Unique Identification Authority of India
Government of India

AADHAAR

UIDAI Hackathon

Team Name : SigmaMewSquared
Team Reference ID : iDcaDjNa8m

Team Member Details

Sl No	Name	E Mail ID
1.	S Ashwin	ashwins1211@gmail.com
2.	Akash Ambashankar	akashambashankar@gmail.com
3.	Ganesh Chandrasekar	cb.ganesh666@gmail.com
4.	Monica Benjamin	monicabenjamin473@gmail.com
5.	Praveen Kumar	sign24masssth@gmail.com

About the Problem Statement / Solution

- **Theme :** Authentication Reimagined
- **Problem Statement :**
  - PS-1 : Airport/Stadium/Railway check-in Application
  - PS-2 : Aadhaar backed Video KYC for availing resident facing services
  - PS-3 : Use of Aadhaar as additional factor in 3rd party transaction
  - PS-4 : Achieving 100% auth success in Rural Areas
- **Solution :**

  To provide a more secure and robust architecture to authenticate legitimate aadhaar users.

Abstract of Initial Idea:

- Multimodal authentication (Resident & Verifier)
  - Face ID
  - Fingerprint
  - Session Token (Blockchain)
  - QR Code Generation and Scanning
  - Single Sign on
  - Live assistant for Customer service

- Reason :
  - Concept Generation from Literature Survey

Workflow - 1

- An application for authentication that uses face recognition and finger prints to authenticate using the registered mobile.
- It will be a multi-factor authentication and upon login you'll be redirected to a QR code scanner.
- The vendors associated with aadhaar authentication will have a QR code which can be scanned by our app.
- We classify verifiers into two types:
  - Type 1: The ones that need all your bio data.
  - Type 2: The ones that need minimal amount of data.

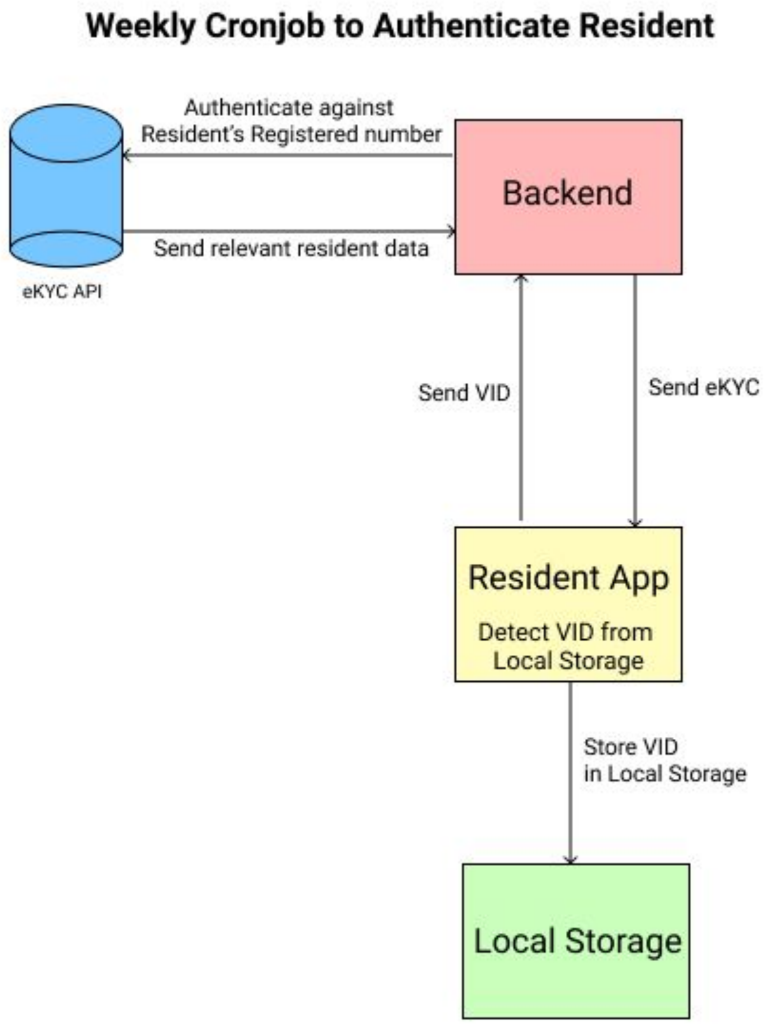
Workflow - 2

- Upon scanning the vendor will get a token of confirmation (based on authentication standards) or the required information based on the type of verifier.
- We prevent data leaks by only sending a token of confirmation instead of the user data (for type 2).
- This could be used as a Single Sign On (SSO) over multiple application platforms.

