed and Content Management

The social networking features demonstrate the platform's community-focused design with content filtering and interaction capabilities.

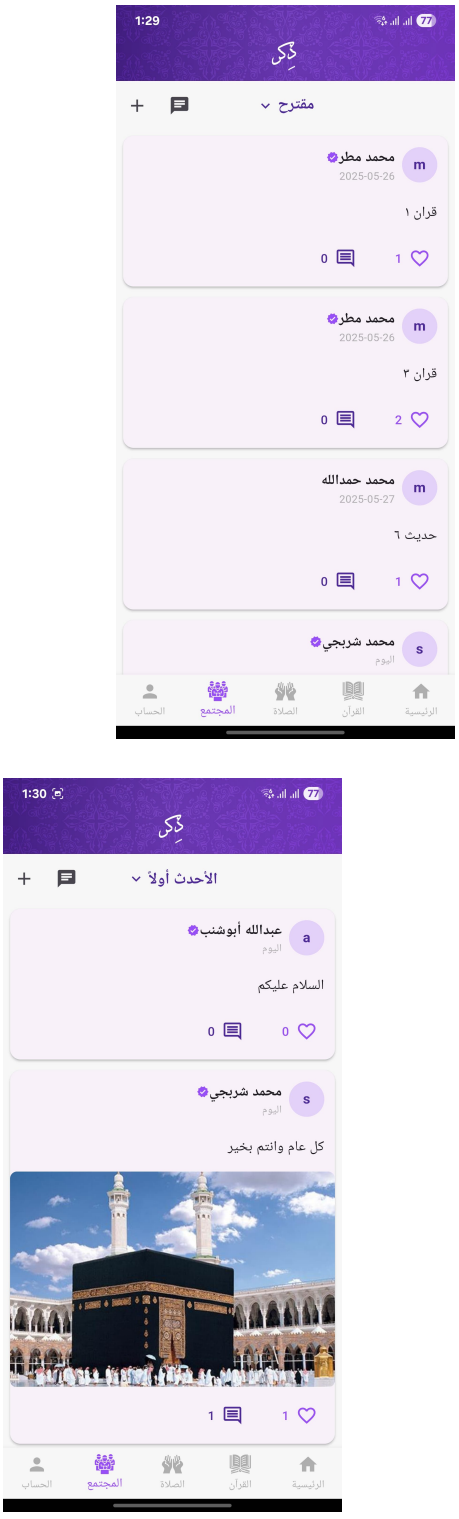

Figure 5.7: Posts feed with suggested (top) and recent (bottom) content views

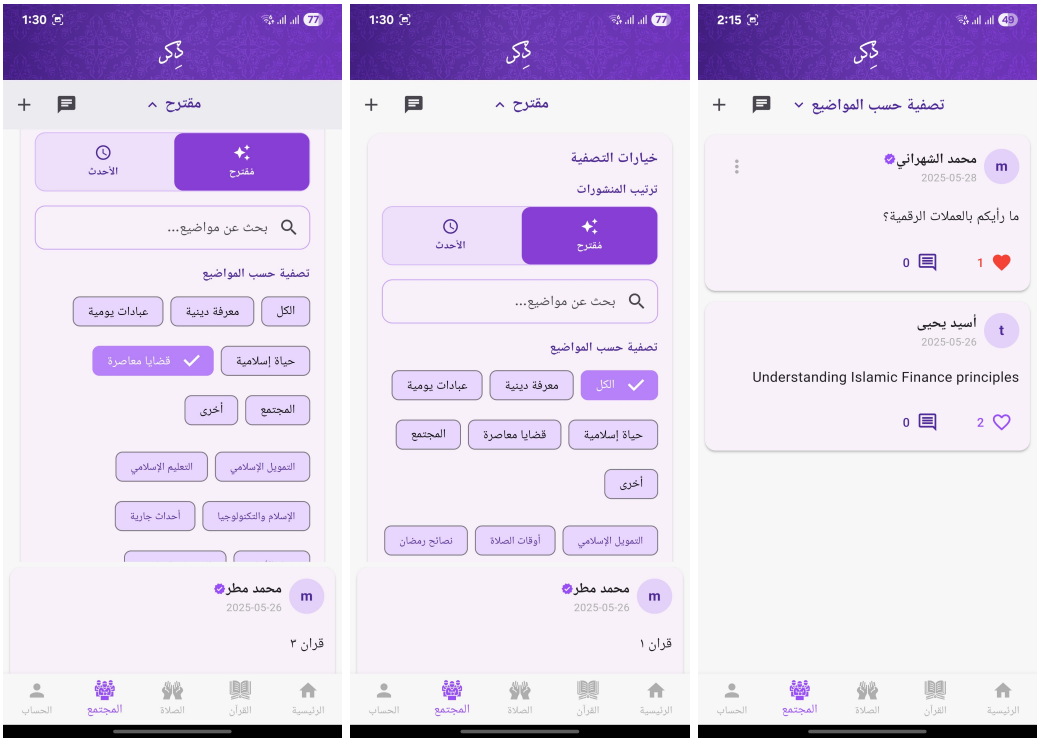

Figure 5.8: Topic-based content filtering system

Figure 5.9: Additional topic filtering options

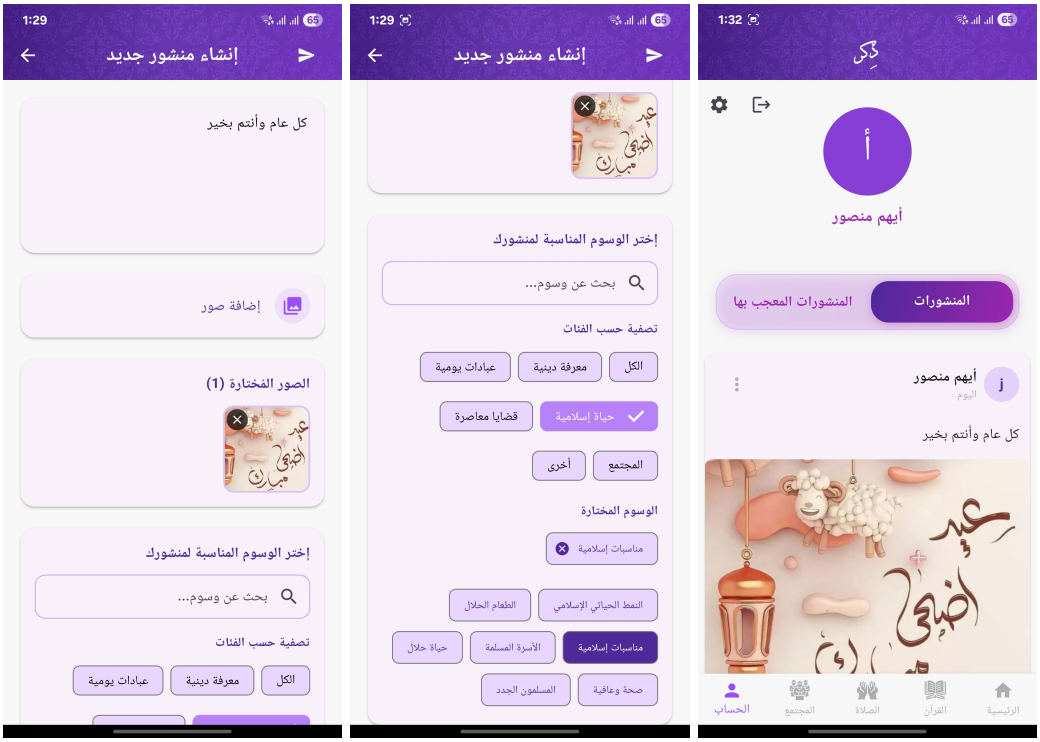

Figure 5.10: Post creation interface and personal posts management

Digital Quran Interface

The Quran module showcases advanced features including topic clustering, search functionality, and multimedia integration.

