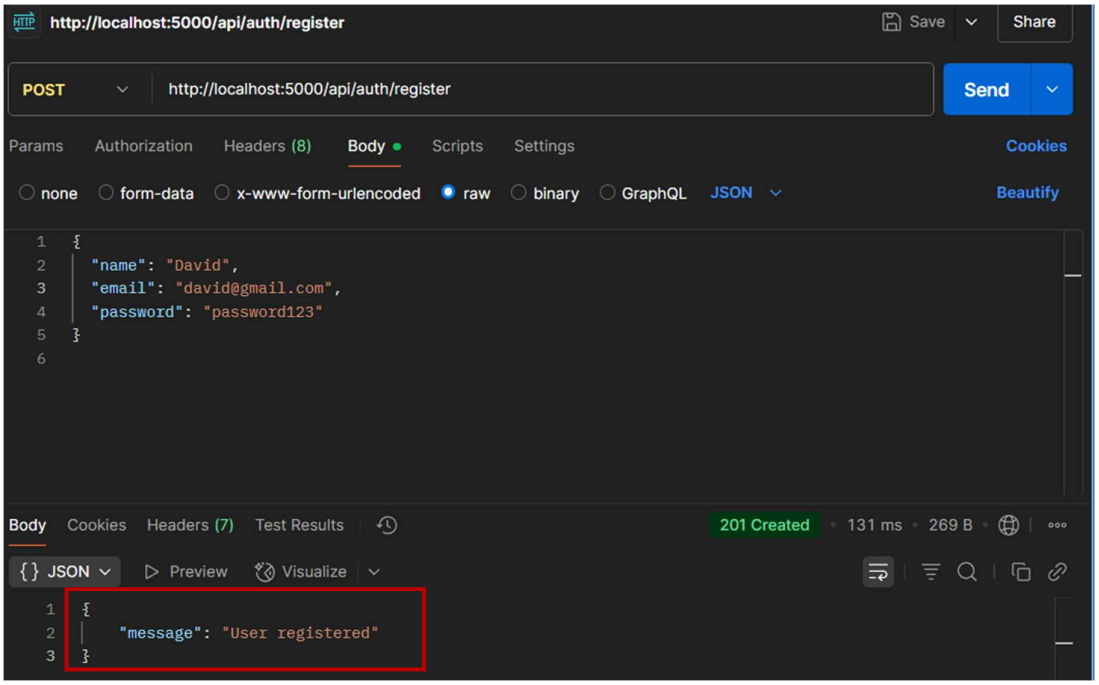
Sample Output for Digital Wallet System with Cash Management and Fraud Detection

1. **User Authentication and Session Management**

   A. User Registration

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

   Example:
   { Name: “David”, E-mail: david@gmail.com, Password: "password123"}

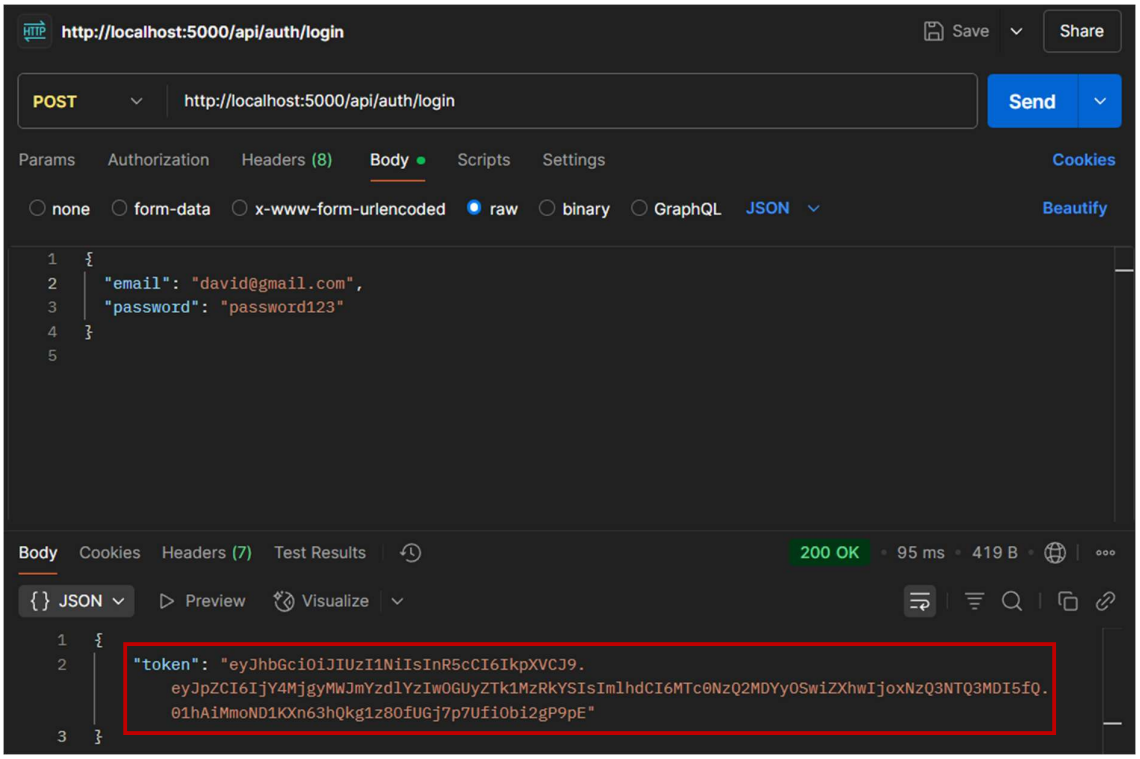


User is registered.