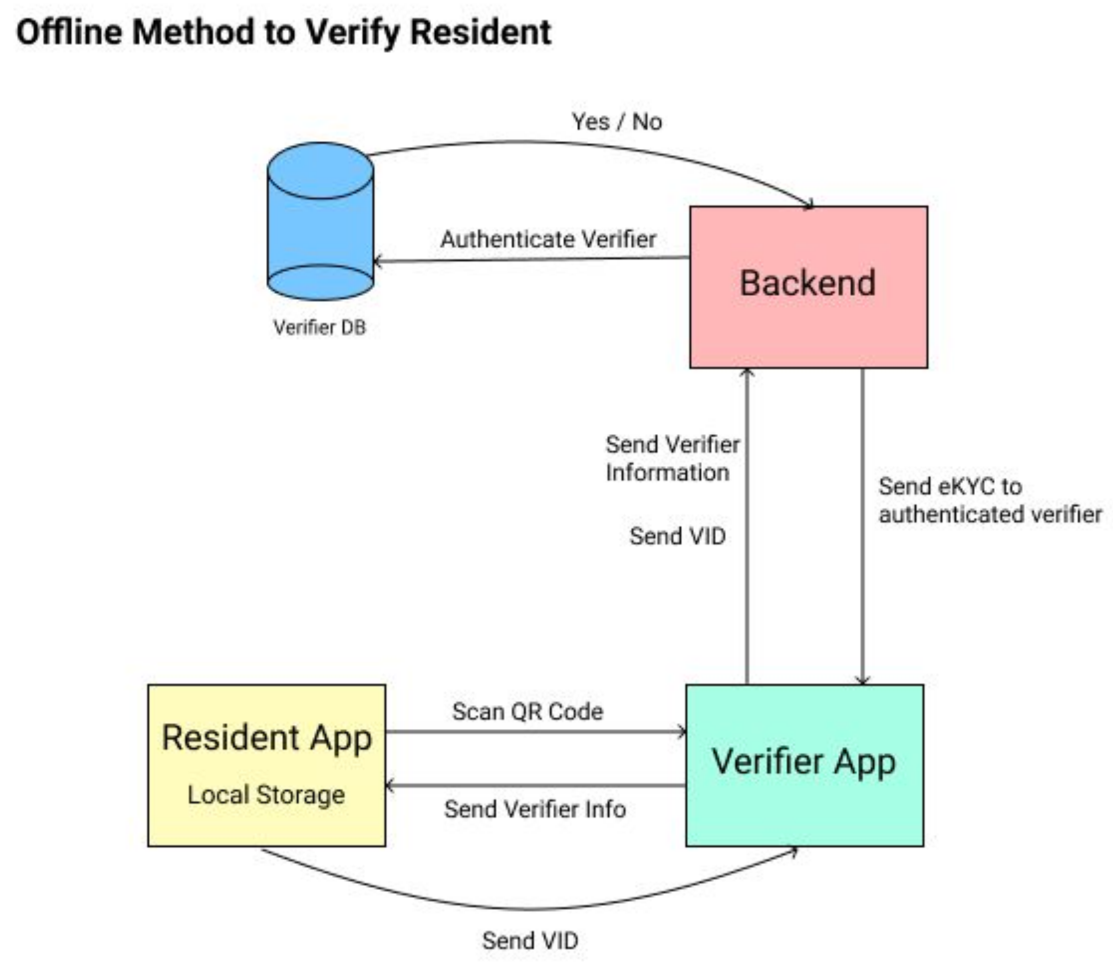
Workflow - 3

- We implement a live assistant hosted over the web to blacklist the registered device from getting the data after passing through multiple layers of security like security questions.
- In case the mobile is stolen with the intention of breaking into the system, and the unauthorized person has a way to break into the system, the live assistant may be used to block access to the device.

