

# End point simulation document

## 1. Introduction

This document provides detailed endpoint documentation for the **Authentication Module** of the virtual *payment System*.

The system was developed using **Node.js**, **Express.js**, and **MySQL**, and simulates secure user registration and login.

The endpoints are designed under the following base URL:

`http://localhost:5000/v1/api/`

The two primary endpoints for authentication are:

1. **/auth** – Handles user registration and login.
2. **/users** – Manages retrieval of registered user data.

The goal of these endpoints is to allow users to securely register, log in, and verify identity via JWT tokens.

## 2. System Overview

The authentication module is a key part of the platform. It ensures that each user interacting with the system is properly verified before accessing sensitive functionalities like wallet services or virtual card generation.

### Technologies Used:

- **Node.js** – Server-side environment for building the API.
  - **Express.js** – Framework for defining routes and handling requests.
  - **MySQL** – Database for storing user information.
  - **bcryptjs** – Library for hashing passwords.
  - **jsonwebtoken (JWT)** – Used to issue login tokens for session management.
  - **Postman** – Tool for testing API endpoints.
- 

## 3. Endpoint 1: User Registration

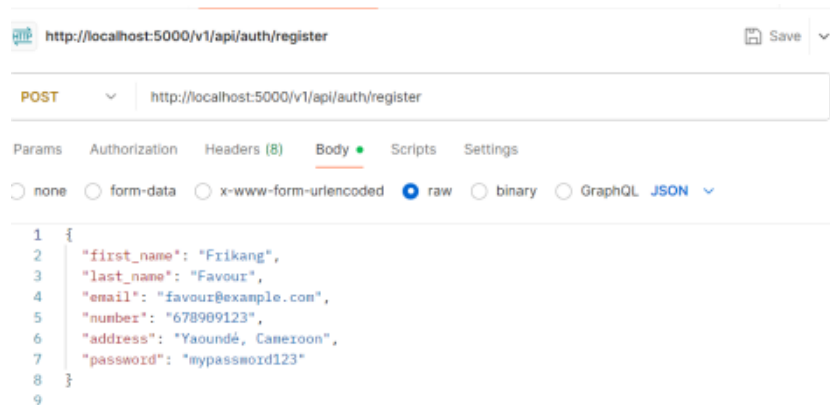
### Endpoint:

`POST /v1/api/auth/register`

### Purpose:

To create a new user account by collecting personal information and storing it securely in the database. Passwords are hashed using bcrypt before being saved.

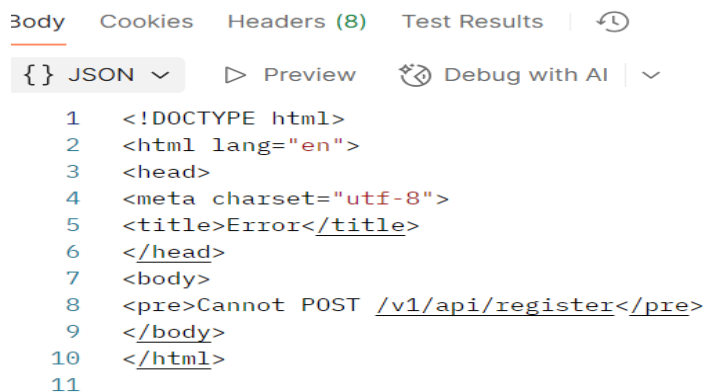
## Request Example (JSON):



## Response Example (Success):



## Response Example (Error):



## 4. Endpoint 2: User Login

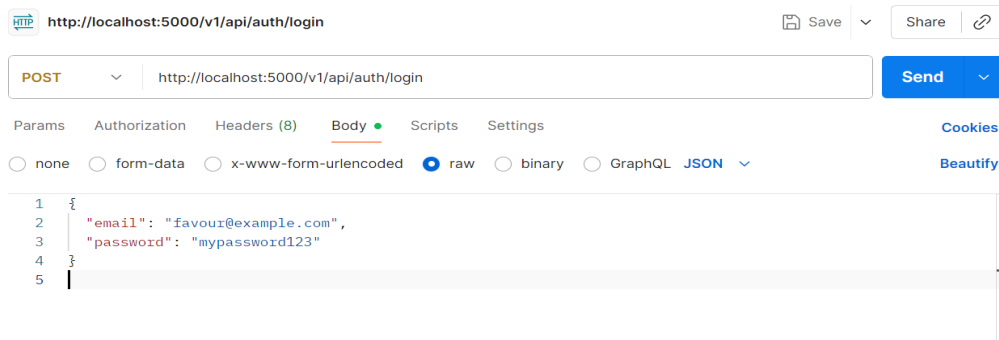
### Endpoint:

`POST /v1/api/auth/login`

### Purpose:

Authenticates existing users by verifying email and password. Upon successful verification, a **JWT token** is generated to maintain a secure session.

### Request Example (JSON):



### Response Example (Success):



### Process Flow:

1. The API receives the email and password.
2. It checks if the email exists in the database.
3. If found, bcrypt verifies the password.
4. On success, JWT creates a signed token valid for 1 hour.
5. The token is returned to the user for use in subsequent requests.

## 5. Endpoint 3: Retrieve All Users


### Endpoint:

GET /v1/api/users

### Purpose:

Displays all users currently stored in the system.  
Primarily for administrators or testing purposes.

### Response Example:

 http://localhost:5000/v1/api/users

GET

http://localhost:5000/v1/api/users

Send

ParamsAuthorizationHeaders (8)Body ●ScriptsSettings

BodyCookiesHeaders (7)Test Results🔄

200 OK • 29 ms • 377 B • 🌐 • ⋮

{ } JSON

PreviewVisualize

🔍📄🔗

```
1  [
2    {
3      "id": 1,
4      "first_name": "Frikang",
5      "last_name": "Favour",
6      "email": "favouri@example.com",
7      "phone_number": "678909123",
8      "address": "Yaoundé, Cameroon"
9    }
10 ]
```