B. **User Login**

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

Example: {E-mail: david@gmail.com, Password: "password123"}

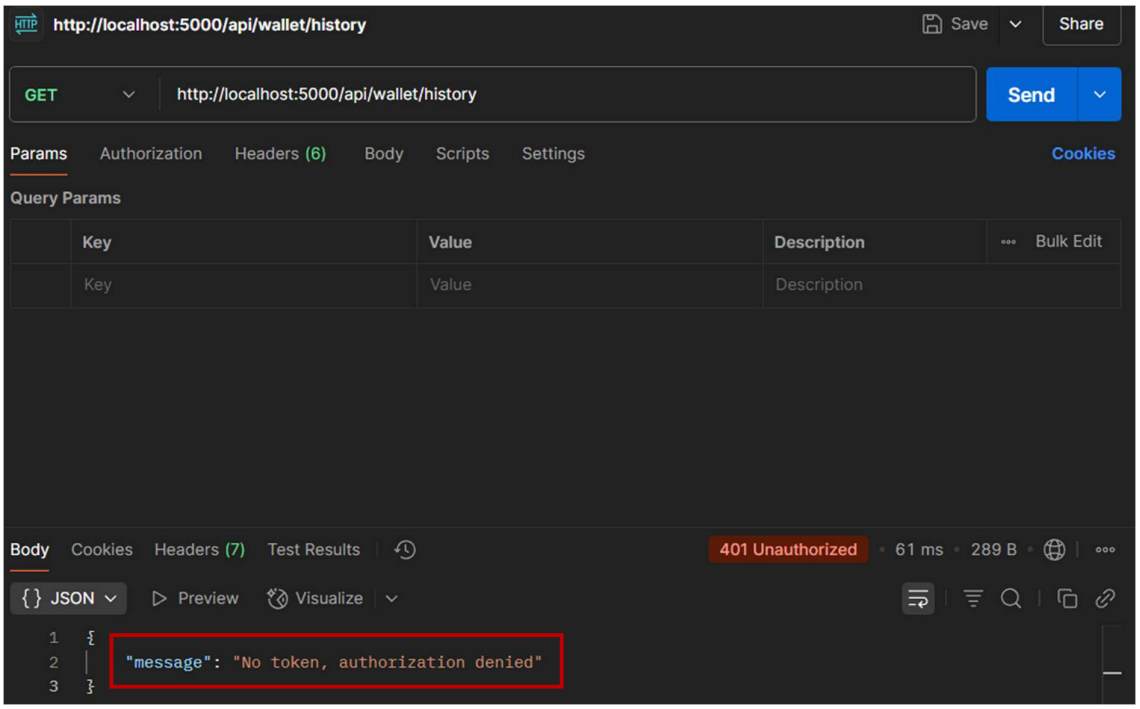


A JWT token is generated for authenticated access.

C. Access Protected Endpoint without Token

Endpoint: http://localhost:5000/api/wallet/history

Headers: None



Token is required, in the absence it shows "No token, authorization denied".

Token generated :
“eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjgyMWJmYzdlYzIwOGUyZTk1MzRkYSIsImlhdCI6MTc0NzQ2MDYyOSwiZXhwIjoxNzQ3NTQ3MDI5fQ.01hAiMmoND1KXn63hQkg1z8OfUGj7p7UfiObi2gP9pE”

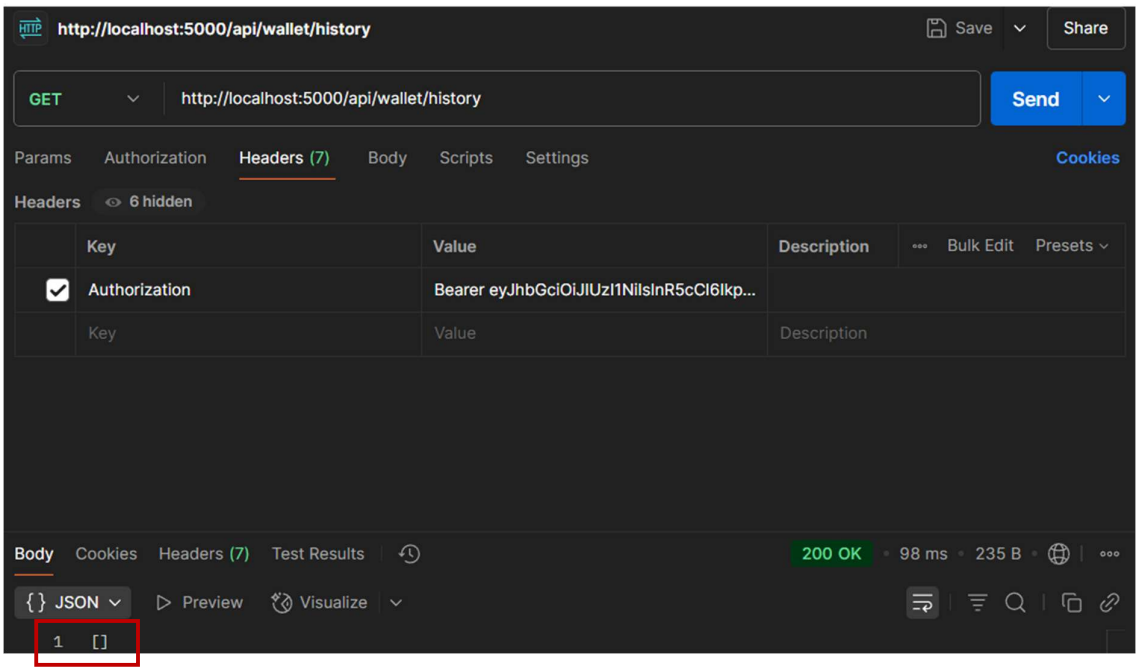
D. Access Protected Endpoint with Token

Endpoint: http://localhost:5000/api/wallet/history

Headers:

Key: Authorization

Value: "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjgyMWJmYzdlYzIwOGUyZTk1MzRkYSIsImlhdCI6MTc0NzQ2MDYyOSwiZXhwIjoxNzQ3NTQ3MDI5fQ.01hAiMmoND1KXn63hQkg1z8OfUGj7p7UfiObi2gP9pE"



Returns transaction history (empty at first)