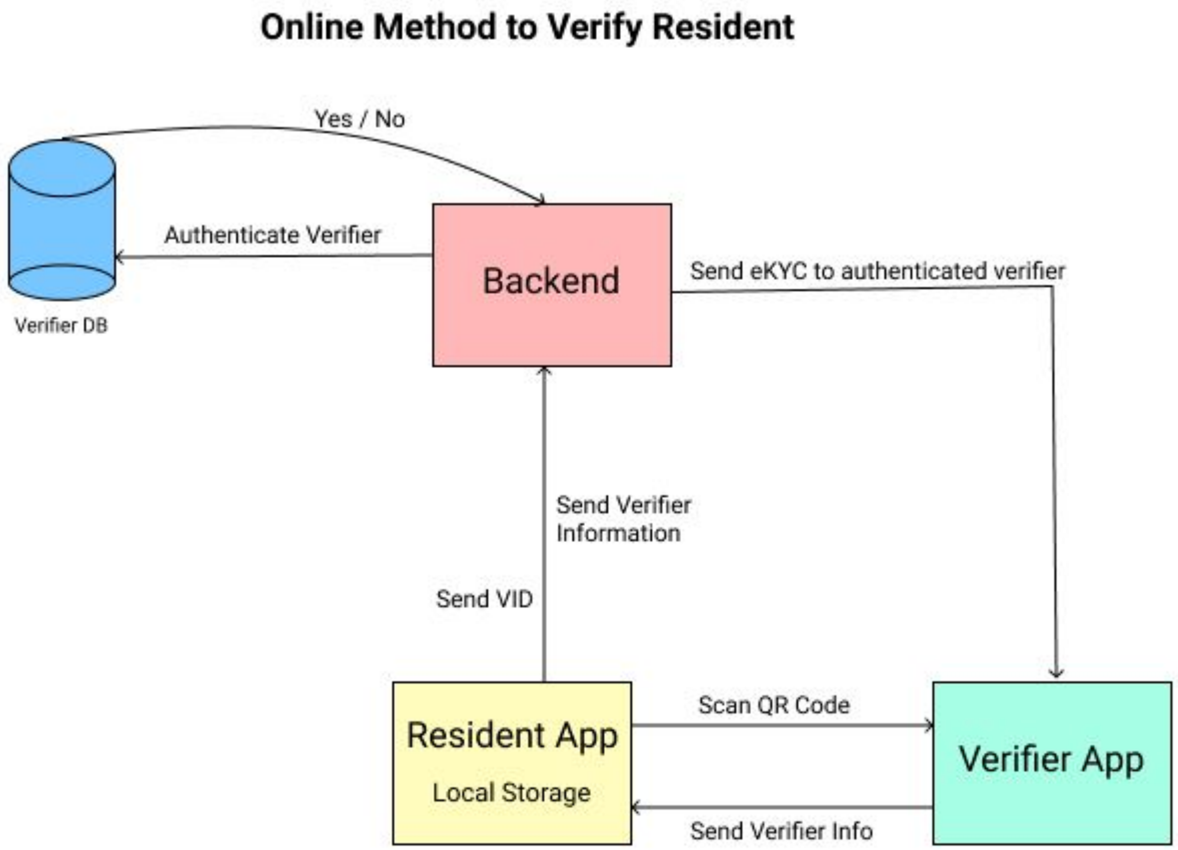
Architectural Diagram



Architectural Diagram



Architectural Diagram



Implementation Process

Frontend - Development

