

API Documentation

(Due to time constraints this is the placeholder API doc.)

This API provides functionality for user authentication and profile management, including support for both local and future social login methods (Google, Facebook, Twitter, GitHub). Users can manage their profiles, set their profiles to public or private, and view other users' profiles based on their access permissions.

Base URL : `http://localhost:3000/api`

Authentication :

Local Authentication:

Register User

Endpoint: `/auth/register`

Method: POST

Description: Register a new user.

Request Body:

```
{
  "email": "user@example.com",
  "password": "password123"
}
```

Response:

```
{
  "message": "User registered successfully"
}
```

Login User

Endpoint: `/auth/login`

Method: POST

Description: Authenticate a user and return a JWT token.

Request Body:

```
{  
  "email": "user@example.com",  
  "password": "password123"  
}
```

Response:

```
{  
  "token": "jwt-token"  
}
```

Profile Management :

View Own Profile

Endpoint: /profiles/profile

Method: GET

Description: Get the authenticated user's own profile.

Headers: x-auth-token: jwt-token

Response:

```
{  
  "_id": "user-id",  
  "email": "user@example.com",  
  "name": "User Name",  
  "bio": "User bio",  
  "phone": "1234567890",  
  "photo": "photo-url",  
  "isPublic": true  
}
```

Edit Profile

Endpoint: /profiles/profile

Method: PUT

Description: Edit the authenticated user's profile.

Headers: x-auth-token: jwt-token

Request Body (example):

```
{  
  "name": "Updated Name",
```

```
"bio": "Updated bio",
"phone": "0987654321",
"photo": "new-photo-url",
"email": "newemail@example.com",
"password": "newpassword123",
"isPublic": false
}
```

Response:

```
{
  "_id": "user-id",
  "email": "newemail@example.com",
  "name": "Updated Name",
  "bio": "Updated bio",
  "phone": "0987654321",
  "photo": "new-photo-url",
  "isPublic": false
}
```

View Public Profiles

Endpoint: /profiles/public

Method: GET

Description: List all public profiles.

Headers: x-auth-token: jwt-token

Response:

```
[
  {
    "_id": "user-id",
    "name": "Public User",
    "bio": "Public bio",
    "photo": "public-photo-url"
  },
  ...
]
```

View Specific Profile

Endpoint: /profiles/:id

Method: GET

Description: View a specific user's profile by ID.

x-auth-token: jwt-token

Response (if public or admin):

```
{
  "_id": "user-id",
  "name": "User Name",
  "bio": "User bio",
  "phone": "1234567890",
  "photo": "photo-url",
  "isPublic": true
}
```

Response (if private and not admin):

```
{
  "message": "Access denied"
}
```

View All Profiles (Admin Only)

Endpoint: /profiles/admin/all

Method: GET

Description: List all profiles (public and private) for admin users.

Headers: x-auth-token: jwt-token

Response:

```
[
  {
    "_id": "user-id",
    "name": "User Name",
    "bio": "User bio",
    "phone": "1234567890",
    "photo": "photo-url",
    "isPublic": true
  },
  ...
]
```

Error Handling :

Errors are returned with appropriate HTTP status codes and a JSON object containing an error message.

Example Error Response:

```
{  
  "message": "No token, authorization denied"  
}
```

Conclusion :

This API provides comprehensive endpoints for managing user authentication and profile settings, with provisions for future social login integrations. The design ensures that public and private profiles are appropriately handled, with robust authentication and authorization mechanisms to protect user data.