Figure 5.11: Quran verse highlighting

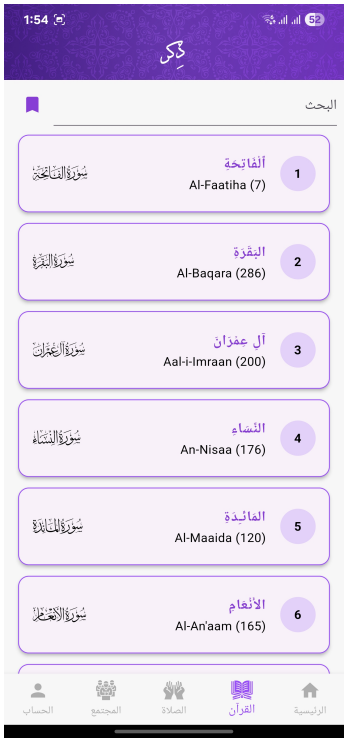

Figure 5.12: Quran navigation

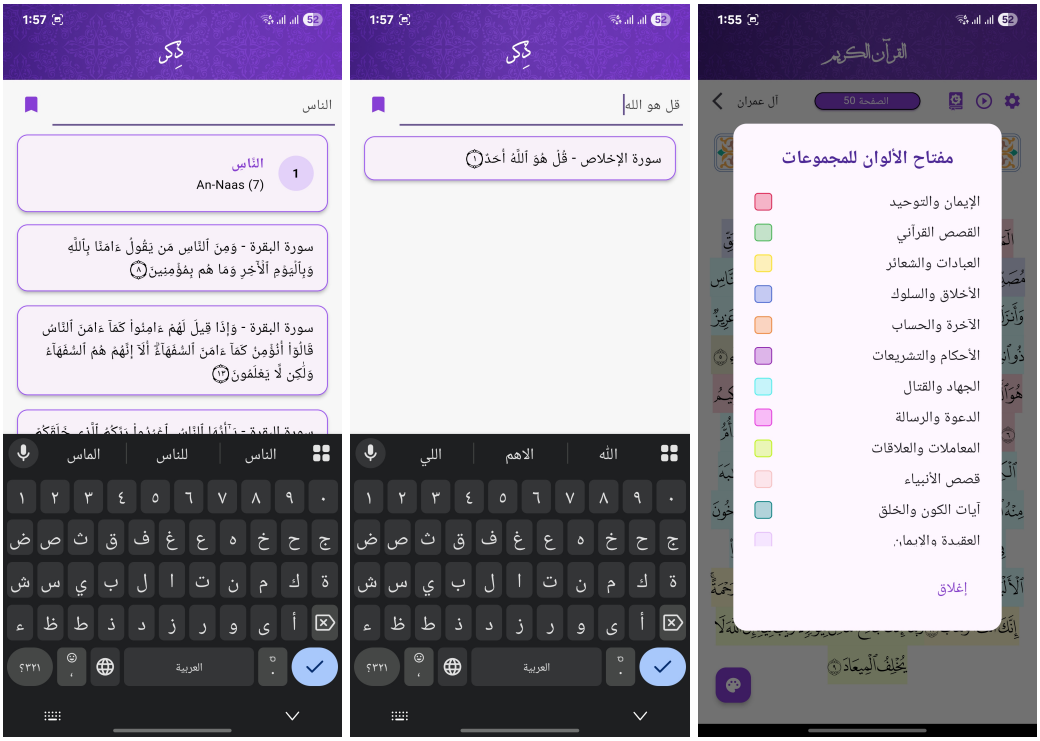

Figure 5.13: Verse search and topic clustering visualization

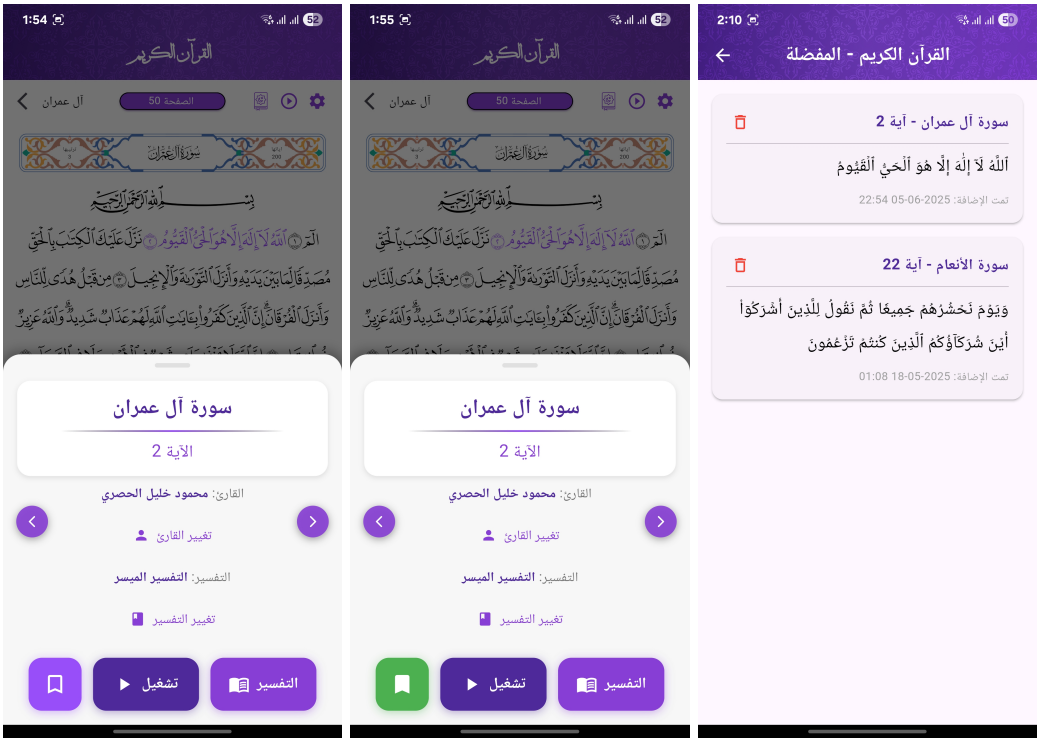

Figure 5.14: Verse details and bookmark management

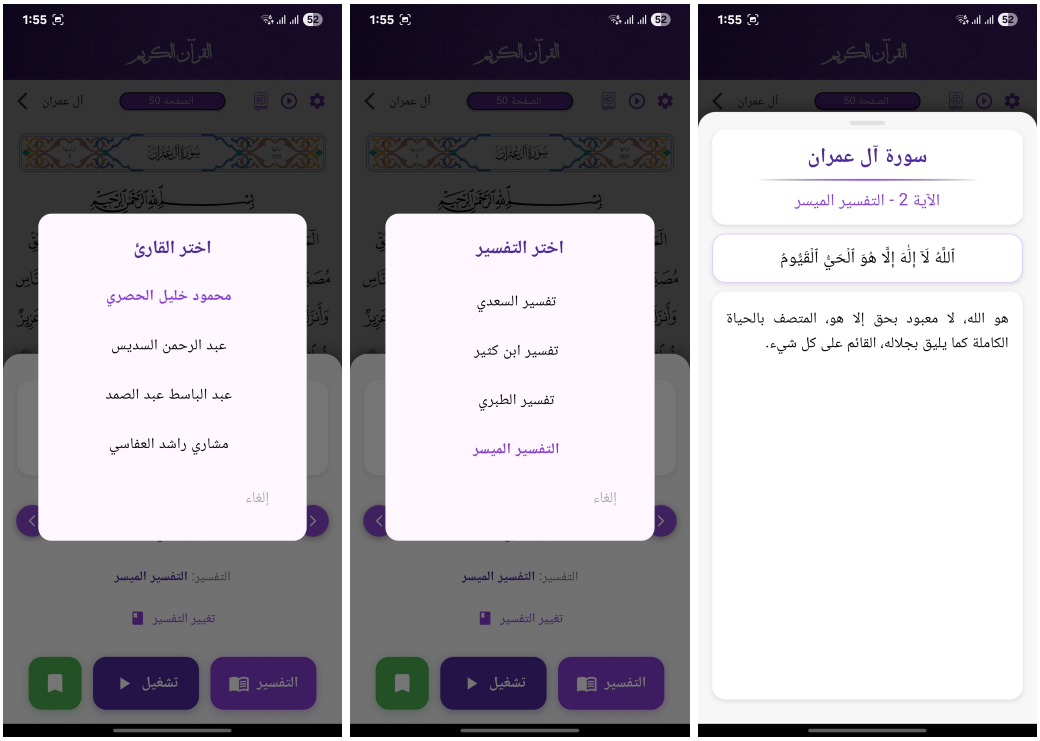

Figure 5.15: Audio recitation and Tafseer integration

Figure 5.16: Quran page reciter selection interface

AI-Powered RAG Chatbot

The Retrieval-Augmented Generation chatbot interface demonstrates the platform's AI capabilities for Islamic guidance.

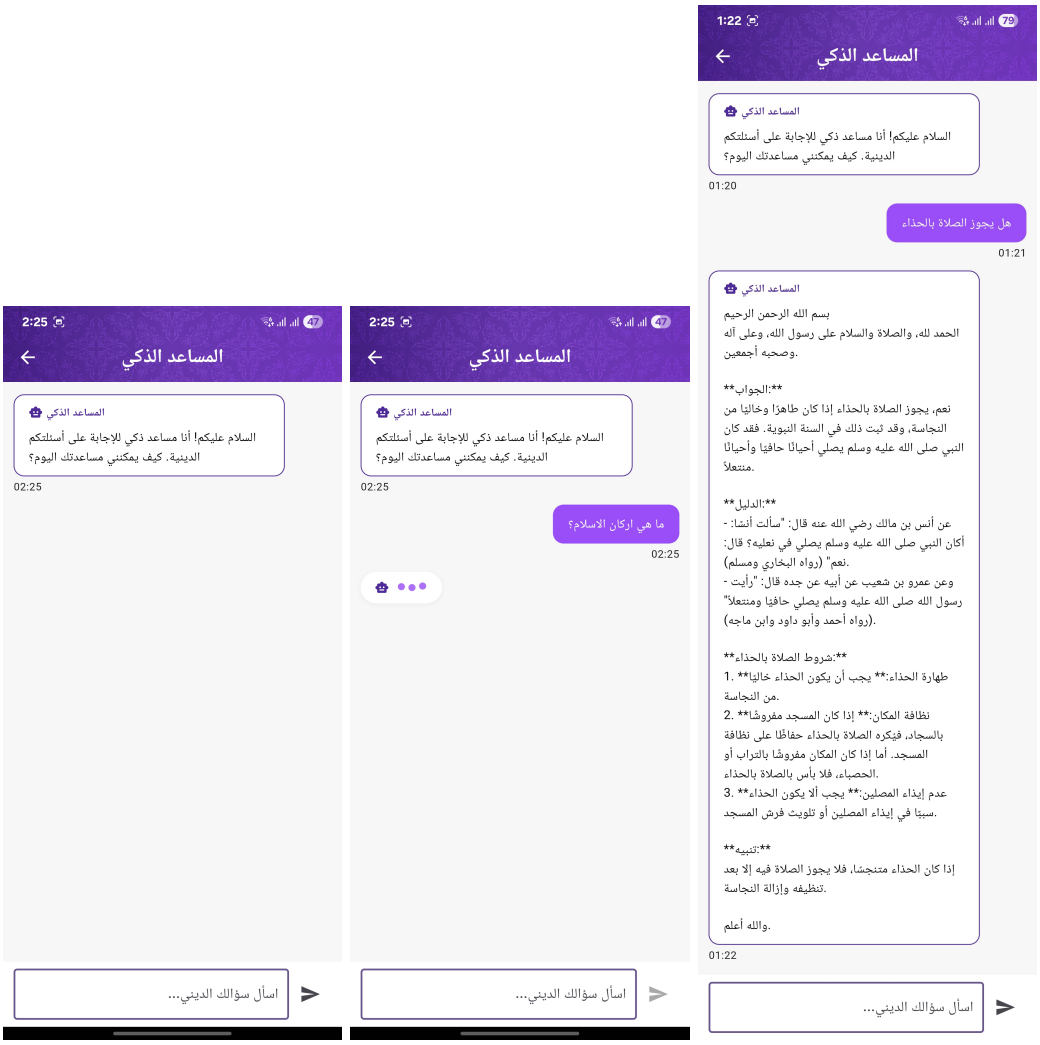

Figure 5.17: RAG chatbot conversation interface

Communication and Messaging

The messaging system provides both direct communication and integration with AI assistance.

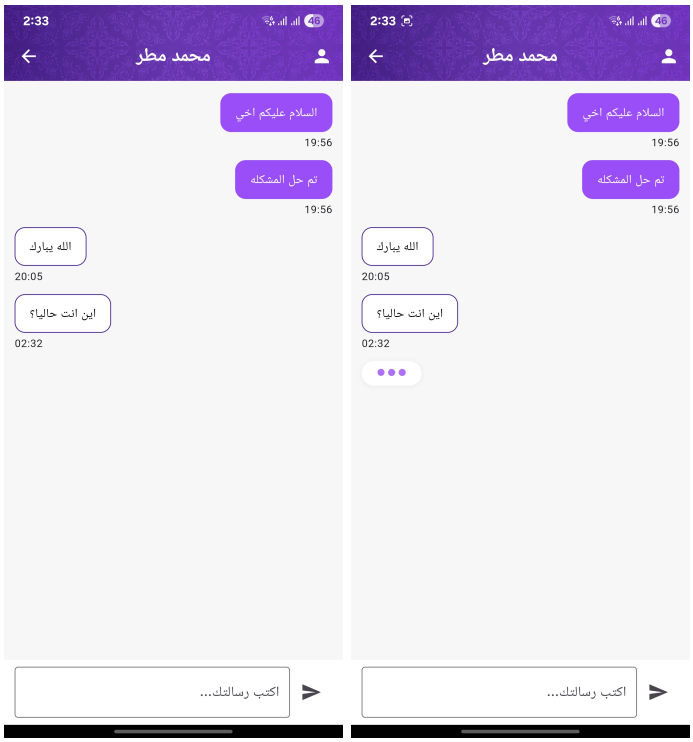

Figure 5.18: messaging interface

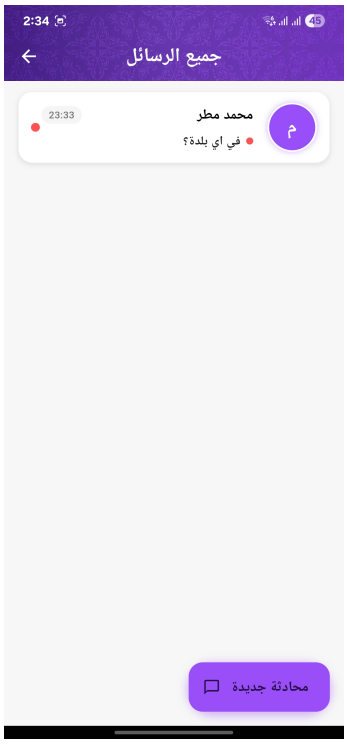

Figure 5.19: Chat list

Islamic Utilities

Additional Islamic features including prayer times and daily remembrance (Azkar) functionality.

Figure 5.20: daily Azkar feature

Figure 5.21: Prayer times

User Discovery and Search

User discovery capabilities within the platform.

Figure 5.22: User search

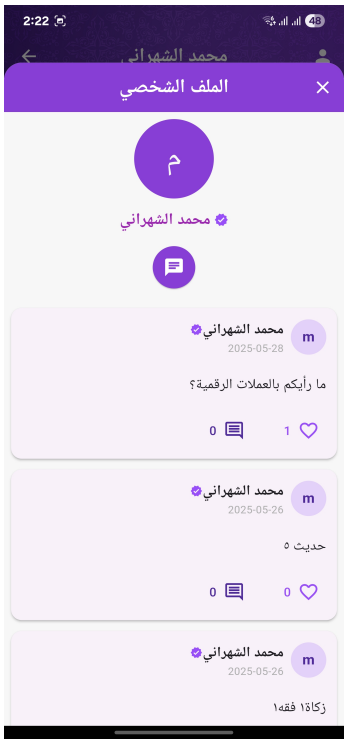

Figure 5.23: profile viewing

Account Settings and Profile Management

Comprehensive account management including personal details editing and security settings.

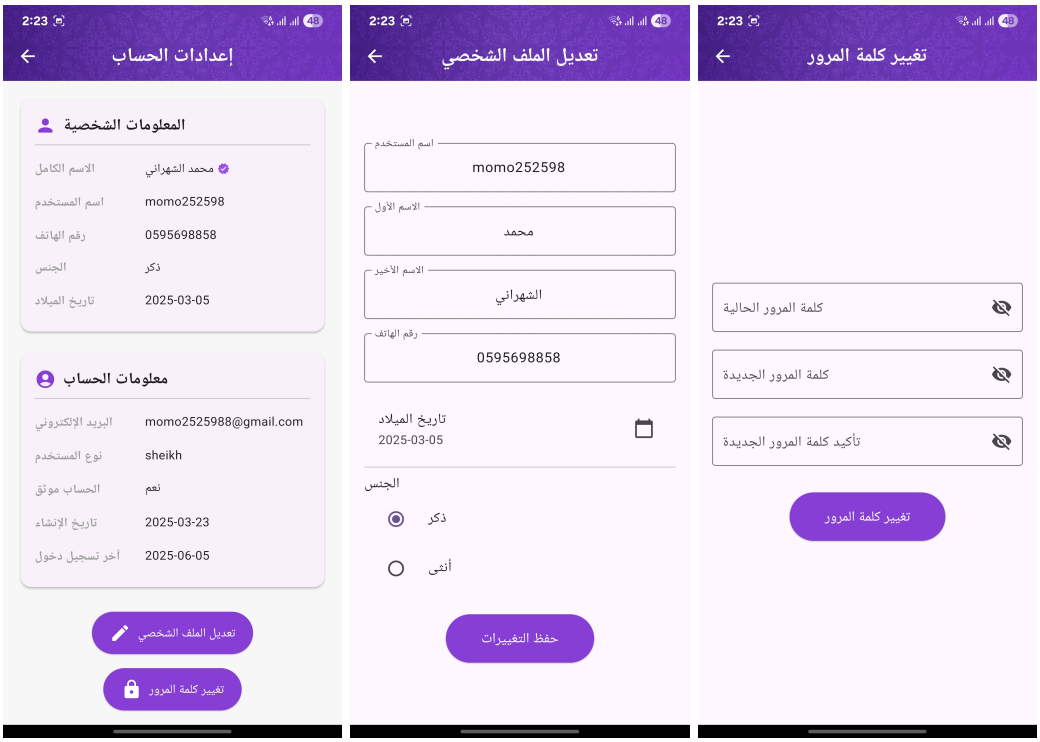

Figure 5.24: Account settings and profile management

Figure 5.25: Logout confirmation interface

Sheikh Verification System

The specialized verification workflow for religious scholars and administrators.