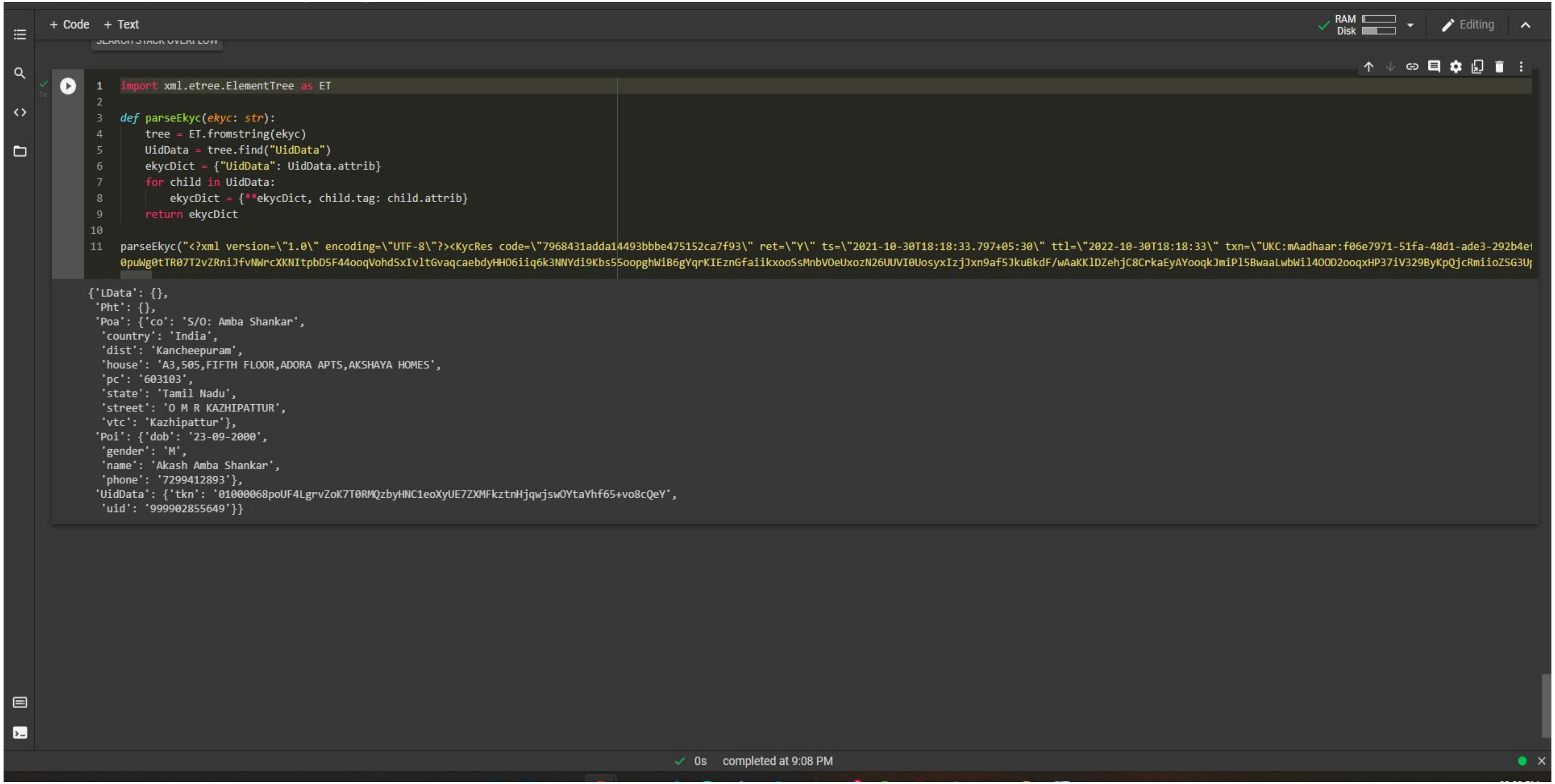
- Steps :

(Code Available in Github Repo: https://github.com/Akashamba/SigmaMewSquared-Submission)