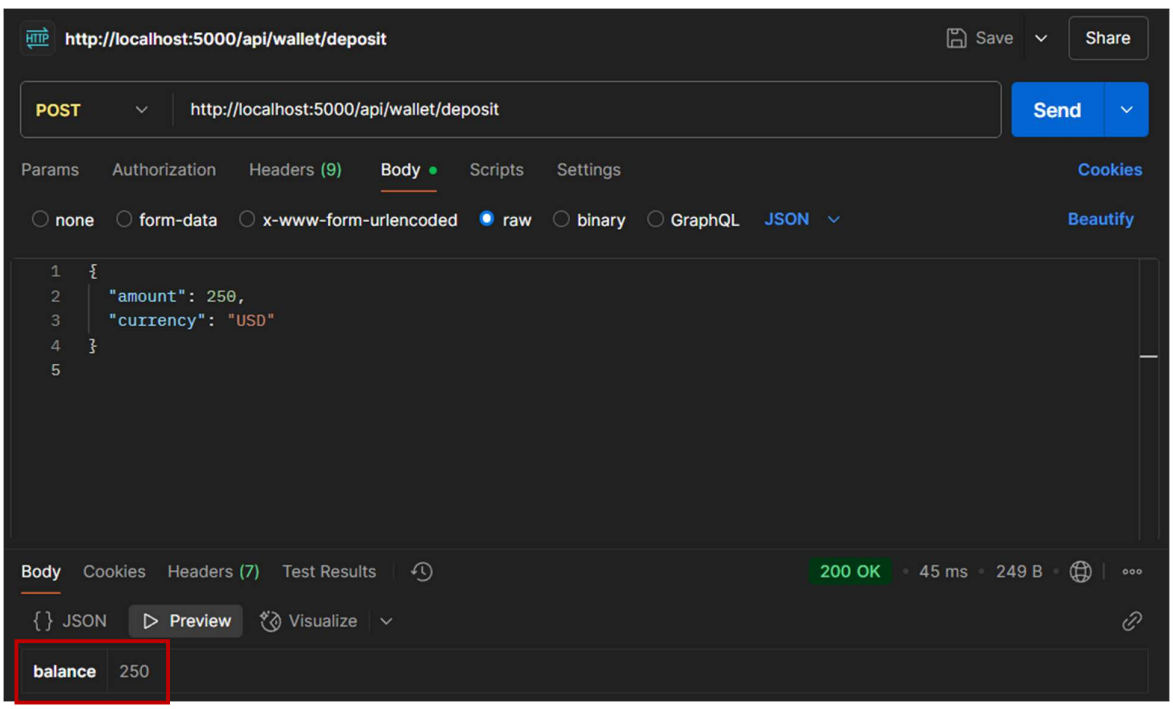
2. Wallet Operations

A. **Deposit Virtual Cash**

Endpoint: http://localhost:5000/api/wallet/deposit

Headers:
Key: Authorization
Value: “Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjgyMWJmYzdlYzIwOGUyZTk1MzRkYSIsImlhdCI6MTc0NzQ2MDYyOSwiZXhwIjoxNzQ3NTQ3MDI5fQ.01hAiMmoND1KXn63hQkg1z8OfUGj7p7UfiObi2gP9pE”

Example: Deposit: {"amount": 250,"currency": "USD"}



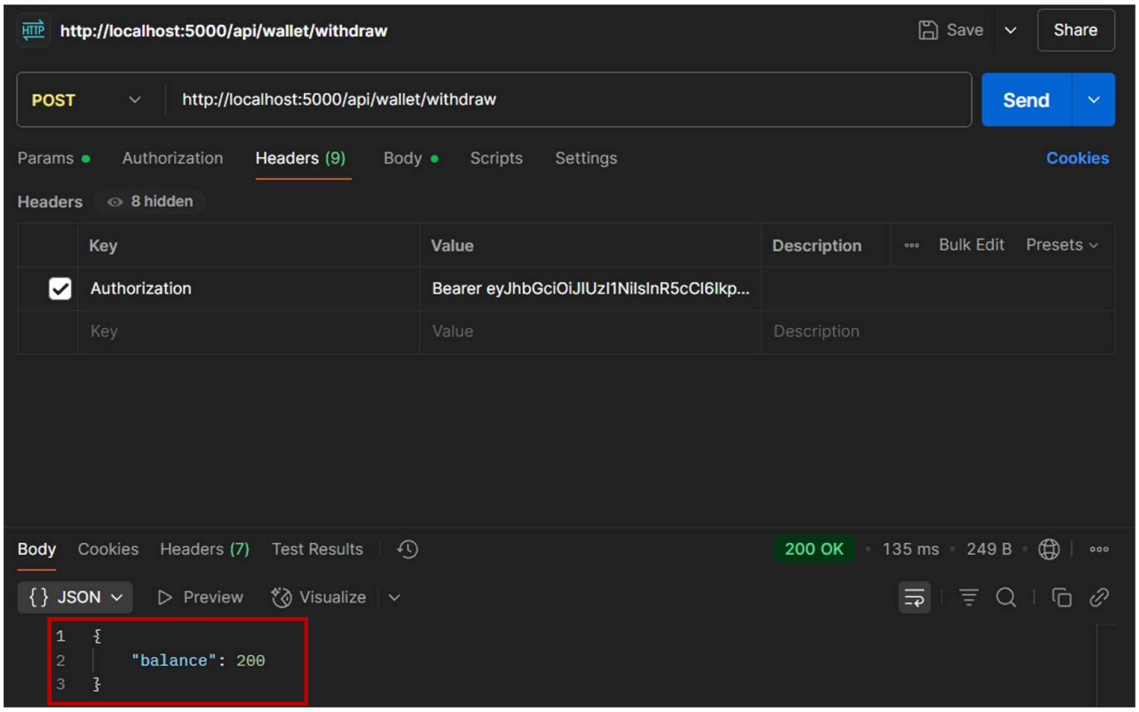
Response shows updated balance. (balance: 250)