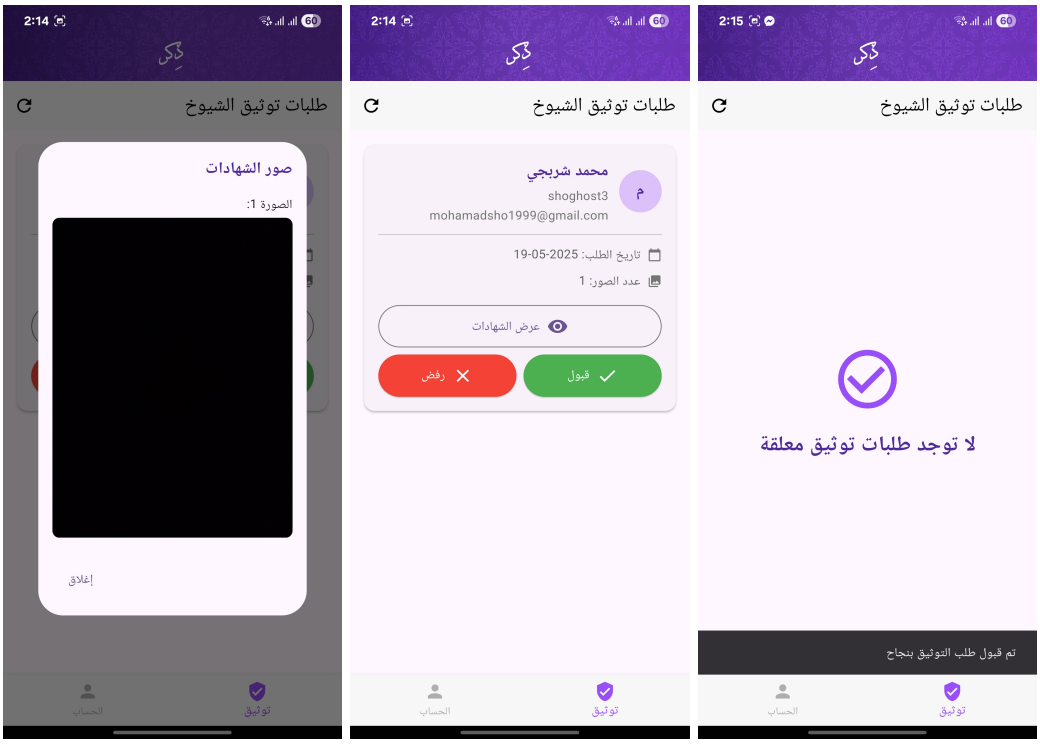

Figure 5.26: Sheikh verification process and administrative review

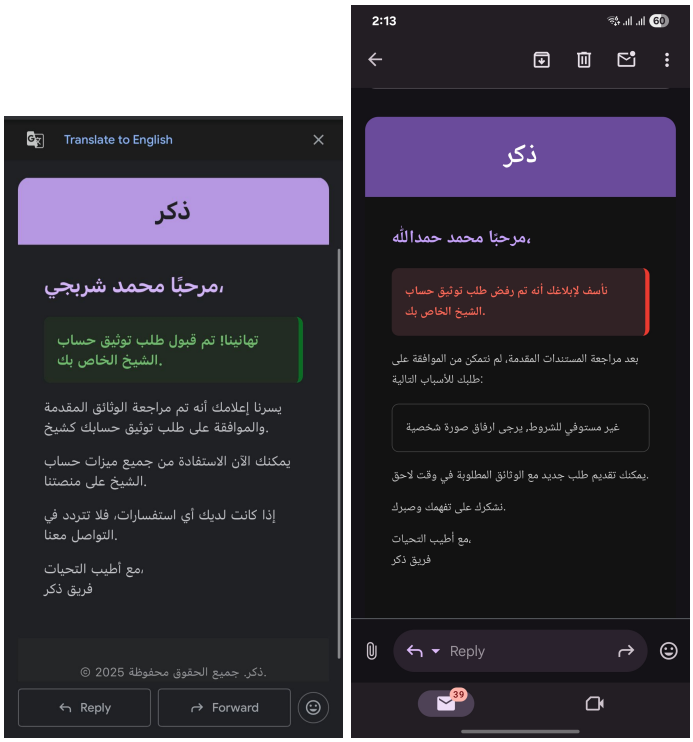

Figure 5.27: Sheikh verification email notifications

Notification System

The platform's comprehensive notification system for different types of user interactions and Islamic reminders.

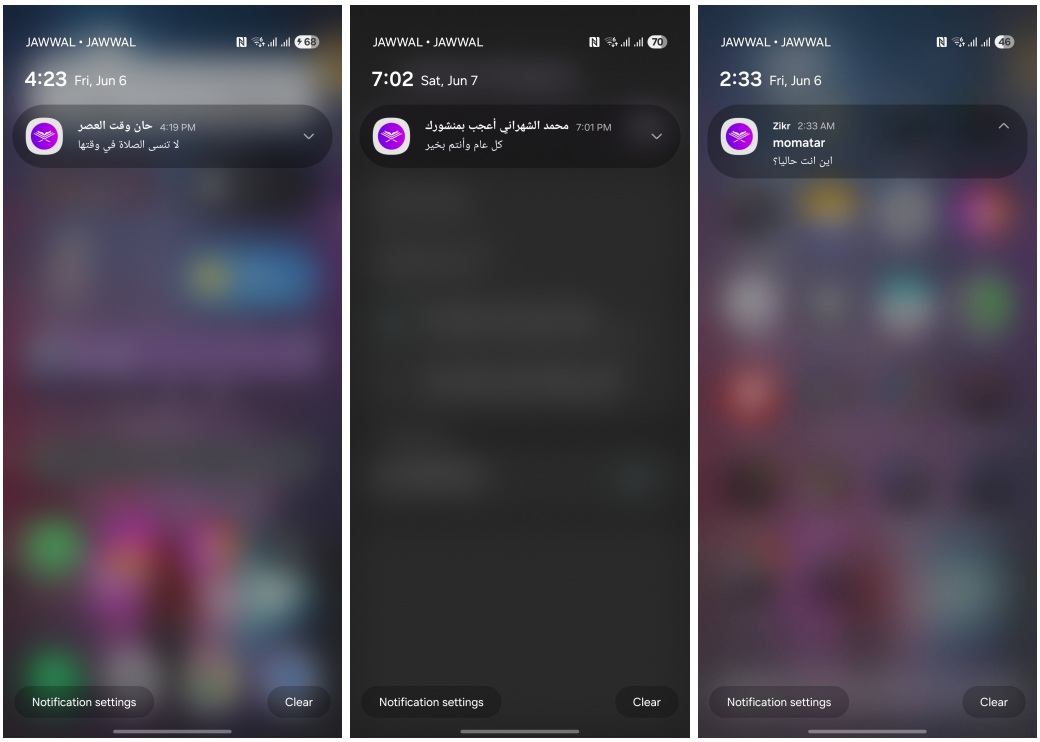

Figure 5.28: Prayer, social interaction, and messaging notifications

5.3.2 Web App UI

The Zikr web application provides a comprehensive desktop experience that mirrors the mobile functionality while taking advantage of larger screen real estate and desktop interaction patterns. The web interface maintains consistency with the mobile design language while optimizing layouts for desktop usage.

Authentication and Account Management

The web authentication system provides streamlined access with responsive design optimized for desktop browsers.

