**API Documentation**

## Module: Authentication (auth.js)

**Base Path**

bash

CopyEdit

/api/auth

**1. Register a New User**

**Endpoint**

**POST** /api/auth/register

**Description**

This route is responsible for registering a new user. It validates the input fields, securely hashes the password, stores the user information in the database, and returns a JSON Web Token (JWT) for authentication purposes.

**Request Headers**

* None (This is a public route and does not require authentication.)

**Request Body**

* Content-Type: application/json

Example:

json

CopyEdit

{

"name": "John Doe",

"email": "john@example.com",

"password": "password123"

}

**Validations**

* name: Must not be empty.
* email: Must be a valid email address.
* password: Must be at least 6 characters long.

**Responses**

| **Status** | **Meaning** | **Response Example** |
| --- | --- | --- |
| 200 OK | Registration successful | { "token": "JWT\_TOKEN\_HERE" } |
| 400 Bad Request | Validation errors or Email already in use | { "errors": [{ "msg": "Email in use" }] } |
| 500 Server Error | Server encountered an unexpected condition | "Server error" |

**Flow Summary**

1. Validate the incoming request fields.
2. Check if a user with the provided email already exists.
3. Hash the password securely using bcrypt (with 10 salt rounds).
4. Save the new user to the MongoDB database.
5. Generate a JWT token using the user's ID as the payload.
6. Return the generated token to the client.

**2. Authenticate a User (Login)**

**Endpoint**

**POST** /api/auth/login

**Description**

This route allows an existing user to log in by verifying their email and password. Upon successful authentication, it returns a JWT token which can be used for authorized access to protected resources.

**Request Headers**

* None (This is a public route and does not require authentication.)

**Request Body**

* Content-Type: application/json

Example:

json

CopyEdit

{

"email": "john@example.com",

"password": "password123"

}

**Validations**

* email: Must be a valid email address.
* password: Must be provided.

**Responses**

| **Status** | **Meaning** | **Response Example** |
| --- | --- | --- |
| 200 OK | Login successful | { "token": "JWT\_TOKEN\_HERE" } |
| 400 Bad Request | Invalid credentials provided | { "errors": [{ "msg": "Invalid credentials" }] } |
| 500 Server Error | Server encountered an unexpected condition | "Server error" |

**Flow Summary**

1. Validate the input fields.
2. Find a user by the provided email address.
3. If the user is not found, return an "Invalid credentials" error.
4. Compare the provided password with the stored hashed password using bcrypt.
5. If the passwords do not match, return an "Invalid credentials" error.
6. If authentication is successful, generate a JWT token.
7. Return the token to the client.

**3. Retrieve Current Authenticated User Information**

**Endpoint**

**GET** /api/auth/me

**Description**

This route fetches the authenticated user's basic information. It excludes sensitive details such as the password.

**Request Headers**

* Authorization: Bearer <your\_jwt\_token\_here>

**Request Body**

* None

**Responses**

| **Status** | **Meaning** | **Response Example** |
| --- | --- | --- |
| 200 OK | Successfully retrieved user information |  |

json

CopyEdit

{

"\_id": "609b1e4e2b3e5a3c4427b456",

"name": "John Doe",

"email": "john@example.com",

"\_\_v": 0

}

| 401 Unauthorized | Missing or invalid token | { "msg": "Token is not valid" } | | 404 Not Found | User not found | { "msg": "User not found" } | | 500 Server Error | Server encountered an unexpected condition | "Server error" |

**Flow Summary**

1. Verify the JWT token using authentication middleware.
2. Extract the user ID from the token's payload.
3. Query the database for the user by ID, excluding the password field.
4. If the user is found, return the user details.
5. If no user is found, return an appropriate error response.

**Postman Test Scripts (Examples)**

These examples can be used for testing the endpoints using Postman:

**1. Register a New User**

http

CopyEdit

POST http://localhost:5000/api/auth/register

Headers:

Content-Type: application/json

Body:

{

"name": "John Doe",

"email": "john@example.com",

"password": "password123"

}

**2. Login User**

http

CopyEdit

POST http://localhost:5000/api/auth/login

Headers:

Content-Type: application/json

Body:

{

"email": "john@example.com",

"password": "password123"

}

*Note: Save the returned token for accessing protected routes.*

**3. Retrieve Current User Information**

http

CopyEdit

GET http://localhost:5000/api/auth/me

Headers:

Authorization: Bearer <your\_jwt\_token\_here>

## Module: Azkar (Supplications) (azkar.js)

### Base Path

bash

CopyEdit

/api/azkar

## 1. Retrieve List of Azkar (Adhkar)

### Endpoint

**GET** /api/azkar

### Description

This route retrieves a collection of Azkar (Islamic remembrances or supplications) based on an optional category filter. The user must be authenticated to access this route, which requires a valid JSON Web Token (JWT).