B. **Withdraw Virtual Cash**

Endpoint: http://localhost:5000/api/wallet/withdraw

Headers:
Key: Authorization
Value: “Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjgyMWJmYzdlYzIwOGUyZTk1MzRkYSIsImlhdCI6MTc0NzQ2MDYyOSwiZXhwIjoxNzQ3NTQ3MDI5fQ.01hAiMmoND1KXn63hQkg1z8OfUGj7p7UfiObi2gP9pE”

Example: {"amount": 50,"currency": "USD"}



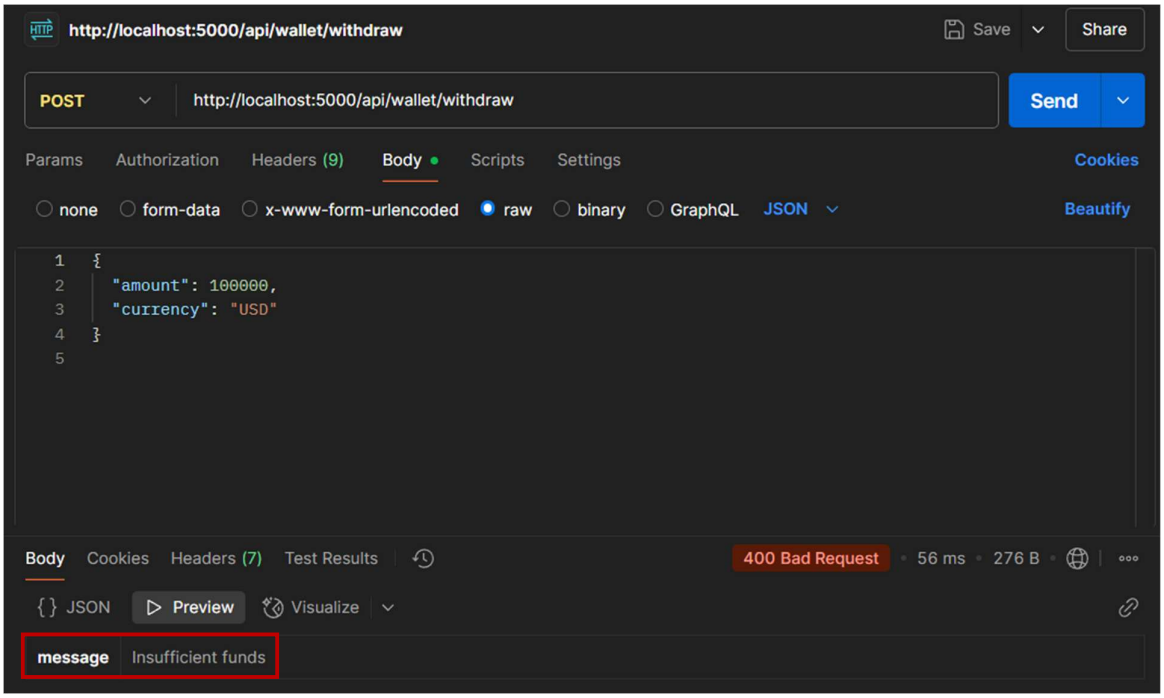
Withdrawn amount: 50
Balance: 200

C. **Withdraw More Than Balance**

Endpoint: http://localhost:5000/api/wallet/withdraw

Headers:
Key: Authorization
Value: “Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjgyMWJmYzdlYzIwOGUyZTk1MzRkYSIsImlhdCI6MTc0NzQ2MDYyOSwiZXhwIjoxNzQ3NTQ3MDI5fQ.01hAiMmoND1KXn63hQkg1z8OfUGj7p7UfiObi2gP9pE”

Example: {"amount": 10000,"currency": "USD"}



While trying to withdraw more amount than available in the account, it shows “Insufficient funds”.

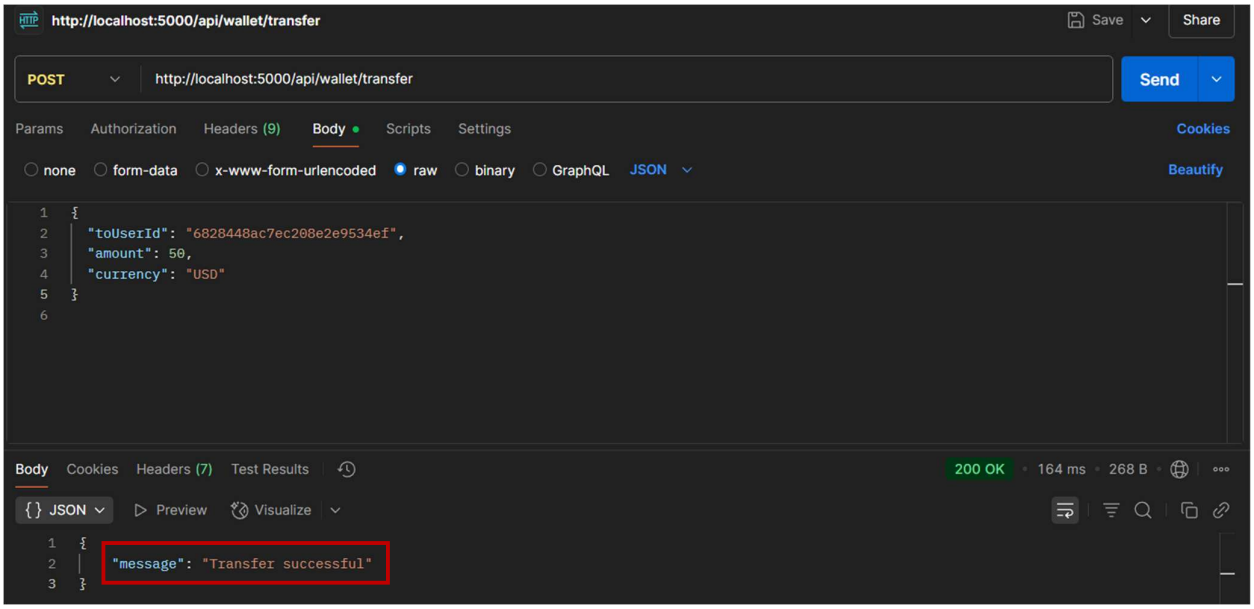
D. **Transfer Funds to Another User**

Endpoint: http://localhost:5000/api/wallet/transfer

Example : (Sender: David, Receiver : John)