  - Captcha Generation
  - OTP Generation
  - VID Generation

- Future Work:
  - Smoothening the workflow
  - Better authentication
  - Little to no screen touch required

eKYC Parsed Into Json



```python
import xml.etree.ElementTree as ET

def parseEkyc(ekyc: str):
    tree = ET.fromstring(ekyc)
    UidData = tree.find("UidData")
    ekycDict = {"UidData": UidData.attrib}
    for child in UidData:
        ekycDict = {**ekycDict, child.tag: child.attrib}
    return ekycDict

parseEkyc("<?xml version=\"1.0\" encoding=\"UTF-8\"?><KycRes code=\"7968431adda14493bbbe475152ca7f93\" ret=\"Y\" ts=\"2021-10-30T18:18:33.797+05:30\" ttl=\"2022-10-30T18:18:33\" txn=\"UKC:mAadhaar:f06e7971-51fa-48d1-ade3-292b4e
0puWg0tTR07T2vZRniJfvNWrcXKNItpbD5F44ooqVohdSxIvltGvaqcaebdyHHO6iiq6k3NNYdi9Kbs55oopghWiB6gYqrKIEznGfaiikxooSsMnbVOeUxozN26UUVI0UosyxIzjJxn9af5JkuBkdF/wAaKKlDZehjC8CrkaEyAYooqkJmiPl5BwaaLwbWil4OOD2ooqxHP37iV329ByKpQjcRmiioZSG3U
```

```
{'LData': {},
 'Pht': {},
 'Poa': {'co': 'S/O: Amba Shankar',
  'country': 'India',
  'dist': 'Kancheepuram',
  'house': 'A3,505,FIFTH FLOOR,ADORA APTS,AKSHAYA HOMES',
  'pc': '603103',
  'state': 'Tamil Nadu',
  'street': 'O M R KAZHIPATTUR',
  'vtc': 'Kazhipattur'},
 'Poi': {'dob': '23-09-2000',
  'gender': 'M',
  'name': 'Akash Amba Shankar',
  'phone': '7299412893'},
 'UidData': {'tkn': '01000068poUF4LgrvZoK7T0RMQzbyHNC1eoXyUE7ZXMFkztnHjqwjswOYtaYhf65+vo8cQeY',
  'uid': '999902855649'}}
```

