

Project Information

Field	Value
Project Name	Ramadan Helper - Islam...
Version	3.0 (Final Version)
Author	Chiheb Bahri
Date	January 14, 2026
Repository	https://github.com/che...
License	MIT

1. Executive Summary

Ramadan Helper is a comprehensive full-stack Islamic web application designed to support Muslims in their spiritual journey during Ramadan and beyond. The application provides a RESTful API backend built with FastAPI (Python) that serves multiple services including AI-powered Islamic Q&A, personalized dua generation, real-time chat with Imams, Islamic video search, and a Ramadan events platform specifically for Tunisia.

The system implements industry-standard JWT (JSON Web Token) authentication with email verification, role-based access control (User, Imam, Admin), and follows REST architectural principles for scalable and maintainable API design.

2. Project Description

2.1 Purpose

The Ramadan Helper application aims to:

- Provide Muslims with easy access to Islamic knowledge through AI-powered search

Backend Framework	FastAPI	0.104.1
ASGI Server	Uvicorn	0.24.0
Database ORM	SQLAlchemy	2.0.23
Database	SQLite	3.x
Authentication	JWT (python-jose)	-
Password Hashing	bcrypt	5.0.0
HTTP Client	httpx	0.25.2
AI Service	Groq API	0.4.2
Data Validation	Pydantic	2.5.2

Environment	python-dotenv	1.0.0
Frontend	HTML5, TailwindCSS, Ja...	-

3.2 Architecture Pattern

3.3 Project Structure

```
ramadan-webservice-project- /
```

4. Database Design

4.1 Entity-Relationship Diagram

4.2 Database Models Summary

Model	Description	Key Fields
User	Application users (use...)	email, password_hash, ...
Imam	Islamic scholars for c...	name, expertise, is_av...
DuaHistory	Generated duas history	category, dua_text_en,...
Conversation	Chat threads between u...	user_id, imam_id, topic
Message	Individual chat messages	sender_type, message_t...
Event	Ramadan events in Tunisia	title, city, category,...
AIAnalysis	AI-powered question an...	question, ayah, hadith...
TokenBlacklist	Revoked JWT tokens	token_jti, expires_at
**EmailVerificationTok...	Email verification codes	token, expires_at, used
PasswordResetToken	Password reset tokens	token, expires_at, used

5. API Endpoints Documentation

5.1 Authentication Endpoints (`/api/auth`)

Method	Endpoint	Description	Auth Required
POST	`/signup`	Register new user account	5.2 Dua Generator

Endpoints (`/api/dua`)

Method	Endpoint	Description	Auth Required
GET	`/categories`	Get available dua cate...	Dua Categories:

GET	`/imams`	List available imams	5.4	AI	Analyzer
-----	----------	----------------------	-----	----	----------

Endpoints

(`/api/analyzer`)

Method	Endpoint	Description	Auth Required
POST	`/`	Analyze Islamic question	Features:

GET	`/cities`	Get 24 Tunisian cities	Supported Cities (24 Tunisian Governorates):
-----	-----------	------------------------	--

POST	`/search`	Search YouTube for Isl...	5.7 Admin Panel
------	-----------	---------------------------	-----------------

Endpoints

(`/api/admin`)

Method	Endpoint	Description	Auth Required
GET	`/users`	List all users with fi...	---

6. Security Implementation

6.1 Authentication Flow

6.2 JWT Token Structure

Access Token (30 minutes expiry):

```
{
  "sub": "user@example.com",
  "user_type": "user",
  "exp": 1736884800,
  "iat": 1736883000,
  "jti": "uuid-v4-token-id",
  "type": "access"
}
```

Refresh Token (7 days expiry):

```
{
  "sub": "user@example.com",
  "user_type": "user",
  "exp": 1737489600,
  "iat": 1736884800,
  "jti": "uuid-v4-token-id",
  "type": "refresh"
}
```

6.3 Security Features

Feature	Implementation
Password Hashing	bcrypt with salt
Token Type	JWT Bearer tokens
Token Expiry	Access: 30min, Refresh...
Token Revocation	Blacklist table with JTI
Email Verification	6-digit code, 24h expiry
Brute Force Protection	Account lockout after ...
Lockout Duration	15 minutes
CORS	Configured for cross-o...
Role-Based Access	User, Imam, Admin roles

7. API Request/Response Examples

7.1 User Signup

Request:

Content-Type: application/json

```
{
  "email": "user@example.com",
  "password": "SecurePass123!",
  "name": "Ahmad"
}
```

Response (200 OK):

```
{
  "message": "Verification email sent. Please check your inbox.",
  "email": "user@example.com",
  "requires_verification": true,
  "demo_token": null
}
```