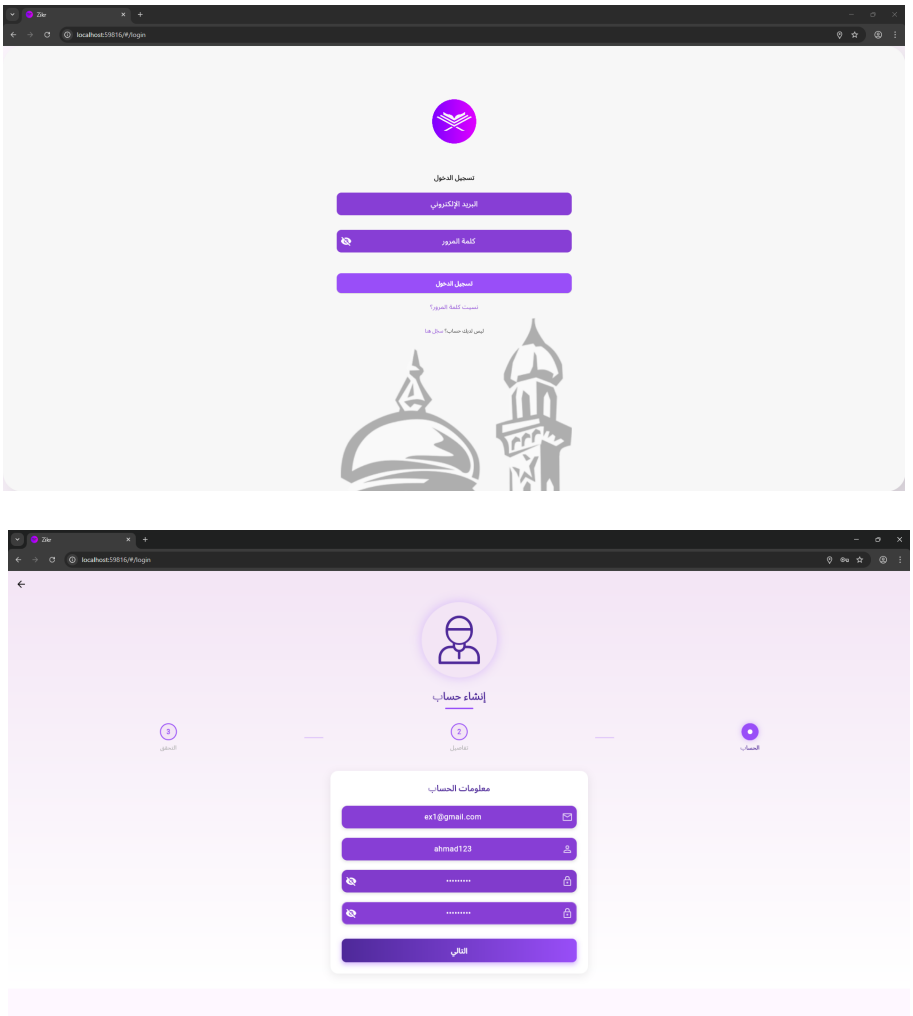

Figure 5.29: Web login interface and account creation

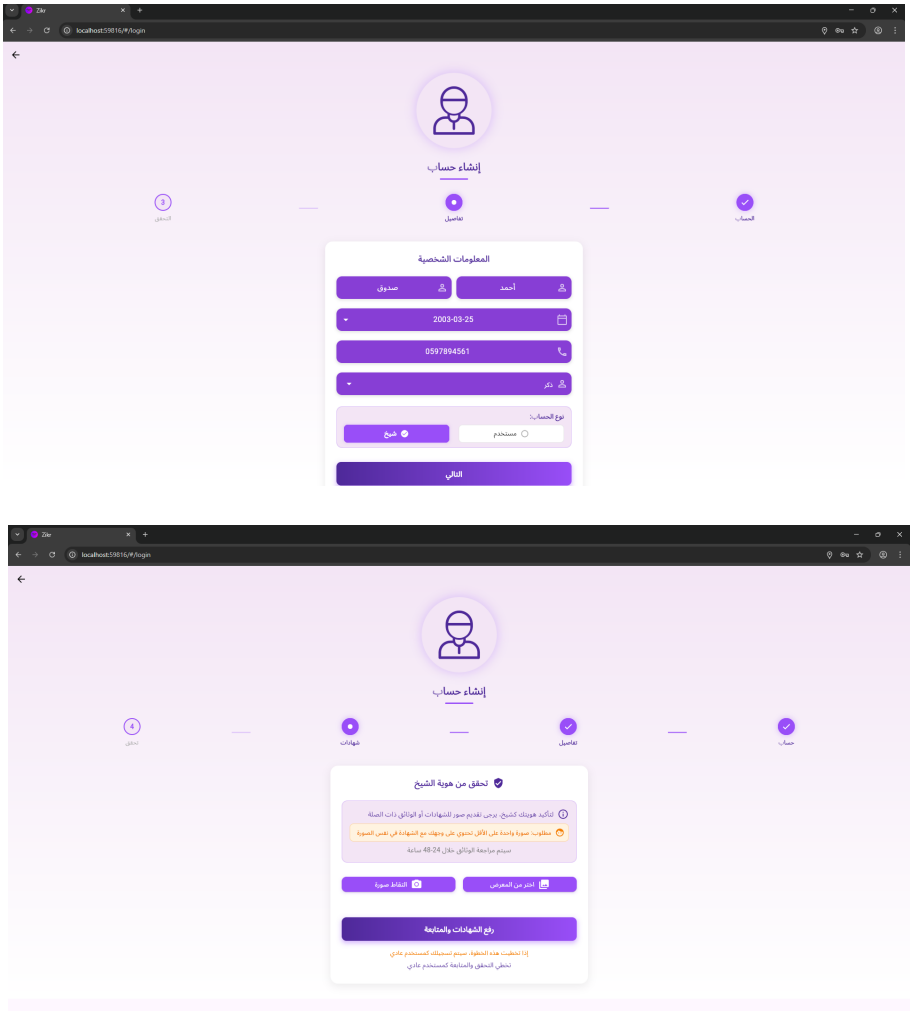

Figure 5.30: Personal details and sheikh certification upload

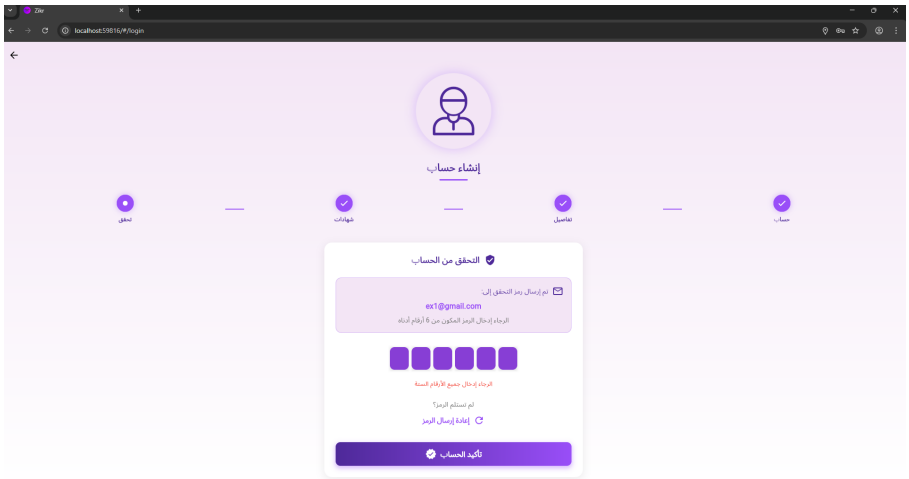

Figure 5.31: Email verification process

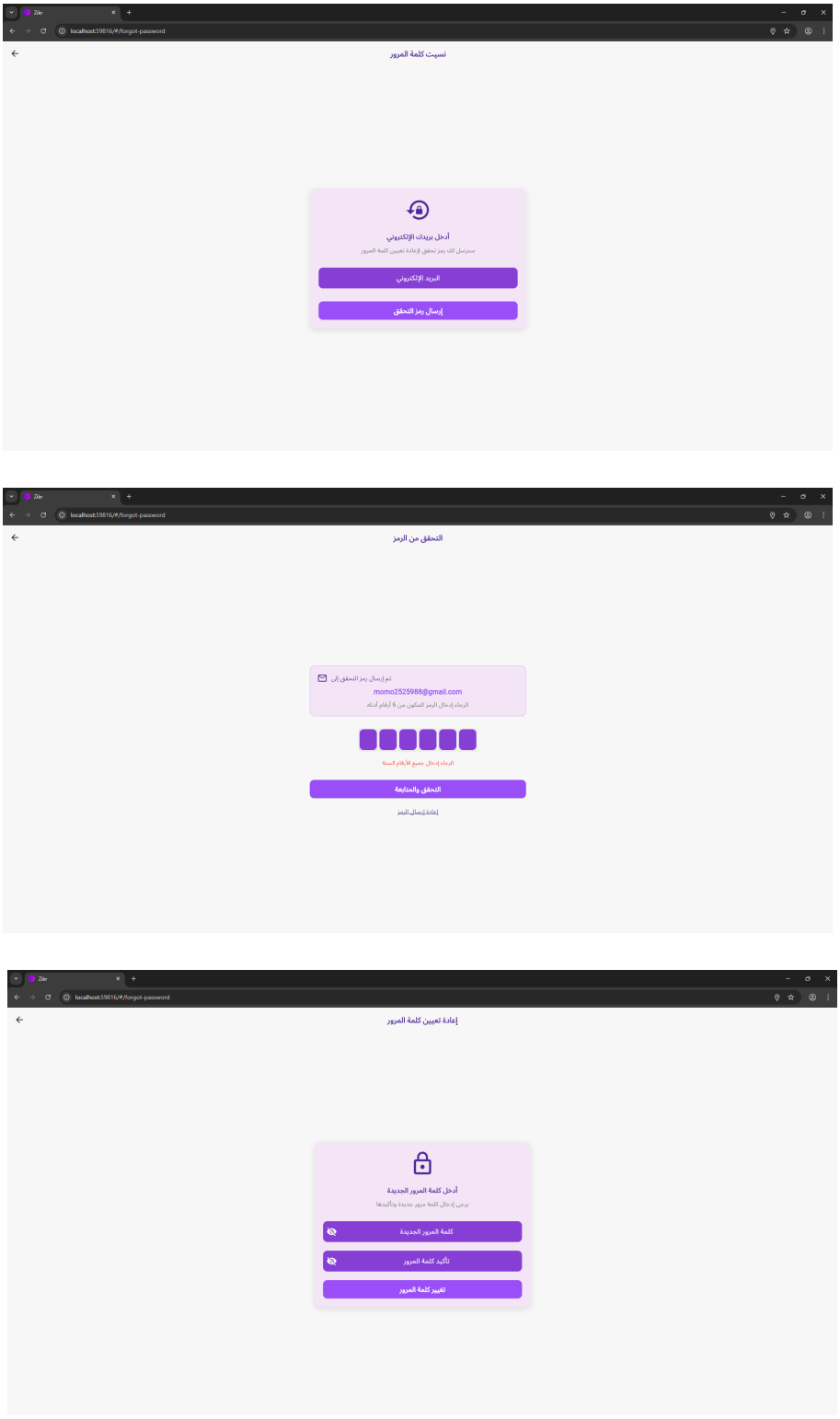

Figure 5.32: Password recovery workflow

Main Dashboard and Navigation

The web dashboard provides comprehensive access to all platform features with optimized desktop navigation.

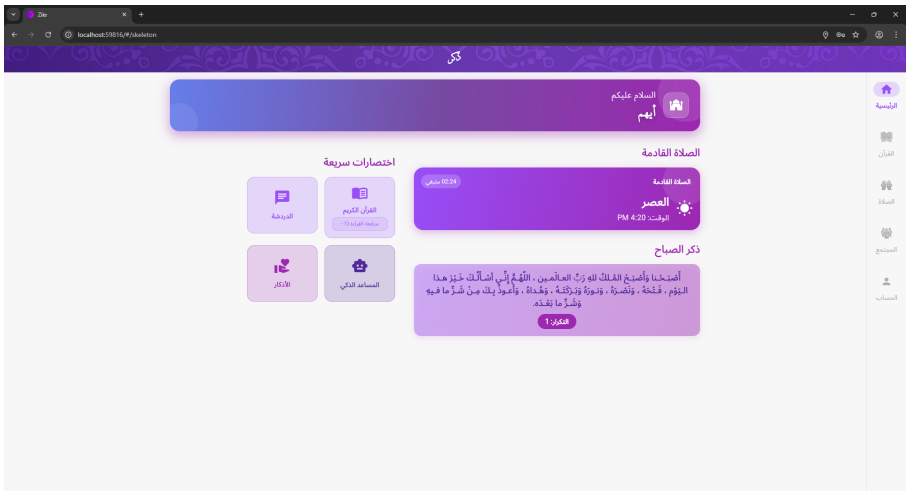

Figure 5.33: Web application home dashboard

Social Feed and Content Management

The web social feed interface takes advantage of desktop screen space to provide enhanced content browsing and interaction capabilities.

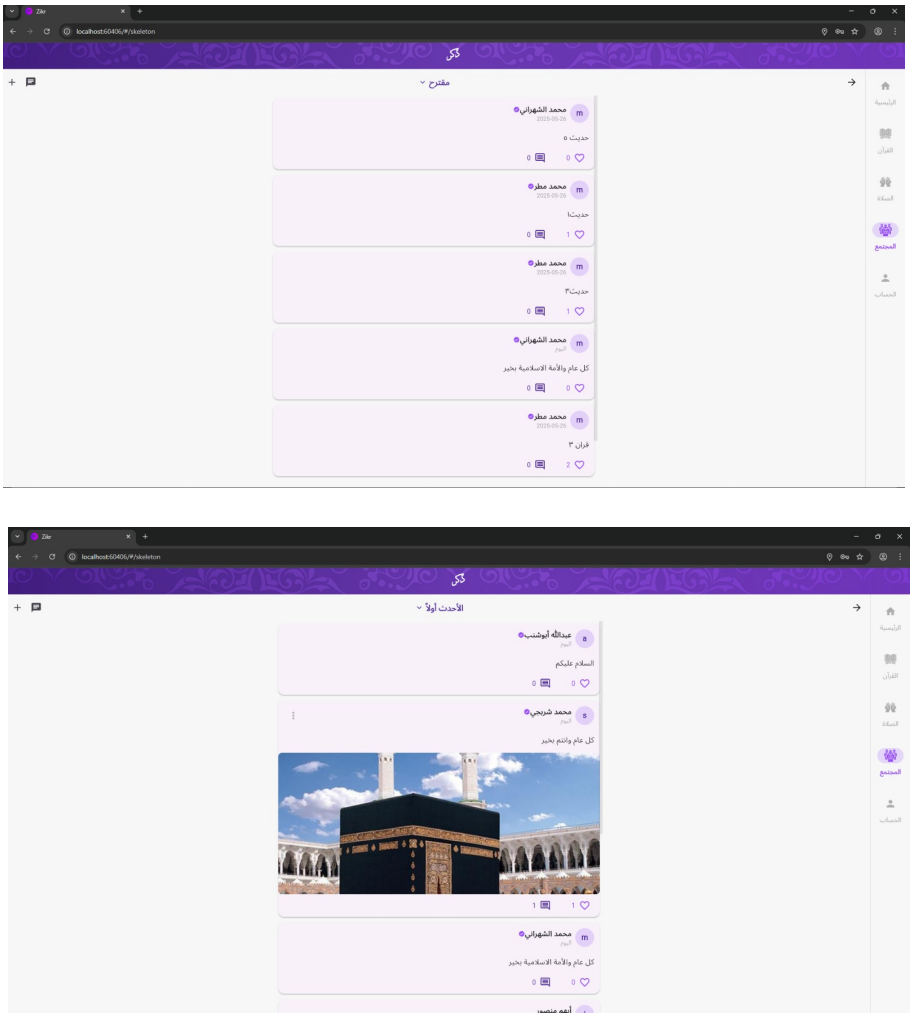

Figure 5.34: Suggested and recent posts feed views

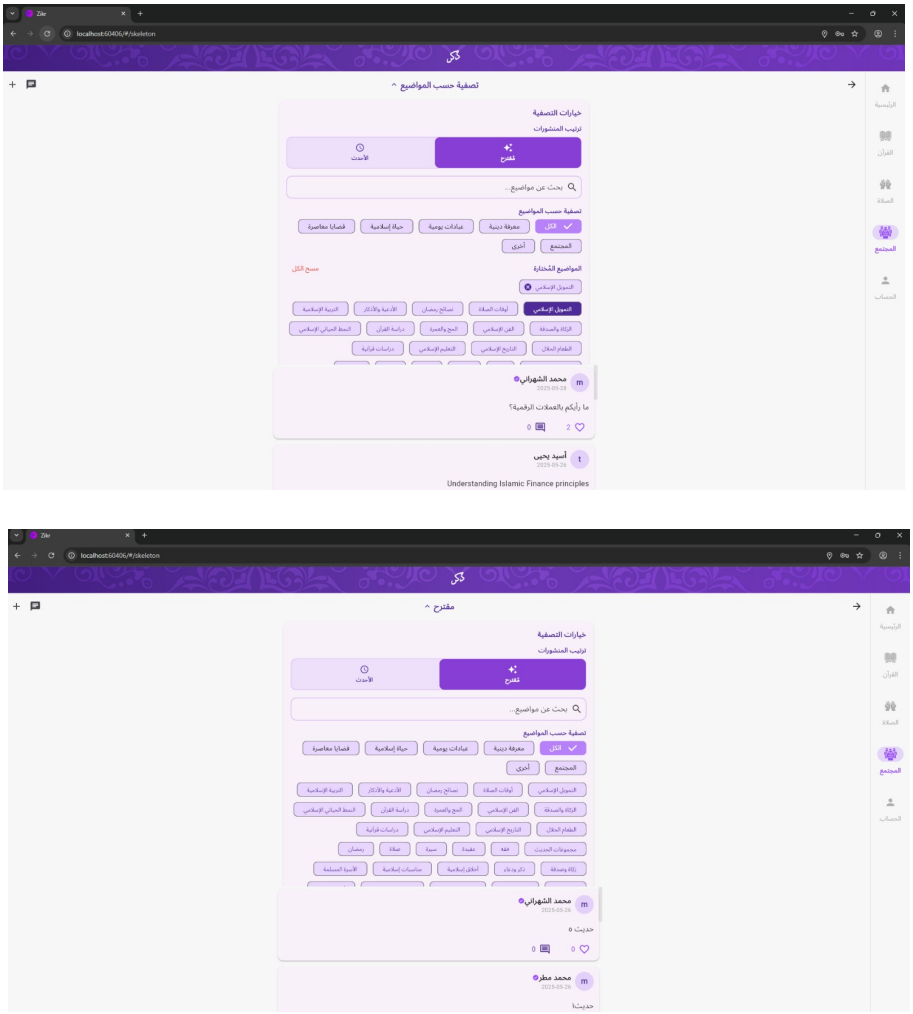

Figure 5.35: Topic-based content filtering system

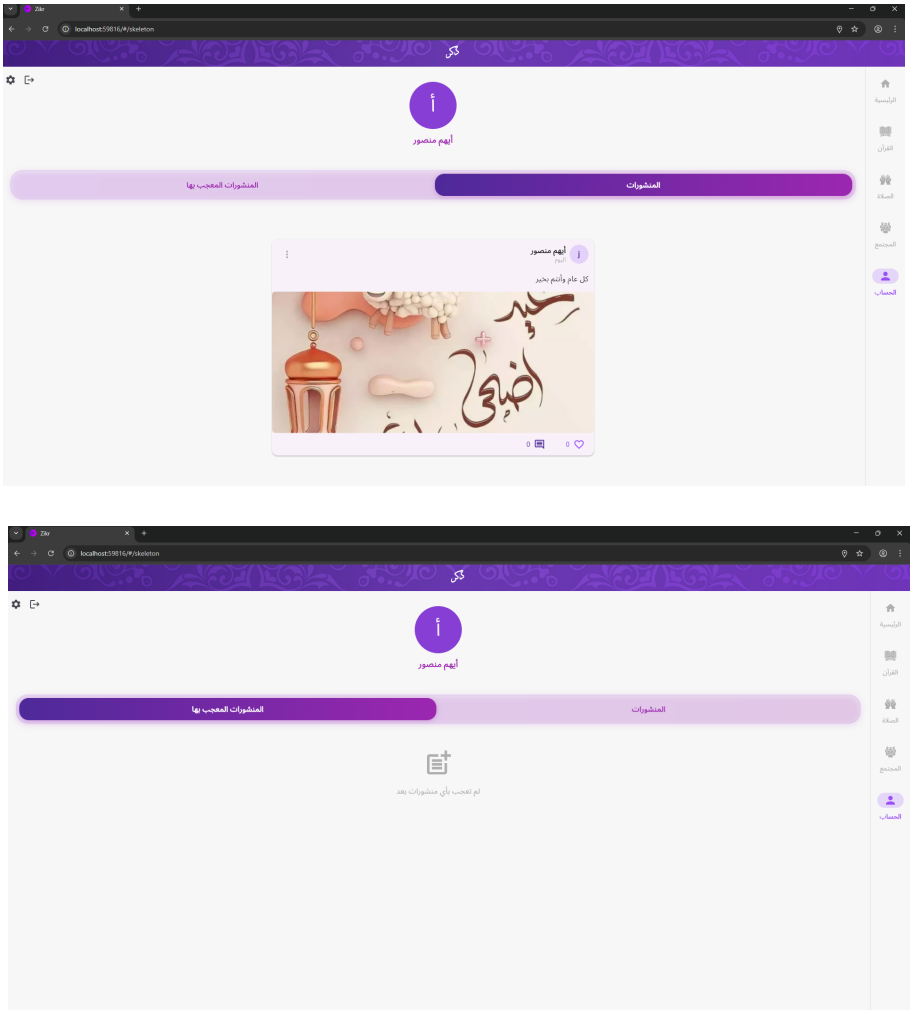

Figure 5.36: Personal posts management and liked posts collection

Digital Quran Interface

The web Quran interface provides an enhanced reading experience with advanced topic visualization and navigation features.