Frontend - Partial Working

Backend - Development

Programming Language : Python

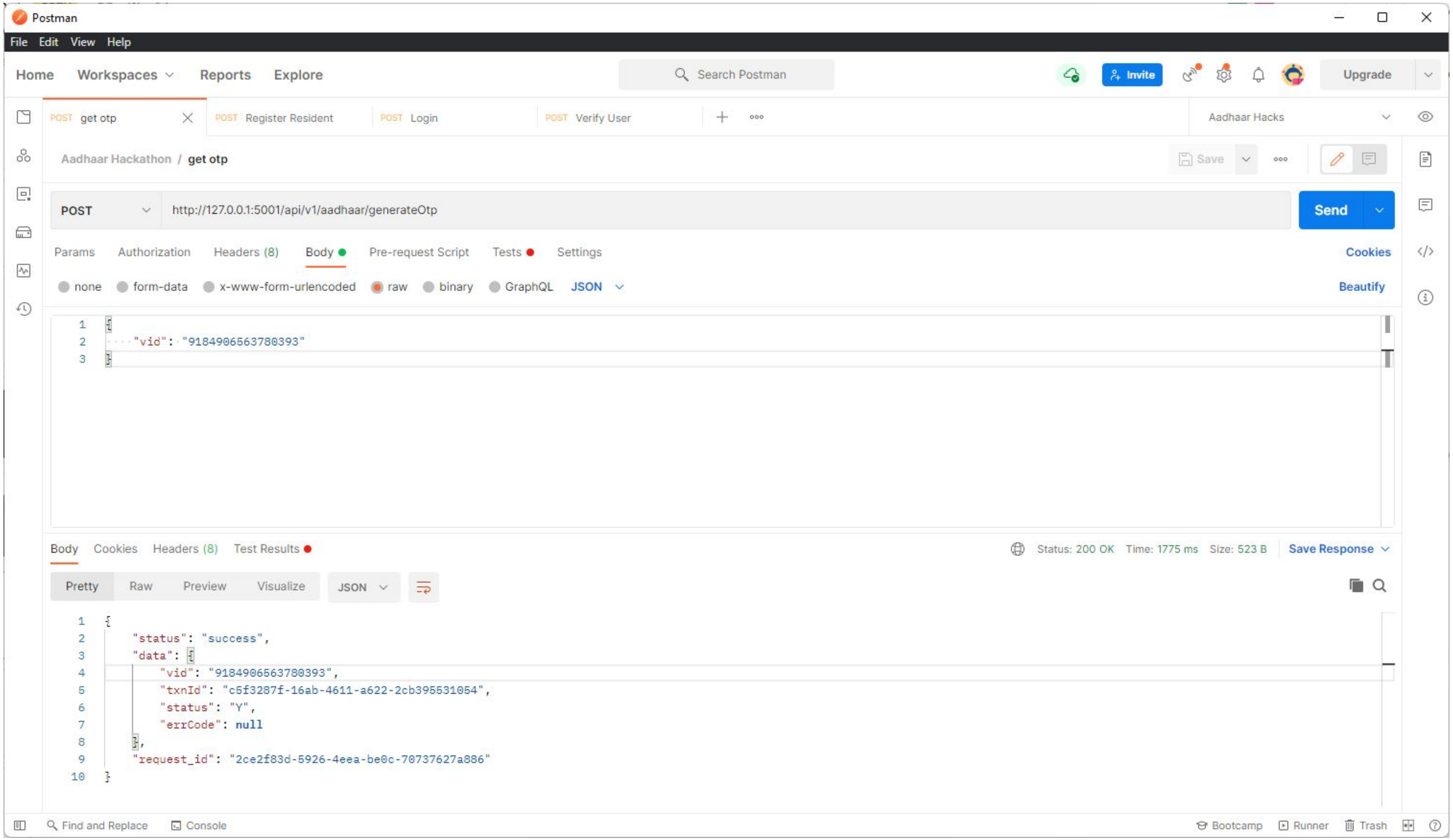
Framework : Flask

Database: PostgreSQL

Endpoints :