David is transferring 50USD to John

{"toUserId": "6828448ac7ec208e2e9534ef","amount": 50,"currency": "USD"}



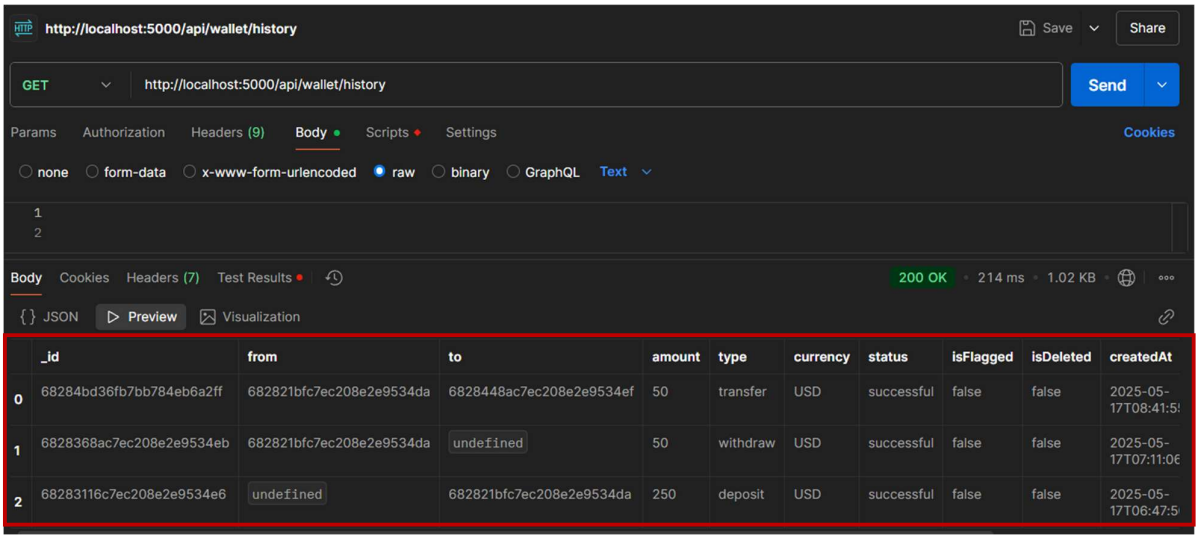
After successful transaction , it shows “Transfer successful”

David's balance decreases, John's increases.

E. Transaction history

Endpoint: http://localhost:5000/api/wallet/history

Headers:
Key: Authorization
Value: “Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjgyMWJmYzdlYzIwOGUyZTk1MzRkYSIsImlhdCI6MTc0NzQ2MDYyOSwiZXhwIjoxNzQ3NTQ3MDI5fQ.01hAiMmoND1KXn63hQkg1z8OfUGj7p7UfiObi2gP9pE”



	_id	from	to	amount	type	currency	status	isFlagged	isDeleted	createdAt
0	68284bd36fb7bb784eb6a2ff	682821bfc7ec208e2e9534da	6828448ac7ec208e2e9534ef	50	transfer	USD	successful	false	false	2025-05-17T08:41:5
1	6828368ac7ec208e2e9534eb	682821bfc7ec208e2e9534da	undefined	50	withdraw	USD	successful	false	false	2025-05-17T07:11:06
2	68283116c7ec208e2e9534e6	undefined	682821bfc7ec208e2e9534da	250	deposit	USD	successful	false	false	2025-05-17T06:47:5

Returns array of transactions.