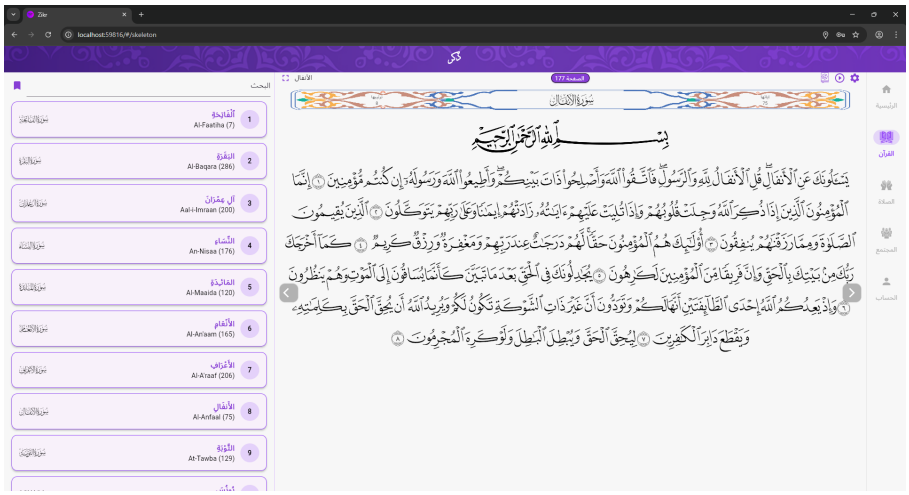

Figure 5.37: Quran reading interface with verse navigation

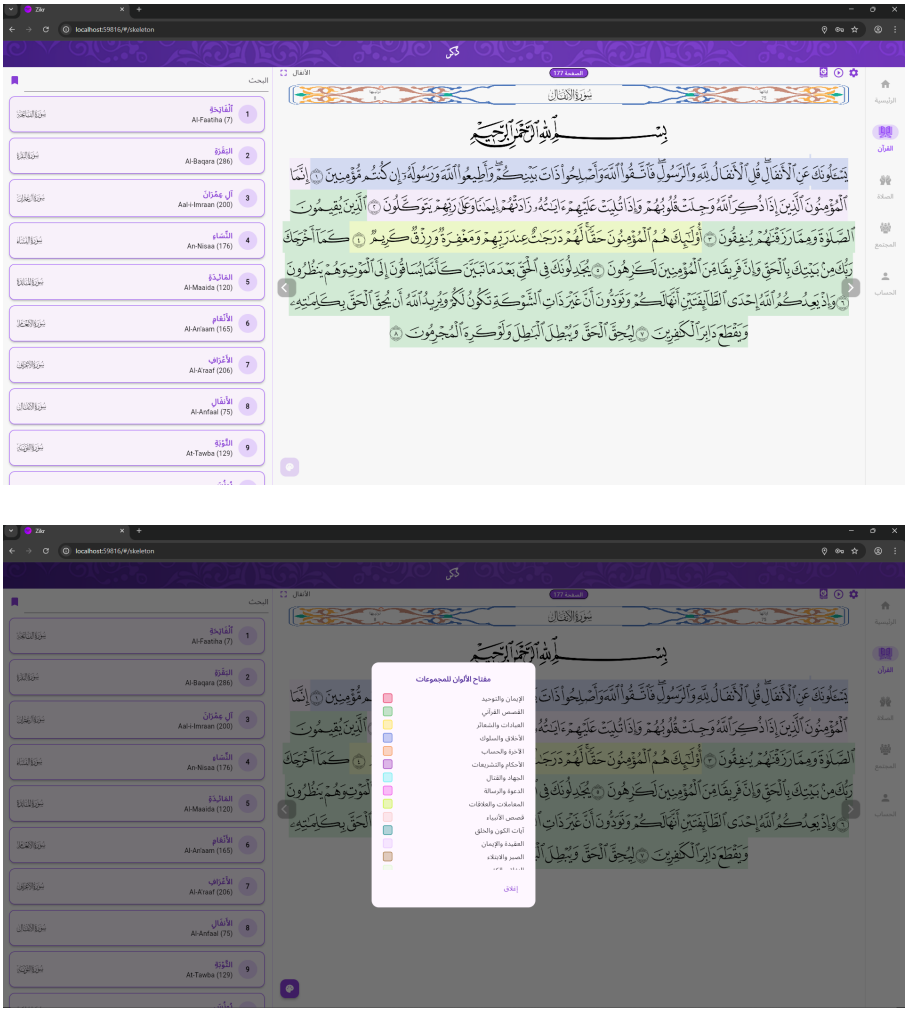

Figure 5.38: Quran page with topic highlighting, clustering guide, and full-screen mode

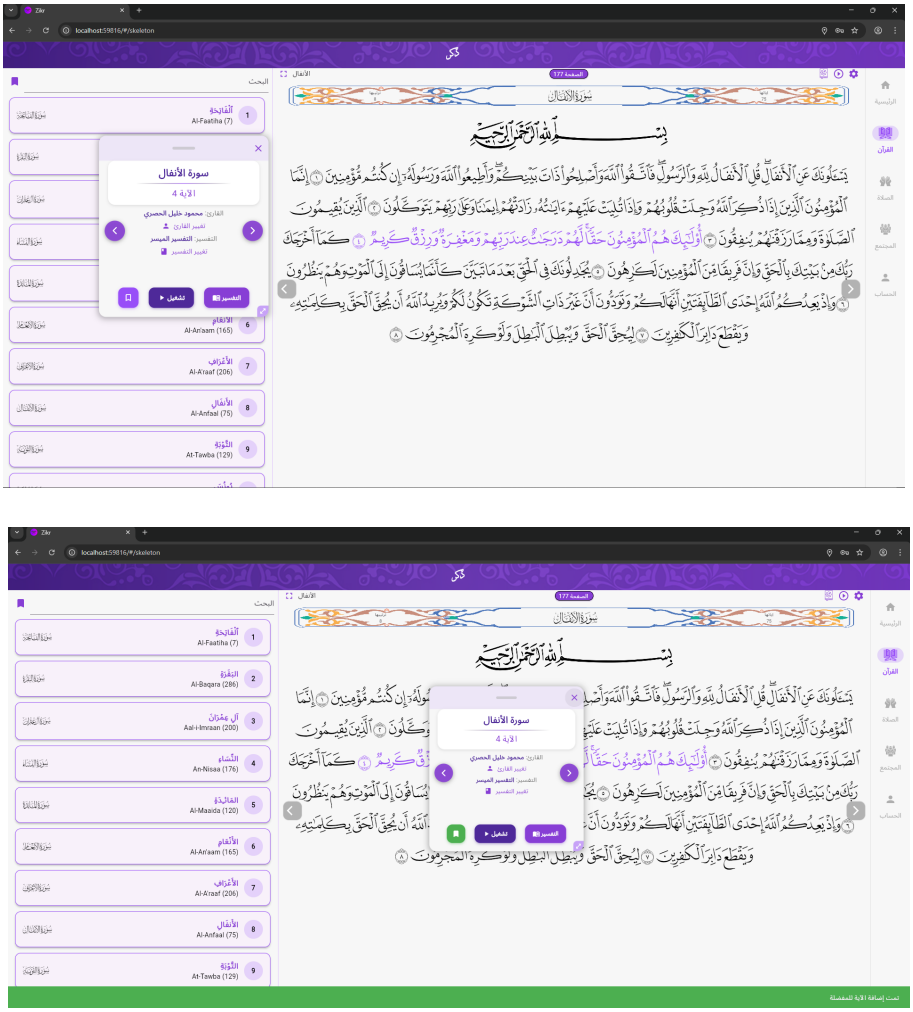

Figure 5.39: Verse details floating interface and bookmarking

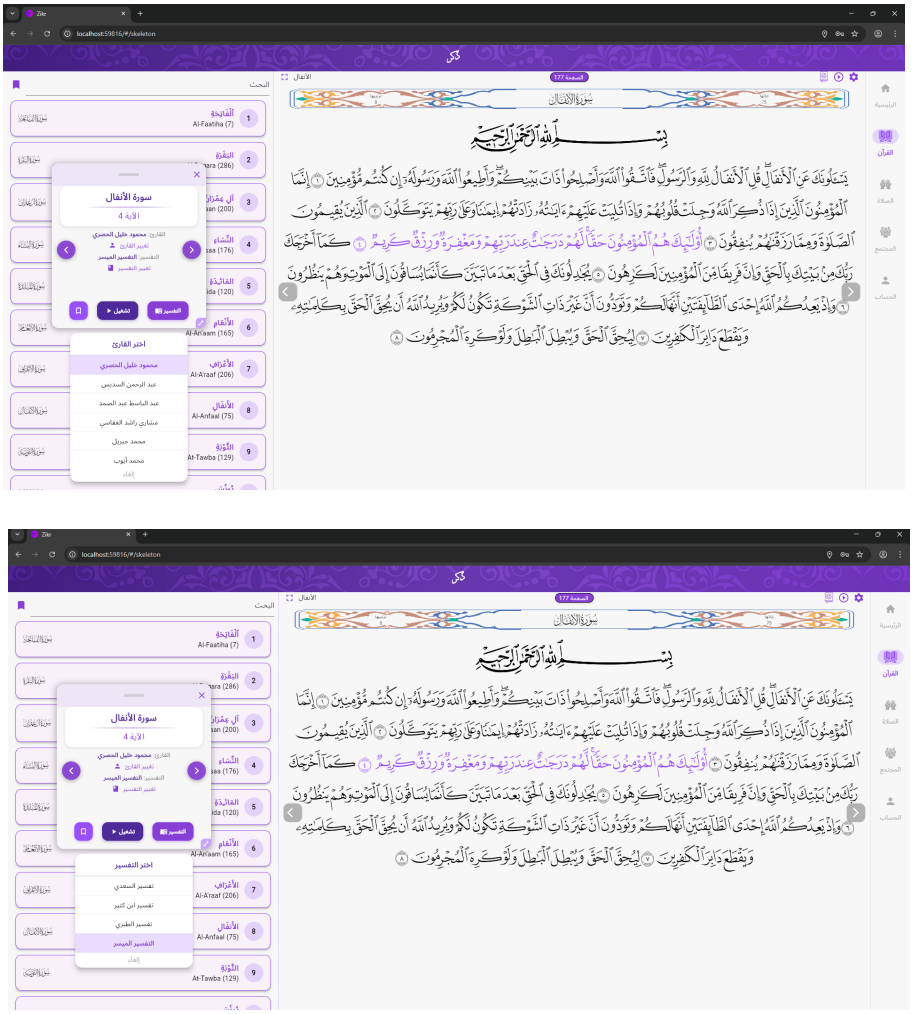

Figure 5.40: Audio recitation and Tafseer selection interfaces

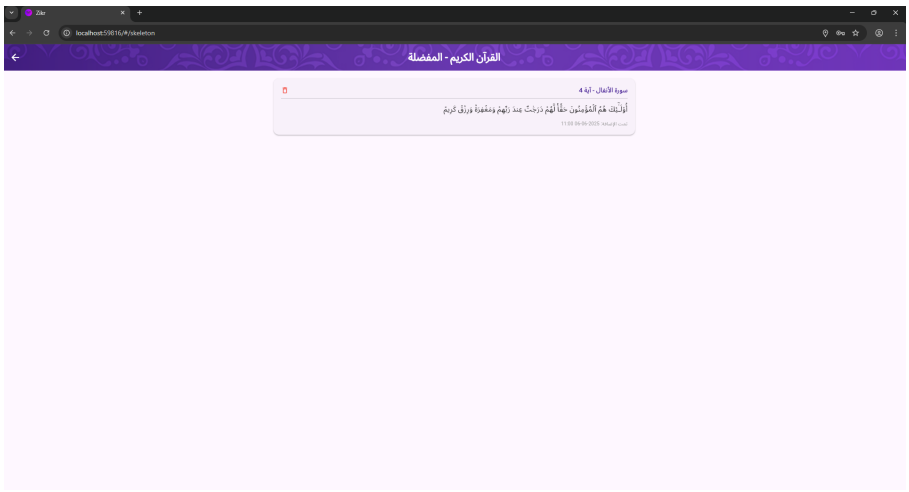

Figure 5.41: Bookmarked verses archive

AI-Powered RAG Chatbot

The web chatbot interface provides an enhanced conversational experience with improved text display and interaction capabilities.