- /generateOtp
- /register
- /verifyUser
- /getResidentData
- /offlineDecode

Backend - Coding

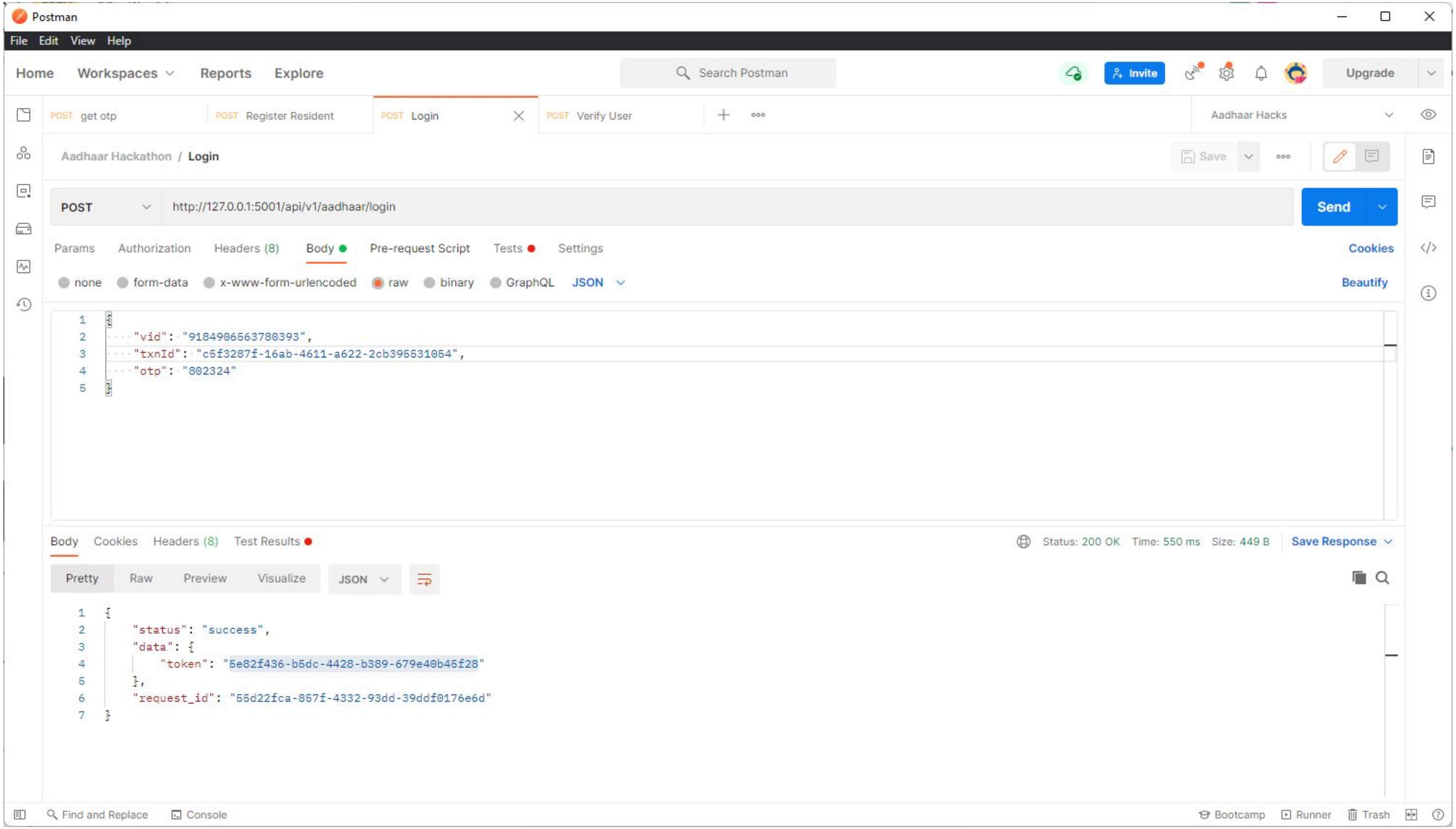

Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
POST get otp
POST Register Resident
POST Login
POST Verify User
Aadhaar Hacks
Aadhaar Hackathon / get otp
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
Beautify
Body Cookies Headers (8) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace
Console
Bootcamp Runner Trash

POST http://127.0.0.1:5001/api/v1/aadhaar/generateOtp

```
{
    "vid": "9184906563780393"
}
```

Status: 200 OK Time: 1775 ms Size: 523 B

```
{
    "status": "success",
    "data": {
        "vid": "9184906563780393",
        "txnId": "c5f3287f-16ab-4611-a622-2cb395531054",
        "status": "Y",
        "errCode": null
    },
    "request_id": "2ce2f83d-5926-4eea-be0c-70737627a886"
}
```