7.2 User Login

Request:

Content-Type: application/json

```
{
  "email": "user@example.com",
  "password": "SecurePass123!"
}
```

Response (200 OK):

```
{  
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
    "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
    "token_type": "bearer",  
    "expires_in": 1800,  
    "user": {  
        "email": "user@example.com",  
        "name": "Ahmad",  
        "user_type": "user",  
        "is_verified": true  
    }  
}
```

7.3 Generate Dua

Request:

POST /dua

```
Content-Type: application/json  
  
{  
    "category": "health",  
    "context": "I am struggling with anxiety and seeking peace",  
    "email": "user@example.com"  
}
```

Response (200 OK):

```
{  
    "id": 1,  
    "category": "health",  
    "context": "I am struggling with anxiety and seeking peace",  
    "dua_text_en": "O Allah, I seek refuge in You from anxiety and sorrow...",  
    "dua_text_ar": "اللهم اني أسألك رحمةك من كل شر، وانت أرحم الراحمين"  
}
```

7.4 Create Event

Request:

POST /event

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...  
Content-Type: application/json  
  
{  
    "title": "Community Iftar",  
    "description": "Join us for a community iftar gathering",  
    "city": "Tunis",  
    "location": "Mosque Al-Zaytuna",  
    "category": "iftar",  
    "event_date": "2026-03-15",  
    "start_time": "18:30",  
    "organizer_name": "Islamic Center",  
    "organizer_contact": "+216 71 123 456",  
    "is_featured": false  
}
```

Response (200 OK):

```
{
  "id": 1,
  "title": "Community Iftar",
  "description": "Join us for a community iftar gathering",
  "city": "Tunis",
  "location": "Mosque Al-Zaytuna",
  "category": "iftar",
  "event_date": "2026-03-15",
  "start_time": "18:30",
  "organizer_name": "Islamic Center",
  "is_verified": false,
  "is_featured": false,
  "created_at": "2026-01-14T12:00:00"
}
```

7.5 AI Analysis

Request:

POST /api/ai-analysis

```
Content-Type: application/json

{
  "question": "What does Islam say about patience?",
  "email": "user@example.com"
}
```

Response (200 OK):

```
{
  "question": "What does Islam say about patience?",
  "ayah": {
    "surah": "Al-Baqarah",
    "surah_number": 2,
    "verse_number": 153,
    ...
  }
}
```

8. Events Monetization Model

8.1 Pricing Structure (Tunisia)

Listing Type	Price	Features
Basic	20 TND	Standard listing in se...
Featured	50 TND	Premium placement, hig...

8.2 Event Categories

Category	Emoji	Description
Iftar	---	

9. External Services Integration

9.1 Groq AI (LLM Service)

- **Purpose:** AI-powered dua generation and Islamic Q&A

```
git clone https://github.com/cheehub213/ramadan-webservice-project-.git
cd ramadan-webservice-project-

# 2. Create virtual environment
python -m venv .venv
.venv\Scripts\activate # Windows
source .venv/bin/activate # Linux/Mac

# 3. Install dependencies
cd backend
pip install -r requirements.txt

# 4. Configure environment
# Create .env file with:
# GROQ_API_KEY=your_api_key
# SMTP_USER=your_email
# SMTP_PASSWORD=your_app_password

# 5. Run backend
python main.py

# 6. Run frontend (new terminal)
cd "app/schemas/frontend webservice site"
python -m http.server 8080
```

10.3 Access Points

Service	URL
Backend API	http://localhost:8000
API Documentation (Swa...)	http://localhost:8000/...
API Documentation (ReDoc)	http://localhost:8000/...
Frontend Application	http://localhost:8080/...

11. Testing

11.1 API Testing with Swagger UI

Access <http://localhost:8000/docs> for interactive API testing:

- Click "Authorize" button

`test_simple_app.py`	Basic endpoint tests
`test_bilingual_api.py`	Bilingual dua tests
`test_chat_dua.py`	Chat and dua integrati...
`test_semantic_search.py`	AI semantic search tests
`test_imam_registratio...`	Imam workflow tests

12. Error Handling

12.1 HTTP Status Codes

Code	Meaning	Example
200	Success	Request completed
201	Created	Resource created
400	Bad Request	Invalid input
401	Unauthorized	Invalid/missing token
403	Forbidden	Insufficient permissions
404	Not Found	Resource doesn't exist
422	Validation Error	Invalid data format
500	Server Error	Internal error

12.2 Error Response Format

```
{
  "detail": "Error message description"
}
```

13. Conclusion

The Ramadan Helper RESTful API provides a comprehensive backend solution for Islamic spiritual guidance applications. Key achievements include:

- **56+ API endpoints** across 7 route modules