### Request Headers

* Authorization: Bearer <your\_jwt\_token\_here>

### Query Parameters (Optional)

| **Parameter** | **Description** | **Example** |
| --- | --- | --- |
| type | Specifies the type or category of Azkar to fetch | m, e, as, t, bs, wu, qd, pd |

**Note:**  
If the type query parameter is omitted, all available Azkar will be returned.

#### Shortcodes for type Parameter

| **Shortcode** | **Meaning** |
| --- | --- |
| m | Morning Azkar |
| e | Evening Azkar |
| as | After Salah (prayer) Azkar |
| t | Tasbeeh (glorification) |
| bs | Before Sleep Azkar |
| wu | Wudu (ablution) Azkar |
| qd | Qunut Dua |
| pd | Post-Dua Supplications |

### Request Examples

* **Retrieve All Azkar**

bash

CopyEdit

GET http://localhost:5000/api/azkar

* **Retrieve Only Morning Azkar**

bash

CopyEdit

GET http://localhost:5000/api/azkar?type=m

* **Retrieve Only Evening Azkar**

bash

CopyEdit

GET http://localhost:5000/api/azkar?type=e

### Response Format

* Content-Type: application/json

Example:

json

CopyEdit

{

"category": "m",

"count": 15,

"items": [

"Dua 1...",

"Dua 2...",

"Dua 3...",

"... etc ..."

]

}

### Responses Summary

| **Status** | **Meaning** | **Response Example** |
| --- | --- | --- |
| 200 OK | Successfully retrieved Azkar | { "category": "m", "count": 15, "items": [...] } |
| 401 Unauthorized | Token is missing or invalid | { "msg": "Token is not valid" } |

## Postman Test Scripts (Examples)

These examples demonstrate how to test the /api/azkar endpoint using Postman:

### 1. Retrieve All Azkar (No Filter)

http

CopyEdit

GET http://localhost:5000/api/azkar

Headers:

Authorization: Bearer <your\_jwt\_token\_here>

### 2. Retrieve Morning Azkar

http

CopyEdit

GET http://localhost:5000/api/azkar?type=m

Headers:

Authorization: Bearer <your\_jwt\_token\_here>

### 3. Retrieve Evening Azkar

http

CopyEdit

GET http://localhost:5000/api/azkar?type=e

Headers:

Authorization: Bearer <your\_jwt\_token\_here>

## Module: Calendar (calendar.js)

**Base Path**

bash

CopyEdit

/api/calendar

**1. Retrieve Today's Islamic Calendar Events**

**Endpoint**

**GET** /api/calendar/today

**Description**

This route provides today’s Islamic calendar details for an authenticated user. The response includes both the Gregorian and Hijri dates, the corresponding weekday, and a list of any significant Islamic events occurring on that day.

Authentication is required, and the user must present a valid JSON Web Token (JWT) to access this endpoint.

**Request Headers**

| **Header** | **Value** |
| --- | --- |
| Authorization | Bearer <your\_jwt\_token\_here> |

**Request Parameters**

* No body parameters are required.
* No query parameters are required.
* All necessary details, such as location settings and calendar type, are handled internally within the system.

**Request Example**

http

CopyEdit

GET http://localhost:5000/api/calendar/today

Headers:

Authorization: Bearer <your\_jwt\_token\_here>

**Response Format**

* Content-Type: application/json

Example Response:

json

CopyEdit

{

"gregorianDate": "2025-04-28",

"hijriDate": "29 Ramadan 1446",

"weekday": "Monday",

"events": [

"Laylat al-Qadr",

"Itikaf Begins"

]

}

**Responses Summary**

| **Status Code** | **Meaning** | **Response Example** |
| --- | --- | --- |
| 200 OK | Successfully fetched today's calendar data | { gregorianDate, hijriDate, weekday, events } |
| 401 Unauthorized | Authentication token is missing or invalid | { "msg": "Token is not valid" } |
| 500 Server Error | Internal server error or external API failure | { "msg": "Server error" } |

**Postman Test Script Example**

**1. Retrieve Today's Calendar Events**

http

CopyEdit

GET http://localhost:5000/api/calendar/today

Headers:

Authorization: Bearer <your\_jwt\_token\_here>

**Expected Outcome:**  
The API should return the current day’s Gregorian and Hijri dates, the name of the weekday, and any significant Islamic events happening today.