3. **Transaction Processing and Validation**

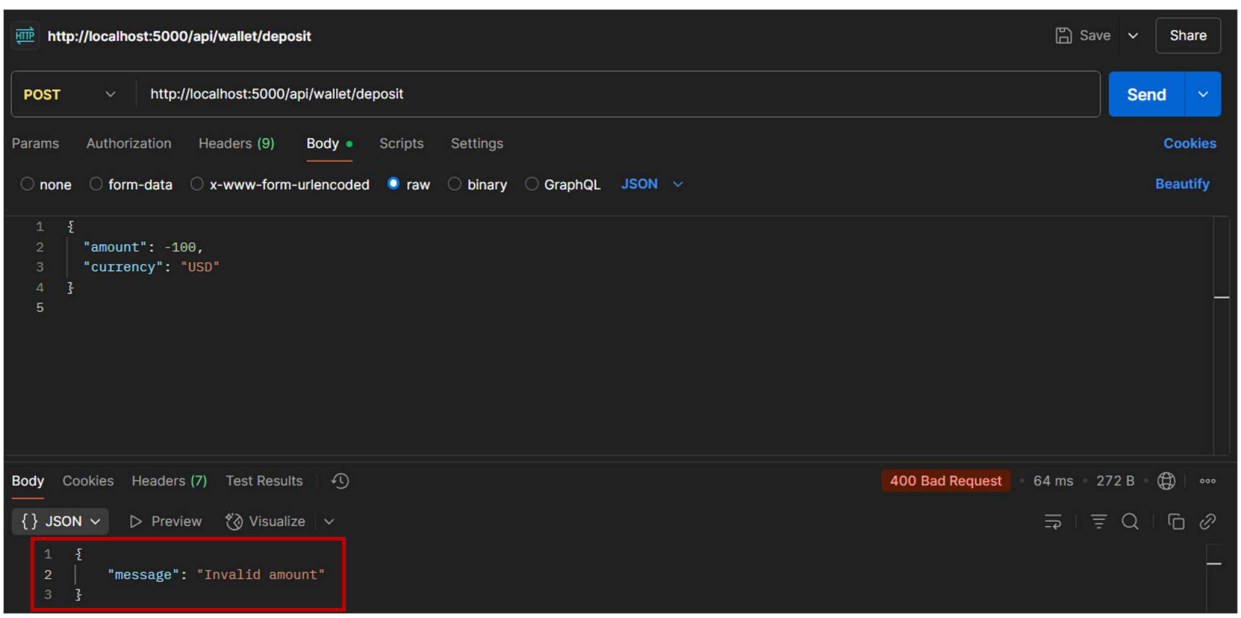
A. **Negative Deposit**

Endpoint: http://localhost:5000/api/wallet/deposit

Headers:
Key: Authorization
Value: “Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjgyMWJmYzdlYzIwOGUyZTk1MzRkYSIsImlhdCI6MTc0NzQ2MDYyOSwiZXhwIjoxNzQ3NTQ3MDI5fQ.01hAiMmoND1KXn63hQkg1z8OfUGj7p7UfiObi2gP9pE”

Example: We try to deposit which has a negative amount value.

{"amount": -100,"currency": "USD"}



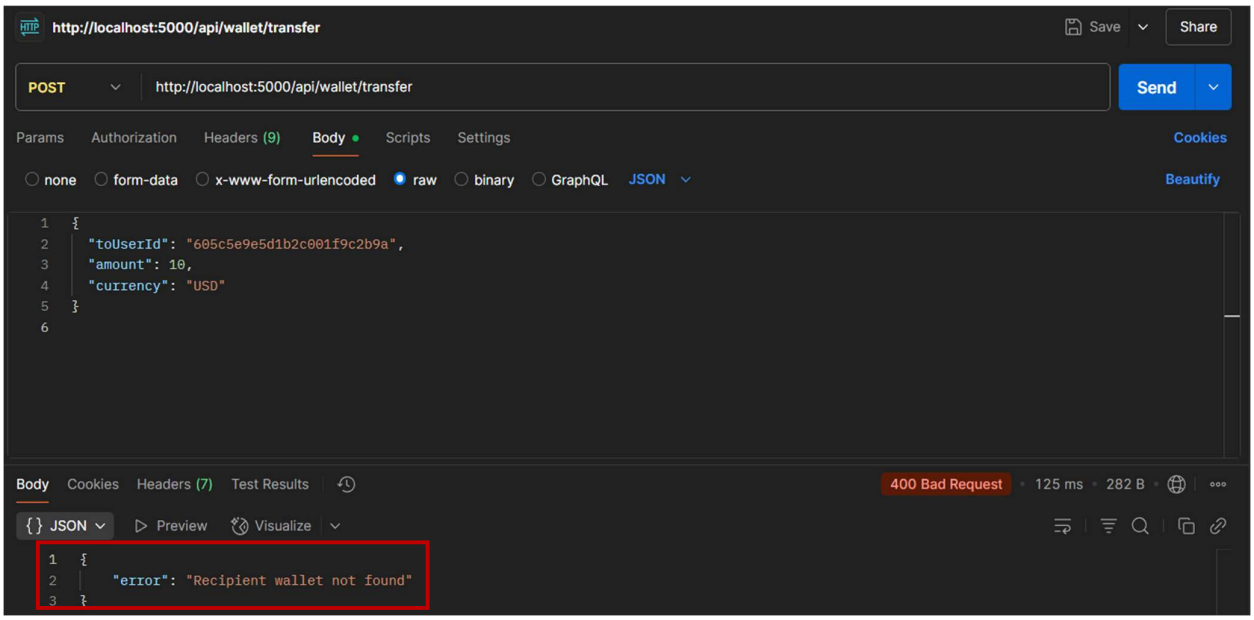
Returns “Invalid amount” as the deposit value cannot be negative.

B. **Invalid Transfer (Non-existent User)**

Endpoint: http://localhost:5000/api/wallet/transfer

Headers:
Key: Authorization
Value: “Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjgyMWJmYzdlYzIwOGUyZTk1MzRkYSIsImlhdCI6MTc0NzQ2MDYyOSwiZXhwIjoxNzQ3NTQ3MDI5fQ.01hAiMmoND1KXn63hQkg1z8OfUGj7p7UfiObi2gP9pE”