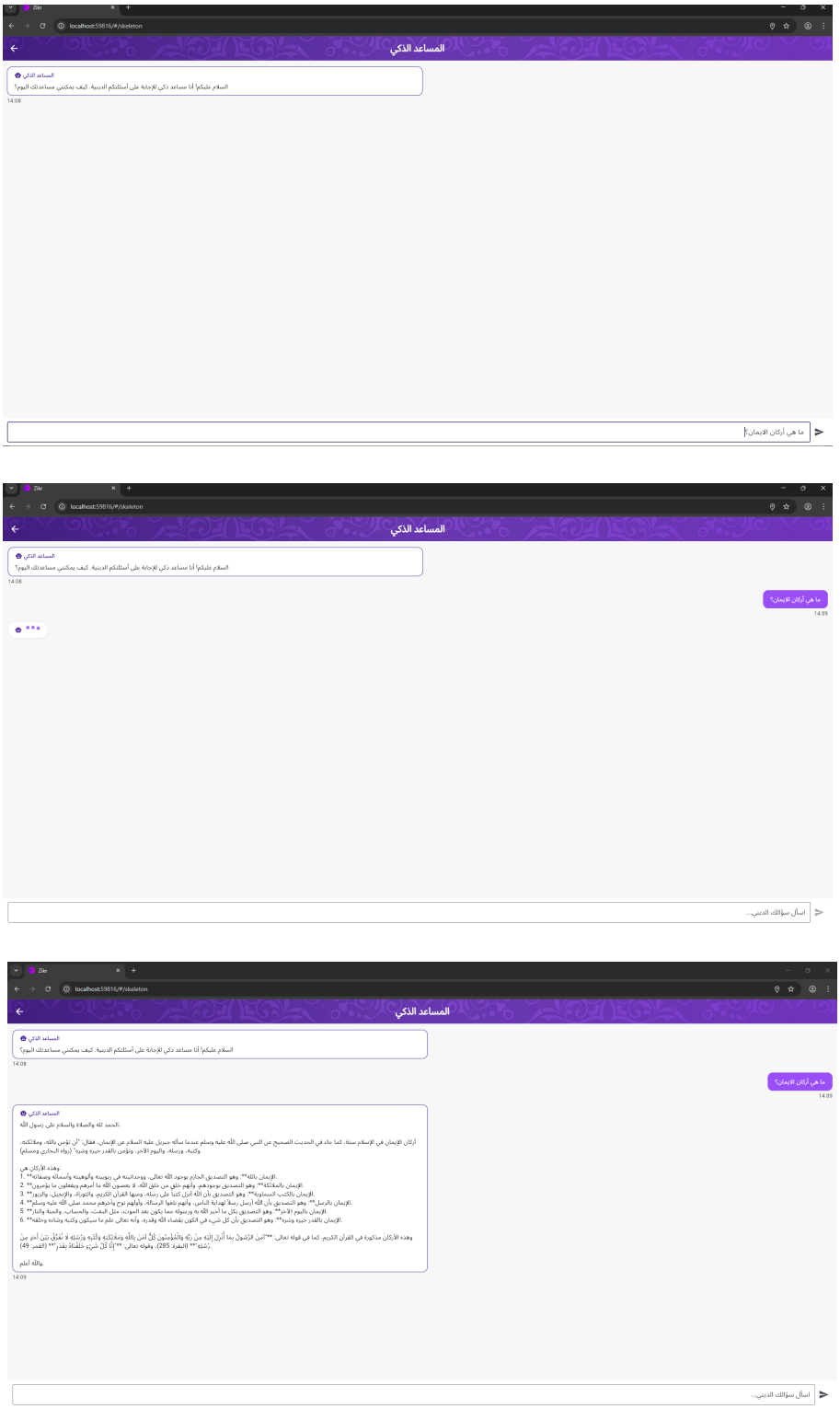

Figure 5.42: RAG chatbot conversation interface and interactions

Communication and Messaging

The web messaging system provides enhanced chat capabilities with improved user search and conversation management.

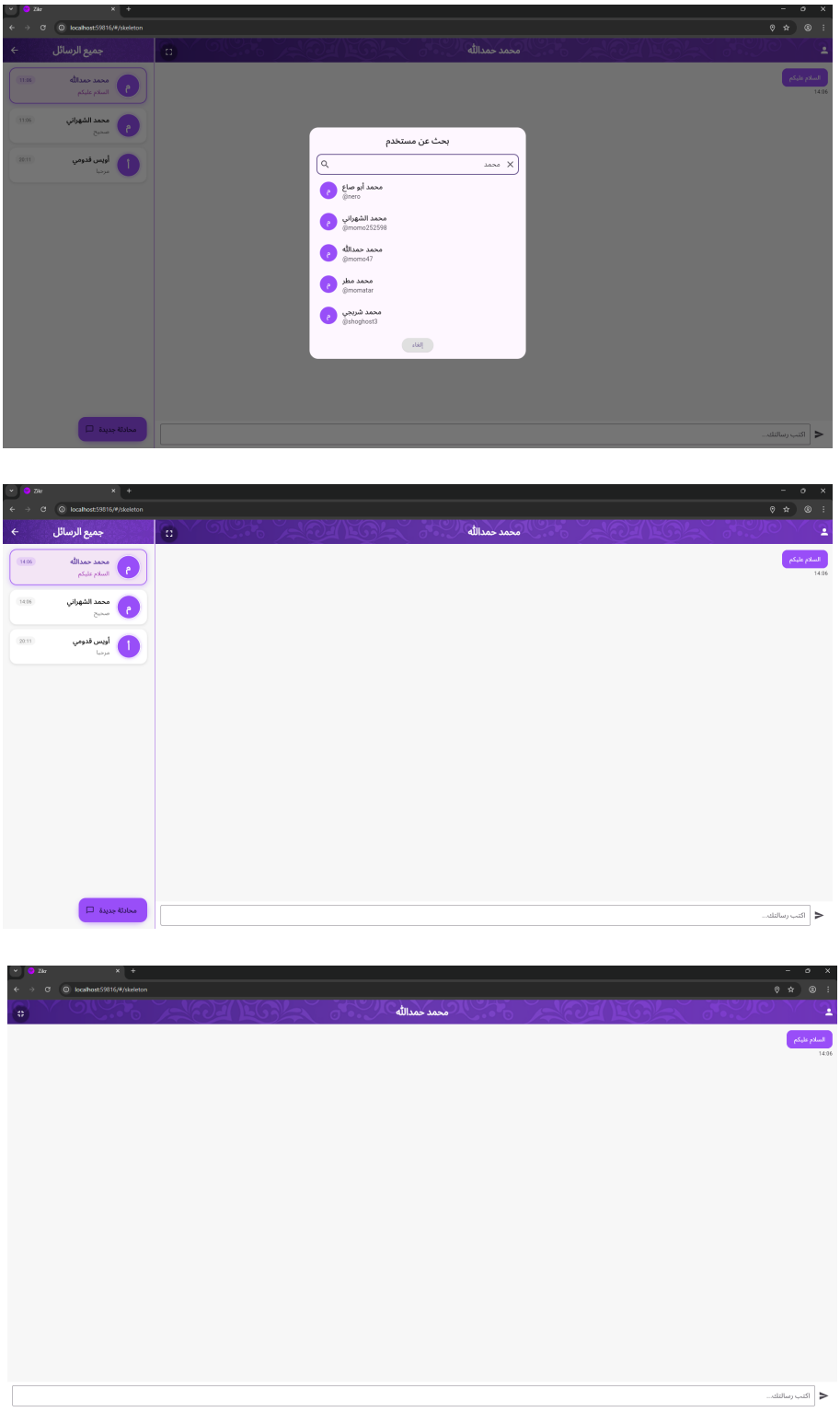

Figure 5.43: User search and chat interface variations

Islamic Utilities

The web platform includes comprehensive Islamic utilities with enhanced desktop layouts.

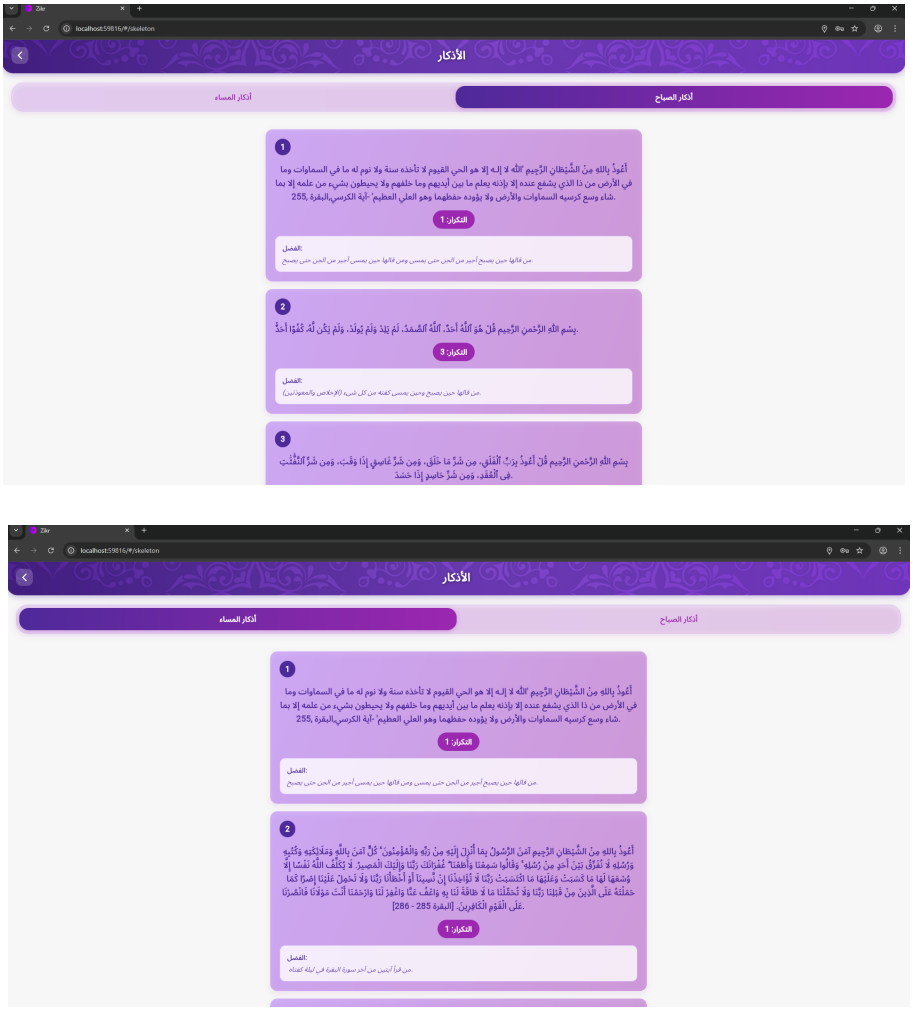

Figure 5.44: Morning and evening Azkar interfaces

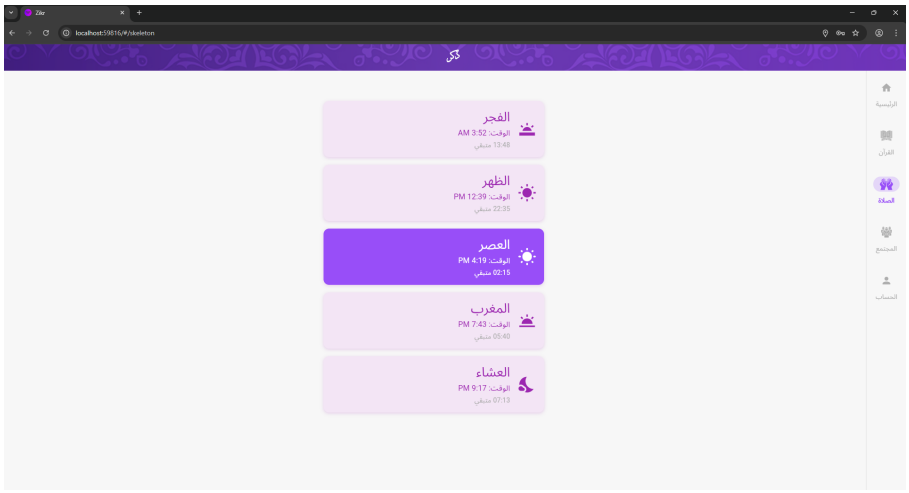

Figure 5.45: Prayer times display with location-based information

Account Settings and Profile Management

The web account management interface provides comprehensive settings access with desktop-optimized layouts.

