Musafir

CSE 4510 - Software Development Lab

Team Members

Md. Mahiul Kabir	220041109
Sheikh Mosheul Akbar	220041137
Nayeem Hossain Ahad	220041139
Mubtasim Hasan Ahan	220041123
Sohom Sattyam	220041141
Sohan Nur	220041151

Project Motivation:

Muslims currently navigate fragmented digital tools, switching between apps like Muslim Pro for prayer times, Tarteel and Quran.com for recitations, Sunnah.com for Hadith, and scattered platforms for community interaction, resulting in inefficiencies, storage overload, and disjointed experiences. Musafir eliminates this app fatigue by consolidating core functions—salah/suhur/iftar times, Quran tracker, alerts, daily verses/duas, Qibla compass, nasheeds, and AI chatbot—with community features like subscription channels, blogging for stories/knowledge/quotes, and QnA forums. This addresses the growing demand for a single, intuitive platform serving the Ummah's spiritual, practical, and social needs.

Project Objectives:

Musafir centralizes verified Islamic resources and fosters safe global interactions through a robust mobile app, providing instant access to prayer utilities, knowledge bases, and personalized community feeds. Primary goals include seamless consolidation of 13+ features, Sharia-compliant content delivery, and enhanced user retention via real-time alerts and moderated social tools. Expected outcomes feature reduced app-switching, daily spiritual engagement, stronger community bonds, and scalable growth for features like local halal/salat updates.

Major Functional Requirements:

- Core Functions
    - Salah/Suhur-Iftar Times
    - Quran Tracker
    - Alerts
    - Daily Verse/Duas
- GPS-based prayer calculations (Aladhan.com API with multiple methods), tilawat/translation/tafseer progress tracking (Quran.com API v4), push

notifications for prayers/forbidden times, rotating masnoon duas and verses from verified sources.

- Community Features
  - Channels + Subscriptions
  - Blogging (Stories/Knowledge/Quotes)
  - QnA

- Moderated feeds for local updates (halal info, salat adjustments), user-generated posts including waz/sermons/lectures, interactive forums for questions.

- Miscellaneous
  - Qibla Compass
  - Nasheed
  - Chatbot

- Client-side sensor/GPS direction finder, curated audio streams (Muslim Central/Archive.org), closed-context LLM for accurate Islamic queries without hallucinations.

System Architecture:

Musafir adopts a scalable modular architecture for maintainability and extensibility:

- Utility Engine (Client-Side): Handles real-time GPS, compass sensors, prayer algorithms, and local notifications.
- Knowledge CMS (Backend): Manages Quran audio/text/tafseer, Hadith library (Sunnah.com indexed), duas database with structured metadata.
- Interactive Service Layer: Powers secure auth, dynamic social feeds, subscription logic, and AI chatbot interfacing with pre-approved datasets.

This backend-driven design ensures data integrity, real-time updates, and API reliability across modules.

Chosen Tech Stack & Justification:

Component	Technologies
Frontend (Mobile)	React Native for iOS/Android cross-platform deployment.Musafir.pdf
Backend	Node.js for APIs/real-time processing, Java for microservices.Musafir.pdf
Database	MongoDB (NoSQL) for JSON-based Quranic/Hadith data.Musafir.pdf
Development Tools	Git (version control), VS Code (IDE), Postman (API testing), Figma (UI/UX prototyping).Musafir.pdf
Integrations	Quran.com v4, Aladhan.com, Sunnah.com, Muslim Central APIs.Musafir.pdf

Key Stakeholders:

- ***Primary Users:***
    - General Muslim Ummah for daily utilities; students of knowledge for advanced tafseer/Hadith tools.
- ***Secondary Users:***
    - Community moderators/admins for channel management; scholars for content verification.
- ***Development Team:***
    - 6-member squad handling coding, QA, Sharia compliance; academic supervisors for review.
- ***Ecosystem Partners:***
    - API providers (Quran.com, Aladhan) and content archives ensuring authenticity.

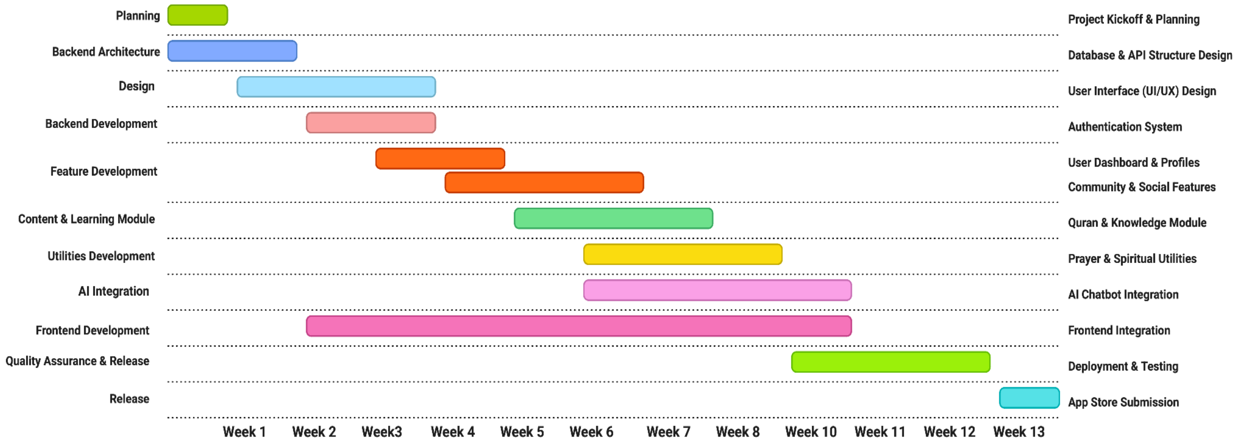
Critical Challenges:

Challenge	Description	Mitigation Strategy
UI/UX Density	Fitting 13+ features into intuitive interface without clutter.	Iterative prototyping, modular navigation, feature prioritization.Musafir.pdf
Data Verification	Absolute accuracy for Quran/Hadith/AI responses (zero error tolerance).	Scholar-vetted sources, closed-context AI, Niyyat-driven reviews.Musafir.pdf
Cross-Compatibility	Seamless performance on low-end Android to high-end iOS.	Optimization testing, lightweight streaming.Musafir.pdf
API Reliability	Dependence on external Islamic databases.	Fallback caches, multiple calculation methods, vetted providers.Musafir.pdf

Conflicting Requirements and Trade-offs

- Audio Quality vs. App Size: High-fidelity tilawat/nasheed inflates downloads—resolved via stream-by-default, download-on-demand.
- AI Power vs. Theological Safety: LLM hallucinations risk—limited to verified closed-context database, no open scraping.
- Social Engagement vs. Moderation: Vibrant communities vs. safety—strict admin controls, real-time filtering.

13-Week Project Timeline Overview



Planning
Backend Architecture
Design
Backend Development
Feature Development
Content & Learning Module
Utilities Development
AI Integration
Frontend Development
Quality Assurance & Release
Release
Project Kickoff & Planning
Database & API Structure Design
User Interface (UI/UX) Design
Authentication System
User Dashboard & Profiles
Community & Social Features
Quran & Knowledge Module
Prayer & Spiritual Utilities
AI Chatbot Integration
Frontend Integration
Deployment & Testing
App Store Submission
Week 1
Week 2
Week3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 10
Week 11
Week 12
Week 13