Example: Trying to send money to a non-existing user

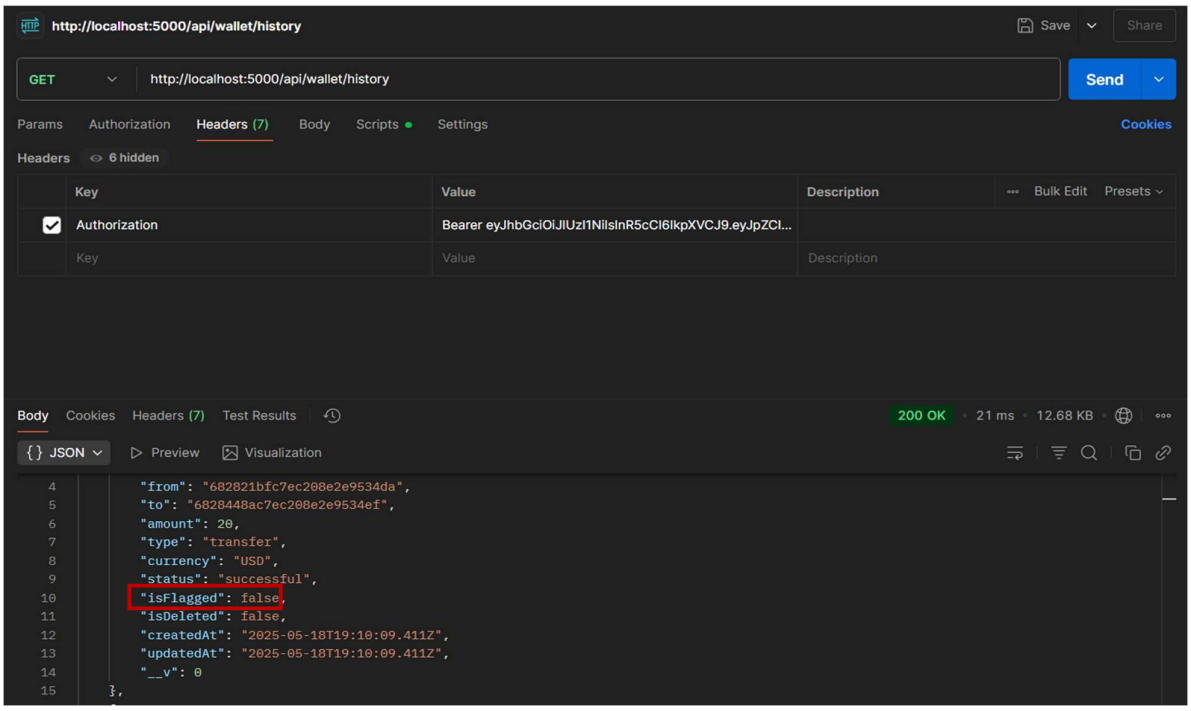


Returns “Recipient wallet not found” as no user is available with the UserID.

4. **Basic Fraud Detection Logic**

A. **Multiple Transfers in Short Period**

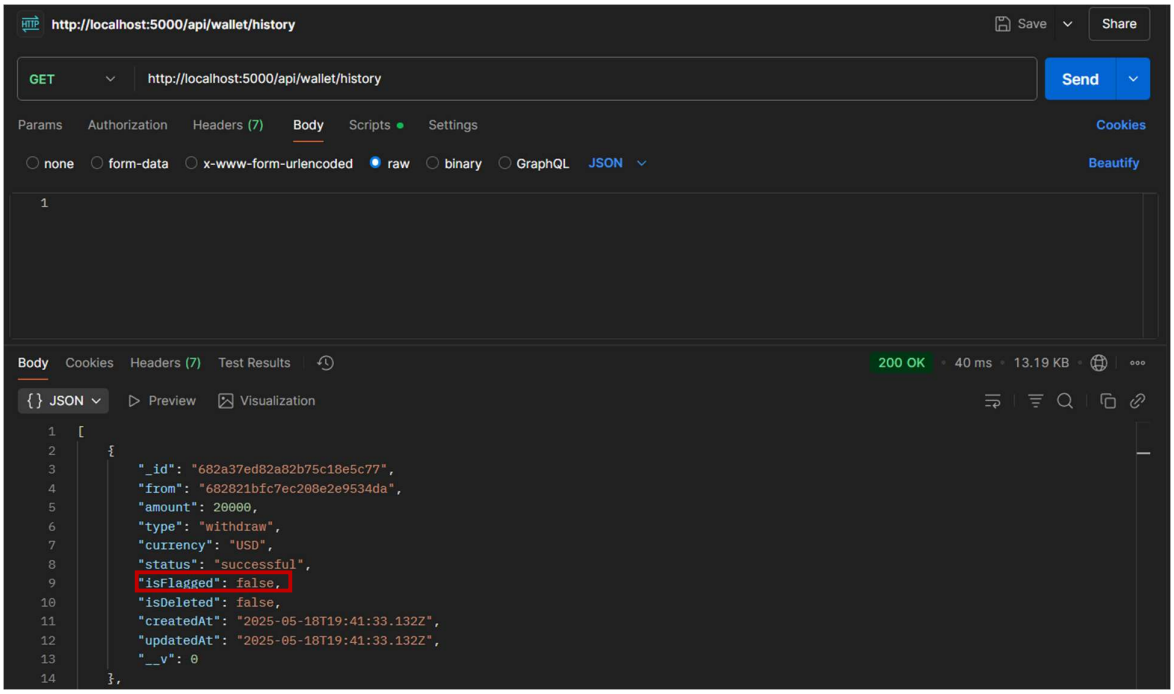
Example: Sending more than 3 quick transfers from David to John.



At least one transaction is flagged.

B. **Sudden Large Withdrawal**

Example: An amount of 20000 USD is being transferred.



Flag is being updated- suspicious activity.

Fraud flag:
User $682a37ed82a82b75c18e5c77 attempted large withdrawal: $20000

5. **Admin and Reporting APIs**

A. **View Flagged transactions**

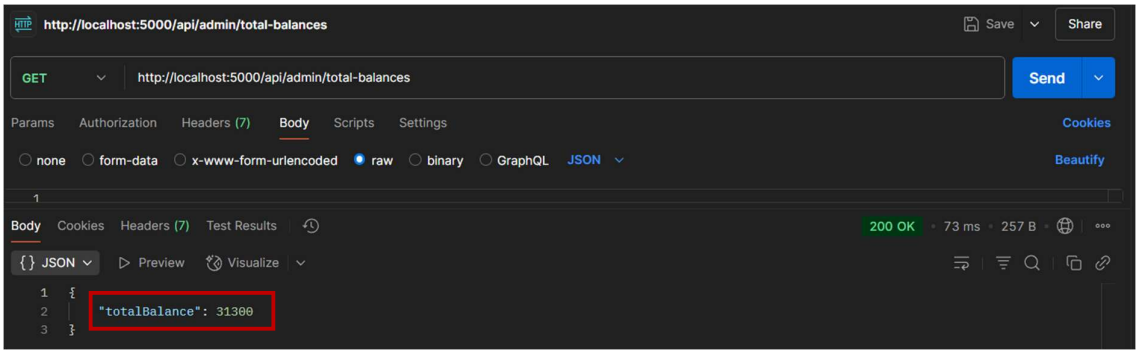
Endpoint: GET http://localhost:5000/api/admin/flagged

Returns all flagged transactions.

