### API documentation

### Overview

This documentation outlines the endpoints, request/response formats, and error handling for the Flask-based user management API.

**Link:**

https://lively-robot-639422.postman.co/workspace/New-Team-Workspace~0f2e8529-ec91-42b9-b18e-22230892c845/collection/18950382-ecd5ae8f-6257-4078-80f2-7f1f9b6a69a5?action=share&creator=18950382

### Authentication

The API utilizes JWT (JSON Web Token) authentication. Clients must obtain a valid token through the /login endpoint and include it in the Authorization header for subsequent requests in the format Bearer <token>.

### Endpoints

#### /login (POST)

Description: Authenticates a user and returns a JWT token.

Request:

* Method: POST
* Content-Type: application/json
* Request Body:

JSON

{

"username": "string",

"password": "string"

}

Response:

Success (200):

JSON

{

"token": "string"

}

Error (401):

JSON

{

"message": "Invalid credentials"

}

#### /create\_tables (GET)

Description: Creates the necessary database tables.

Request:

* Method: GET
* Authorization: Bearer <token>

Response:

Success (201):

JSON

{

"message": "Tables created successfully"

}

Error (401):

JSON

{

"message": "Token is missing!"

}

or

JSON

{

"message": "Token is invalid!"

}

#### /register (POST)

Description: Registers a new user.

Request:

* Method: POST
* Content-Type: application/json
* Authorization: Bearer <token>
* Request Body:

JSON

{

"first\_name": "string",

"last\_name": "string",

"email": "string",

"password": "string",

"phone\_number": "string"

}

Response:

Success (201):

JSON

{

"message": "User registration successful"

}

Error (400):

JSON

{

"error": "Missing required fields"

}

or

JSON

{

"error": "Invalid email format"

}

or

JSON

{

"error": "Invalid phone format"

}

or

JSON

{

"error": "Invalid names"

}

or

JSON

{

"error": "Email already exists"

}

Error (500):

JSON

{

"error": "Server error occurred"

}