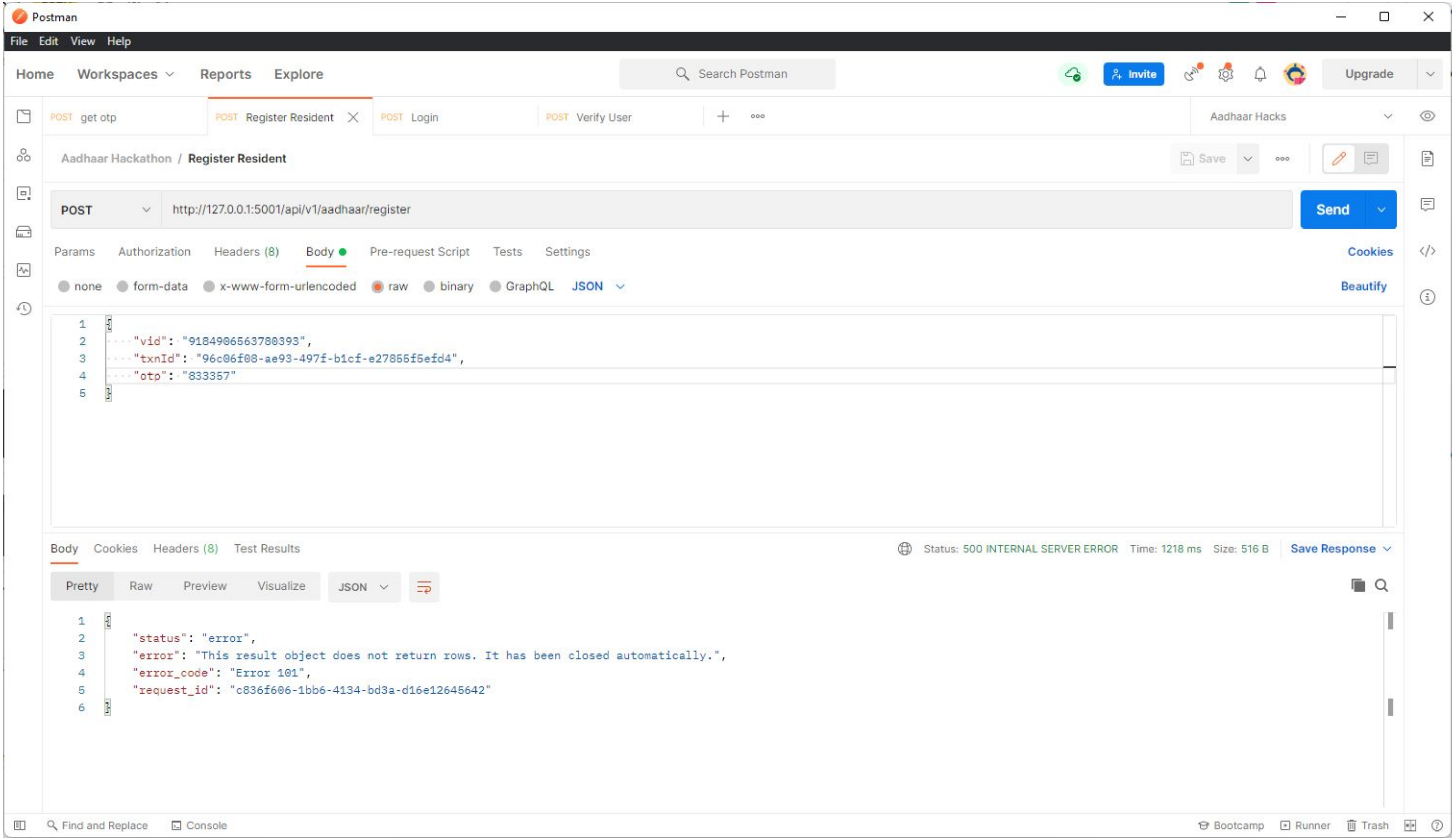
POST http://127.0.0.1:5001/api/v1/aadhaar/login

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "vid": "9184906563780393",
    "txnId": "c5f3287f-16ab-4611-a622-2cb395531054",
    "otp": "802324"
}
```

Status: 200 OK Time: 550 ms Size: 449 B

```
{
    "status": "success",
    "data": {
        "token": "5e82f436-b5dc-4428-b389-679e40b45f28"
    },
    "request_id": "55d22fca-857f-4332-93dd-39ddf0176e6d"
}
```



Aadhaar Hackathon / Register Resident

POST http://127.0.0.1:5001/api/v1/aadhaar/register

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "vid": "9184906563780393",
    "txnId": "96c06f08-ae93-497f-b1cf-e27855f5efd4",
    "otp": "833357"
}
```

Status: 500 INTERNAL SERVER ERROR Time: 1218 ms Size: 516 B

```
{
    "status": "error",
    "error": "This result object does not return rows. It has been closed automatically.",
    "error_code": "Error 101",
    "request_id": "c836f606-1bb6-4134-bd3a-d16e12645642"
}
```

API Usage

- Auth
- OTP
- Face
- Offline eKYC
- VID Wrapper
- Aadhaar eKYC
- UIDAuthVidServiceboundSms

Security Goals - 1

- Multi-modal or multi-layered biometric authentication which is known to be better than traditional password based authentication mechanisms.

- QR code scanning ensures the data of the verifier doesn’t leak to devices other than the intended resident.

- Data session management ensures that you can’t use the data on the application after a certain period of time (session becomes stale and stale data is not valid).

Security Goals - 2

- The verifier can not be spoofed because the data of the resident comes from a backend server which validates the verifier signature. So, unless the verifier is a valid one, the resident data is not exposed.
- Even if the resident app is cracked and someone tries to steal data, the sensitive data is not available at the resident application but at the backend server.

Thank You