B. **Aggregating total user balances**

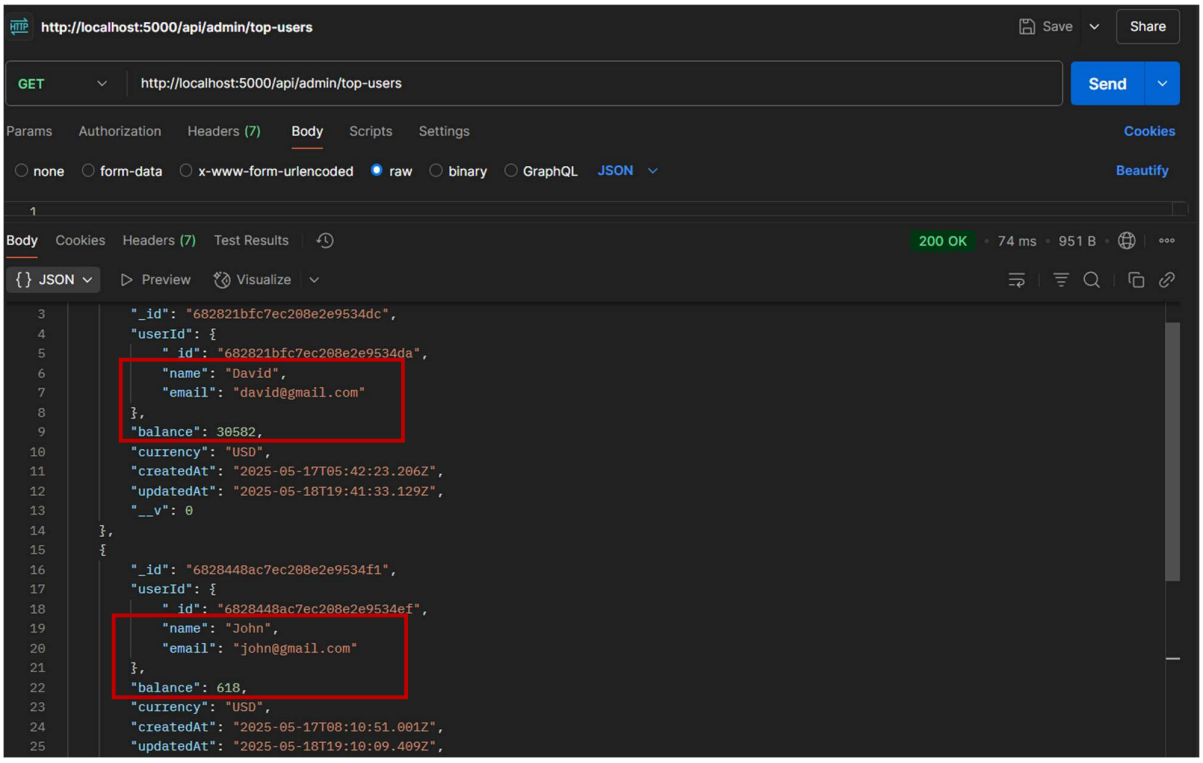
Endpoint: GET http://localhost:5000/api/admin/total-balances

Returns the total available balance



C. **Top Users by Balance**

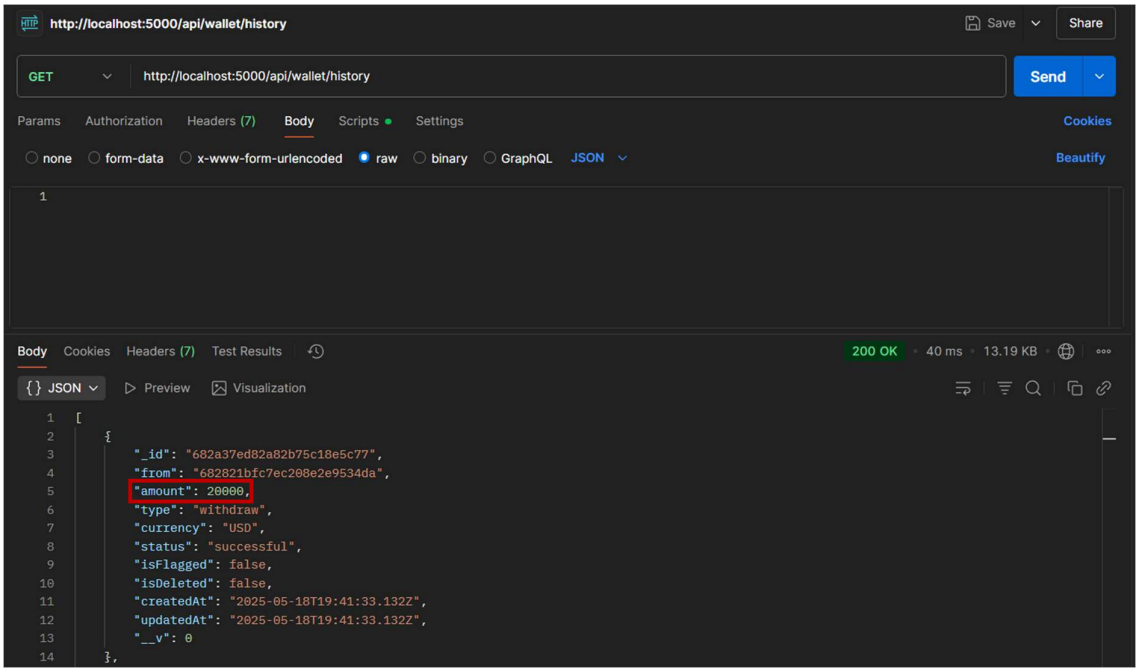
Endpoint: GET http://localhost:5000/api/admin/top-users



Returns array of users sorted by balance.

Bonus Features

A. Email alerts for large or suspicious transactions



In the fraud detection login, withdrawal above 10000 USD is considered as suspicious activity.

‘

[MOCK EMAIL] From: admin@gmail.com | To: david@gmail.com
Subject: Fraud Alert
Text: Transaction 682a37ed82a82b75c18e5c77 flagged as suspicious.
‘

B. Soft delete for accounts and transactions

Mark a Transaction as Deleted in MongoDB.

```
> db.transactions.updateOne(
    { _id: ObjectId("682a37ed82a82b75c18e5c77") },
    { $set: { isDeleted: true } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```