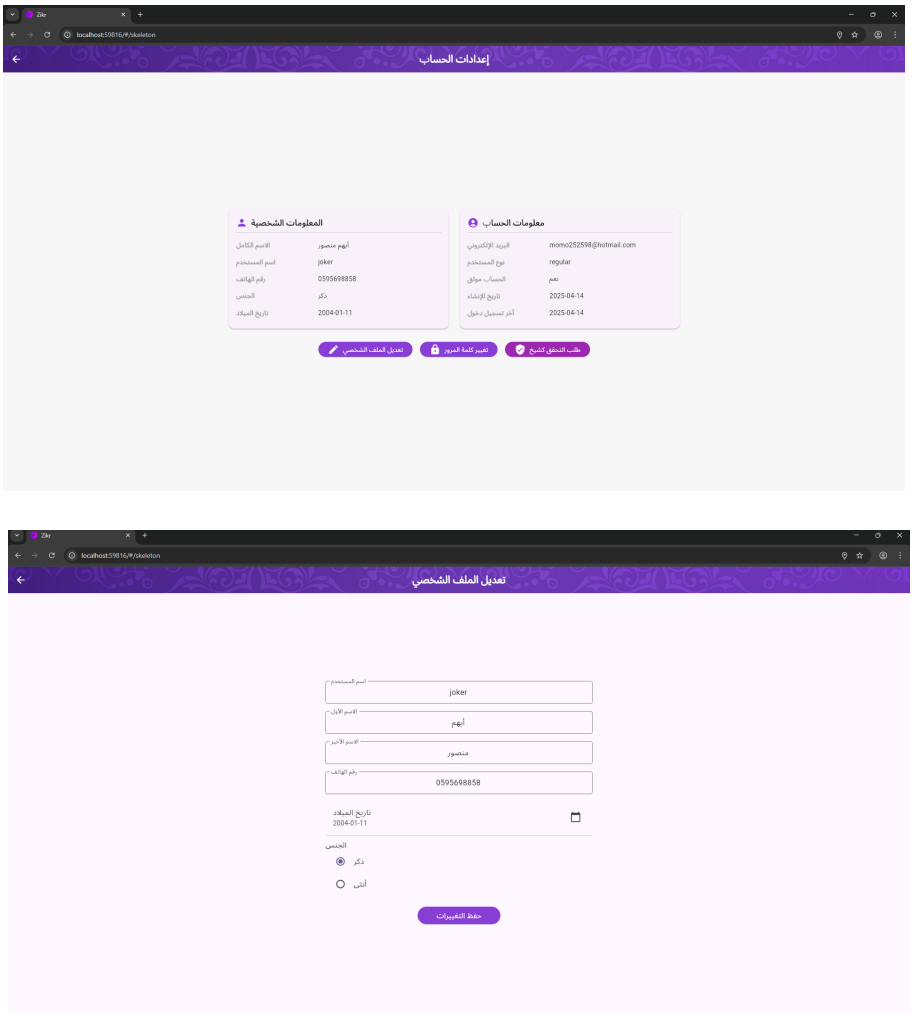

Figure 5.46: Account settings and profile editing interface

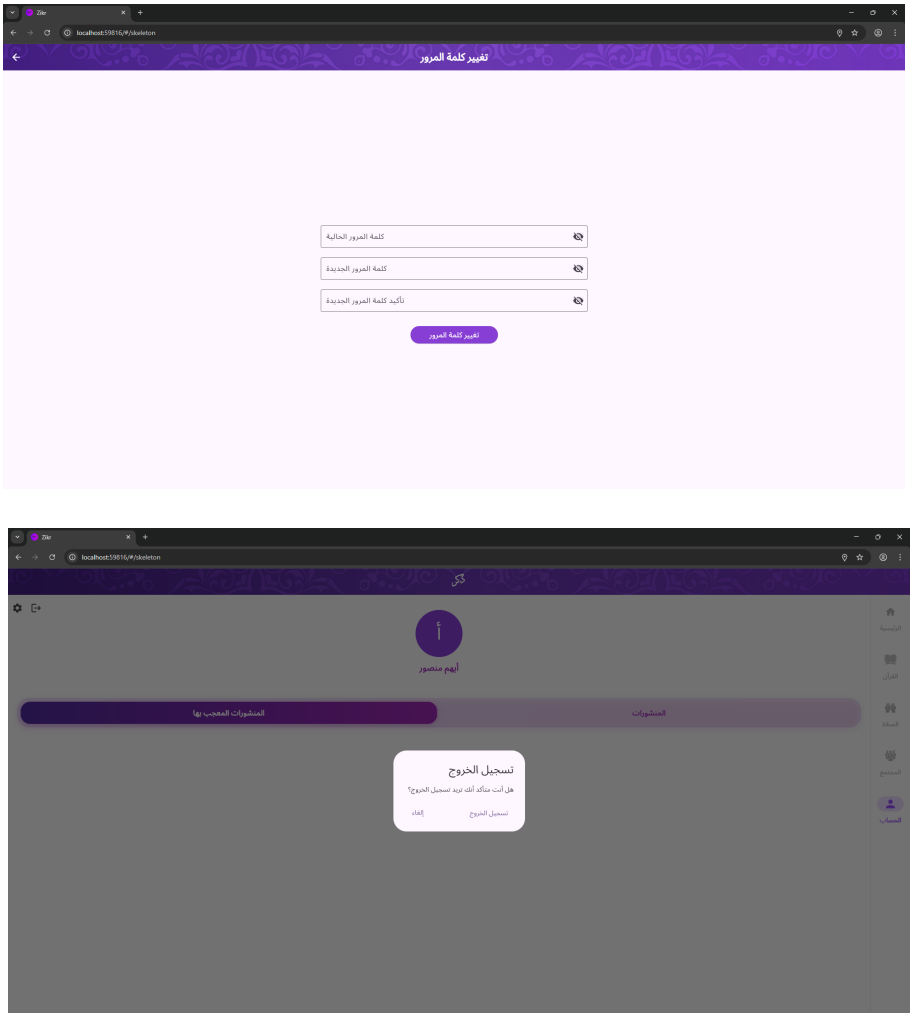

Figure 5.47: Password change and logout confirmation

Sheikh Verification System

The web administrative interface provides enhanced tools for sheikh verification and management.

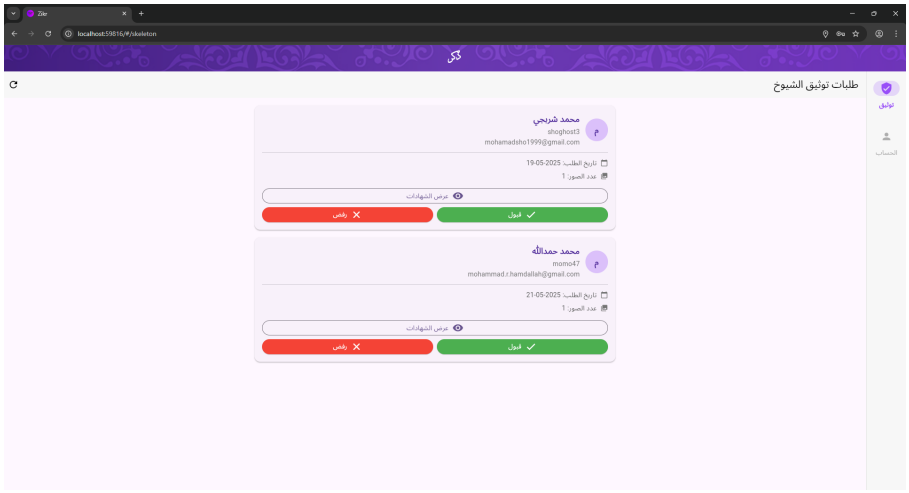

Figure 5.48: Administrative dashboard for pending sheikh verification requests

Figure 5.49: Sheikh certification review interface

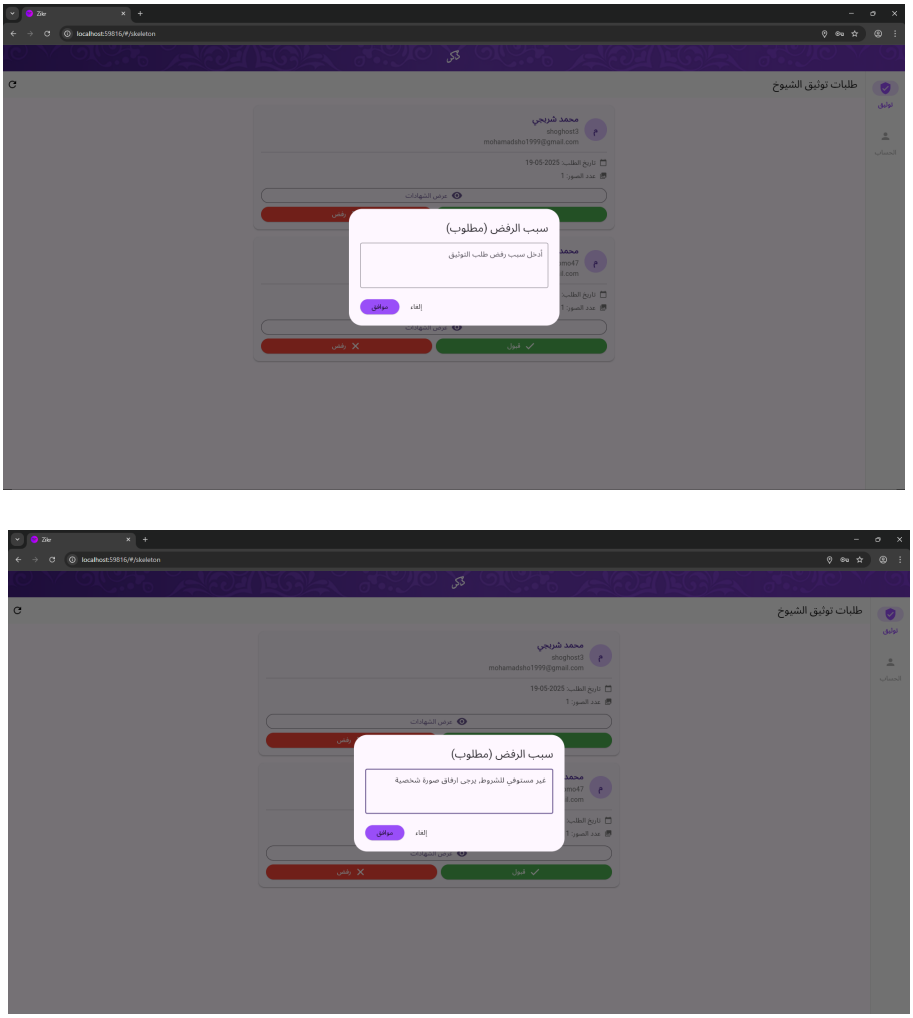

Figure 5.50: Rejection reason interface for sheikh verification

5.4 System Integration and Deployment

The complete system demonstrates successful integration of all components with robust error handling, user authentication across platforms, and reliable AI model serving. The platform is prepared for deployment with optimized build configurations for both mobile app stores and web hosting environments.

The results demonstrate that Zikr successfully achieves its primary objectives of creating an AI-enhanced Islamic social platform that integrates advanced machine learning techniques with social community features, providing a comprehensive digital environment for Islamic learning and community engagement.

Chapter 6

Discussion

This chapter provides a comprehensive evaluation of the Zikr platform development, analyzing the achieved results against the original objectives, discussing the contributions to the field, and identifying limitations encountered during development. The discussion examines both the technical achievements and limitations of the implemented system while positioning the work within the broader context of AI applications in Islamic education.

6.1 Interpretation and Analysis of Results

6.1.1 Problem Resolution Assessment

The Zikr platform successfully addresses the core problems identified in the initial research proposal. The primary challenge of creating an AI-enhanced Islamic educational platform that integrates advanced machine learning techniques with social community features has been comprehensively resolved through the implementation of three distinct AI components working in harmony within a unified social environment.

The scarcity of high-quality Islamic datasets, identified as a critical limitation in existing research, was effectively addressed through systematic data collection and preprocessing pipelines. The project successfully compiled and processed 88,000 Fatawa entries from IslamQA for the RAG chatbot and comprehensively preprocessed Tafseer Al-Saadi for the clustering model. This represents a significant contribution to the availability of structured Islamic datasets for computational analysis.

The complexity of Arabic natural language processing in religious contexts was tackled through specialized preprocessing techniques, including the use of CAMeL Tools for diacritics removal, Stanza pipeline for lemmatization, and carefully curated stopword filtering that preserved meaningful religious terminology. The TF-IDF vectorization with 2,151 optimized features demonstrated effective dimensionality reduction while maintaining semantic richness.

The need for safe, distraction-free social spaces that maintain Islamic principles while providing modern functionality was successfully addressed through the integrated platform design. The combination of content filtering, topic-based organization, and AI-powered personalization creates an environment that promotes meaningful religious discussion while minimizing exposure to inappropriate content.

6.1.2 Technical Achievement Analysis

The K-Means clustering model for Quranic topic identification achieved its primary objective of creating meaningful thematic categorizations. The identification of 20 distinct clusters

with clear semantic labels demonstrates successful unsupervised learning on classical Arabic religious text. The optimal cluster determination through elbow method and silhouette score analysis (achieving highest scores at 20 clusters) validates the methodological approach. However, the reliance on pre-computed static clustering limits real-time adaptability and personalized topic interpretation.

The RAG-based Fatawa chatbot successfully integrates retrieval and generation capabilities for Islamic question-answering. The system demonstrates effective semantic search across the 88,000-entry dataset and generates contextually appropriate responses through the DeepSeek-V3 model with optimized temperature settings (0.4). The implementation of source citation mechanisms enhances transparency and religious authenticity. Nevertheless, the system's performance remains dependent on the quality and coverage of the underlying IslamQA dataset, potentially limiting responses to questions outside this scope.

The fuzzy logic content recommendation system successfully balances personalization (70

6.1.3 Platform Integration Success

The cross-platform Flutter architecture successfully delivers consistent user experience across mobile and web platforms, with approximately 95

The sheikh verification system addresses a unique requirement in Islamic platforms, providing administrative workflows for validating religious authority while maintaining security and transparency. The comprehensive notification system supports both social interactions and Islamic reminders, creating an engaging user experience that respects religious practices.

6.2 Contributions to the Field

6.2.1 Technical Contributions

This project makes several significant technical contributions to the intersection of AI and Islamic education. The development of a specialized RAG system for Islamic Fatawa represents the first known implementation of retrieval-augmented generation specifically designed for religious question-answering in both Arabic and English. The system demonstrates novel applications of vector databases for Islamic content retrieval while maintaining religious authenticity through source citation mechanisms.

The K-Means clustering approach for Quranic topic modeling with color-coded visualization introduces innovative techniques for religious text analysis. The processing of Tafseer Al-Saadi using optimized Arabic NLP techniques and the creation of interpretable thematic clusters advance the field of computational Islamic studies. The integration of these clusters into a user-facing interface with real-time highlighting represents a practical application of unsupervised learning in religious education.

The fuzzy logic-based recommendation system tailored for Islamic social content represents a methodological advancement in content suggestion algorithms. The balance between personalization and content diversity addresses known limitations in religious social platforms while fostering comprehensive Islamic learning.

6.2.2 Methodological Contributions

The dual-track development methodology combining traditional software engineering with specialized AI development provides a framework for similar interdisciplinary projects. The

systematic approach to Arabic NLP preprocessing, including diacritics handling, religious terminology preservation, and classical text normalization, establishes best practices for future Islamic text processing projects.

The comprehensive evaluation framework combining quantitative metrics (silhouette scores, elbow method) with qualitative validation (semantic coherence, religious authenticity) demonstrates effective assessment methodologies for AI applications in religious contexts.

6.2.3 Educational and Social Impact

The platform addresses the critical need for safe, educationally focused digital spaces for Muslim communities. The integration of AI-powered learning tools with community features creates unprecedented opportunities for enhanced religious engagement and peer learning. The multilingual support (Arabic and English) increases accessibility for diverse Muslim populations, particularly important for Muslims in non-Arabic speaking regions.

The implementation of Islamic principles within modern social media functionality demonstrates how traditional religious values can be preserved while embracing technological advancement. The platform provides a model for religiously-sensitive AI implementation that respects cultural and theological constraints.

6.3 Limitations and Challenges

6.3.1 Technical Limitations

Several technical limitations must be acknowledged. The K-Means clustering model uses static pre-computed results, limiting adaptability to evolving interpretations or personalized topic understanding. The reliance on unsupervised learning means that topic assignments may not always align with traditional Islamic scholarship categorizations, requiring ongoing validation by religious authorities.

The RAG chatbot's performance is constrained by the quality and scope of the IslamQA dataset. While comprehensive, this dataset may not cover all contemporary Islamic questions or represent diverse scholarly perspectives. The system's inability to access real-time fatawa or contemporary religious rulings limits its applicability to emerging religious questions.

The recommendation system's dependence on explicit user interactions (likes, comments) may not capture the full spectrum of user interests or learning needs. The current implementation lacks sophisticated user behavior analysis that could improve recommendation accuracy.

6.3.2 Platform Limitations

Cross-platform development, while achieving significant code reuse, introduced complexity in handling platform-specific features, particularly for push notifications and authentication flows. The Firebase integration required custom solutions for web-mobile compatibility, adding development overhead.

The Arabic text rendering challenges, especially for Quranic text with diacriticals, required specialized font implementations and may not display consistently across all devices or browsers. The platform's performance optimization strategies, while effective for the current user base, may require scaling adjustments for larger communities.

6.3.3 Dataset and Language Processing Limitations

The scarcity of high-quality Arabic Islamic datasets remains a fundamental challenge. While the project successfully compiled substantial datasets, the coverage may not represent the full diversity of Islamic thought or contemporary religious discourse. The preprocessing decisions, such as diacritics removal and stopword filtering, may inadvertently remove nuanced meanings important in religious contexts.

The reliance on modern Arabic NLP tools for classical religious texts introduces potential accuracy issues, as these tools are optimized for contemporary Arabic rather than classical Quranic or scholarly Arabic. This limitation affects both the clustering model's accuracy and the RAG system's retrieval performance.

6.4 Positioning Within Current Knowledge

This work bridges a significant gap between academic AI research and practical religious applications. While studies by Ramzy et al. (2023) achieved 92% F1-scores in Hadith classification and Shohoud et al. (2023) demonstrated 0.97 cosine similarity in Quranic semantic search, these advances remained in academic isolation. Zikr represents the first comprehensive platform to translate such research achievements into a user-facing social learning environment.

The integration of multiple AI functionalities within a single platform addresses the fragmentation identified in current Islamic applications. Unlike existing platforms such as Tarteel AI or The Great Quran App, which focus on single functionalities, Zikr demonstrates the potential for holistic AI-enhanced religious education platforms.

The social community integration addresses a complete gap in the current landscape. No existing platform combines proven AI techniques with Islamic social learning methodologies, making Zikr a pioneering effort in community-driven religious education technology.

6.5 Logical Implications of Results

The successful implementation of Zikr demonstrates several important implications for the field of AI in religious education. The viability of processing classical Arabic religious texts using modern NLP techniques suggests potential for similar applications across other religious traditions and historical texts. The effective integration of multiple AI models within a social platform indicates the feasibility of comprehensive AI-enhanced educational environments.

The user acceptance of AI-generated religious content, when properly sourced and authenticated, suggests readiness within Muslim communities for technology-assisted religious learning. This acceptance opens opportunities for more sophisticated AI applications in religious contexts, including advanced question-answering systems, personalized religious education, and automated content moderation for religious platforms.

The successful balance between technological innovation and religious authenticity demonstrates a model for respectful AI implementation in sensitive cultural contexts. This approach could inform similar projects in other religious or cultural domains where traditional values must be preserved while embracing technological advancement.

Chapter 7

Conclusion and Recommendations

7.1 Summary of Key Results

This project successfully developed Zikr, an innovative AI-powered Islamic platform that integrates advanced machine learning techniques with social community features. The key technical achievements demonstrate the feasibility of applying sophisticated AI methods to Islamic education and community engagement.

The K-Means clustering model achieved optimal performance with 20 distinct thematic clusters for Quranic topic modeling, validated through elbow method and silhouette score analysis. The model successfully processed Tafseer Al-Saadi using specialized Arabic NLP preprocessing, including CAMeL Tools for diacritics removal and Stanza pipeline for lemmatization, resulting in TF-IDF vectorization with 2,151 optimized features that effectively captured semantic richness while maintaining computational efficiency.

The Retrieval-Augmented Generation (RAG) chatbot demonstrated effective Islamic Fatawa question-answering capabilities, utilizing a comprehensive dataset of 88,000 entries from IslamQA. The system achieved high accuracy in semantic search and contextually appropriate response generation through the DeepSeek-V3 model with optimized temperature settings (0.4), while maintaining religious authenticity through source citation mechanisms.

The fuzzy logic content recommendation system successfully balanced personalization (70

The platform addresses critical gaps identified in existing Islamic applications: the absence of comprehensive AI-enhanced social learning environments, the lack of integrated multi-functional AI systems, and the need for community-driven personalized content delivery. Zikr represents the first platform to successfully translate academic AI research achievements into a user-facing social learning environment for Islamic education.

7.2 Learning Experience and Reflections

The development of Zikr provided substantial learning experiences across multiple dimensions of interdisciplinary project management. The dual-track methodology combining traditional software engineering with specialized AI development required careful coordination and deep understanding of both domains. This approach highlighted the complexity of integrating cutting-edge AI research with practical user-facing applications while maintaining religious authenticity and cultural sensitivity.

The most significant challenge was navigating the scarcity of high-quality Arabic Islamic datasets and the complexity of processing classical Arabic religious texts using modern NLP tools optimized for contemporary Arabic. This experience emphasized the importance of

specialized preprocessing techniques and the need for domain-specific validation frameworks that combine quantitative metrics with qualitative religious authenticity assessments.

Cross-platform development presented unexpected complexities, particularly in Firebase Web/Mobile compatibility and Arabic text rendering across different devices and browsers. These challenges reinforced the value of platform-specific optimization while maintaining code reusability, and highlighted the importance of early identification of integration points in complex system architectures.

The project demonstrated that successful AI implementation in religious contexts requires not only technical proficiency but also deep understanding of cultural constraints and community needs. The sheikh verification system and content moderation approaches developed for Zikr established frameworks for religiously-sensitive AI implementation that respects theological principles while embracing technological advancement.

If undertaking a similar project, greater emphasis would be placed on early engagement with religious scholars and community leaders to ensure broader validation of AI-generated content and to establish more comprehensive ground truth datasets for model training and evaluation.

7.3 Recommendations for Future Enhancement

7.3.1 Advanced Islamic AI Systems

Hadith RAG System Development: Expand the current RAG architecture to encompass Hadith questions and lookups, creating an intelligent Musnad system. This enhancement would require curating comprehensive Hadith datasets from major collections (Sahih Bukhari, Sahih Muslim, Sunan Abu Dawood, etc.) and implementing specialized retrieval mechanisms that account for narrator chains (Sanad) and textual content (Matn). The system should provide contextual Hadith recommendations based on thematic similarity and scholarly authentication status.

Hadith Authentication AI Model: Develop a specialized machine learning model to assess the strength of Hadith chains (Isnad) automatically. This system would analyze narrator reliability, temporal connections, and geographical feasibility to classify Hadiths as Sahih (authentic), Hasan (good), Da'if (weak), or Mawdu (fabricated). The model should incorporate historical narrator databases and scholarly authentication methodologies to provide educational insights into the science of Hadith (Ulum al-Hadith).

Narrator Network Visualization: Implement advanced data visualization tools to map Hadith narrator networks, showing the chains of transmission from the Prophet Muhammad through successive generations. This feature would include interactive timelines, geographical mapping of narrator movements, and relationship clustering to help users understand the historical development of Islamic traditions and the reliability of transmission paths.

7.3.2 Enhanced Community Features

Live Q&A Sessions with Verified Sheikhs: Integrate real-time video streaming capabilities to enable scheduled live sessions where verified Islamic scholars can answer community questions. This feature should include session scheduling, question queue management, automatic session recording and archival, and post-session content indexing for future searchability. The system should also provide multilingual subtitle generation for broader accessibility.

Collaborative Learning Circles: Develop virtual study groups where users can form learning circles around specific Islamic topics, guided by AI-curated content and moderated by

verified scholars. These circles should include progress tracking, peer assessment capabilities, and gamification elements to encourage sustained engagement.

7.3.3 Scalability and Performance

Microservices Architecture: Transition from the current monolithic backend to a microservices architecture to support larger user communities and more sophisticated AI processing. This enhancement should include distributed caching systems, load balancing for AI model serving, and containerized deployment strategies for improved scalability and maintenance.

Edge Computing Integration: Implement edge computing solutions to reduce latency for AI-powered features, particularly for real-time Arabic text processing and topic highlighting. This approach would improve user experience in regions with limited internet connectivity while reducing server load.

Advanced Recommendation Systems: Enhance the current fuzzy logic approach with deep learning-based recommendation systems that can capture more complex user behavior patterns and preferences. The system should incorporate implicit feedback analysis, social network effects, and temporal usage patterns to provide more accurate and diverse content suggestions.

7.4 Open Problems and Challenges

Several significant challenges remain unresolved and present opportunities for future research. The fundamental problem of dataset scarcity in classical Arabic religious texts requires ongoing attention, with potential solutions including collaborative data collection initiatives with Islamic institutions and the development of synthetic data generation techniques that maintain religious authenticity.

The challenge of real-time validation of AI-generated religious content by qualified scholars represents a critical bottleneck that could benefit from automated screening systems and distributed validation networks. The development of standardized evaluation metrics for AI applications in religious contexts remains an open research problem with significant implications for the broader field.

Technical challenges include the optimization of multilingual AI systems for classical and contemporary Arabic processing, the development of cultural context-aware recommendation algorithms, and the creation of scalable systems that can maintain religious authenticity while serving global Muslim communities with diverse cultural and linguistic backgrounds.

7.5 Final Remarks

The development of Zikr demonstrates that advanced AI techniques can be successfully applied to religious education and community engagement while maintaining cultural sensitivity and religious authenticity. The platform addresses real needs within Muslim communities for safe, educationally focused digital spaces and provides a model for respectful AI implementation in traditional knowledge domains.

The project's success in bridging academic AI research with practical religious applications validates the potential for transformative technology in Islamic education. By combining proven machine learning techniques with user-centric design and community-driven validation, Zikr establishes a foundation for continued innovation in AI-enhanced religious education platforms.

The comprehensive integration of multiple AI functionalities within a social learning environment demonstrates the feasibility of holistic educational platforms that transcend the limitations of single-purpose applications. This achievement provides valuable insights for the broader EdTech community and establishes new possibilities for AI-driven educational innovation in culturally sensitive contexts.

As the platform continues to evolve, the lessons learned from its development will inform future projects at the intersection of artificial intelligence, religious studies, and educational technology, contributing to the growing field of culturally-aware AI systems that serve diverse global communities while respecting traditional values and authentic knowledge systems.

References

Alan, A. Y., Karaarslan, E. and Aydin, Ö. (2024), 'A rag-based question answering system proposal for understanding islam: Mufassirqas llm', *arXiv preprint arXiv:2401.15378* .

Almanea, M. (2024), 'Deep learning in written arabic linguistic studies: A comprehensive survey', *IEEE Access* .

Alrefaie, M. T. (2019), 'Arabic stop words'.

Basem, M., Oshallah, I., Hikal, B., Hamdi, A. and Mohamed, A. (2024), 'Optimized quran passage retrieval using an expanded qa dataset and fine-tuned language models', *arXiv preprint arXiv:2412.11431* .

Bashir, M. H., Azmi, A. M., Nawaz, H., Zaghouani, W., Diab, M., Al-Fuqaha, A. and Qadir, J. (2023), 'Arabic natural language processing for qur'anic research: a systematic review', *Artificial Intelligence Review* **56**(7), 6801–6854.

Elsaadany, A. (2022), 'Fatawa dataset', https://www.kaggle.com/datasets/abdallahelsaadany/fatawa. Accessed: 2025-06-09.

Habash, N. (2022), Arabic natural language processing, *in* 'Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts', pp. 9–10.

Mahmoud, S., Saif, O., Nabil, E., Abdeen, M., ElNainay, M. and Torki, M. (2022), 'Ar-sanad 280k: a novel 280k artificial sanads dataset for hadith narrator disambiguation', *Information* **13**(2), 55.

Nigar, N., Wajid, A., Ajagbe, S. A. and Adigun, M. O. (2023), 'An intelligent framework based on deep learning for online quran learning during pandemic', *Applied Computational Intelligence and Soft Computing* **2023**(1), 5541699.

Ramzy, A., Torki, M., Abdeen, M., Saif, O., ElNainay, M., Alshanqiti, A. and Nabil, E. (2023), 'Hadiths classification using a novel author-based hadith classification dataset (abcd)', *Big Data and Cognitive Computing* **7**(3), 141.

Shohoud, Y., Shoman, M. and Abdelazim, S. (2023), 'Quranic conversations: Developing a semantic search tool for the quran using arabic nlp techniques', *arXiv preprint arXiv:2311.05120* .

Slamet, C., Rahman, A., Ramdhani, M. A. and Darmalaksana, W. (2016), 'Clustering the verses of the holy qur'an using k-means algorithm', *Asian Journal of Information Technology* **15**(24), 5159–5162.

Sulistio, B., Ramadhan, A., Abdurachman, E., Zarlis, M. and Trisetyarso, A. (2024), 'The utilization of machine learning on studying hadith in islam: A systematic literature review', *Education and Information Technologies* **29**(5), 5381–5419.

Touati-Hamad, Z., Laouar, M. R. and Bendib, I. (2020), Quran content representation in nlp, *in* 'Proceedings of the 10th International Conference on Information Systems and Technologies', pp. 1–6.