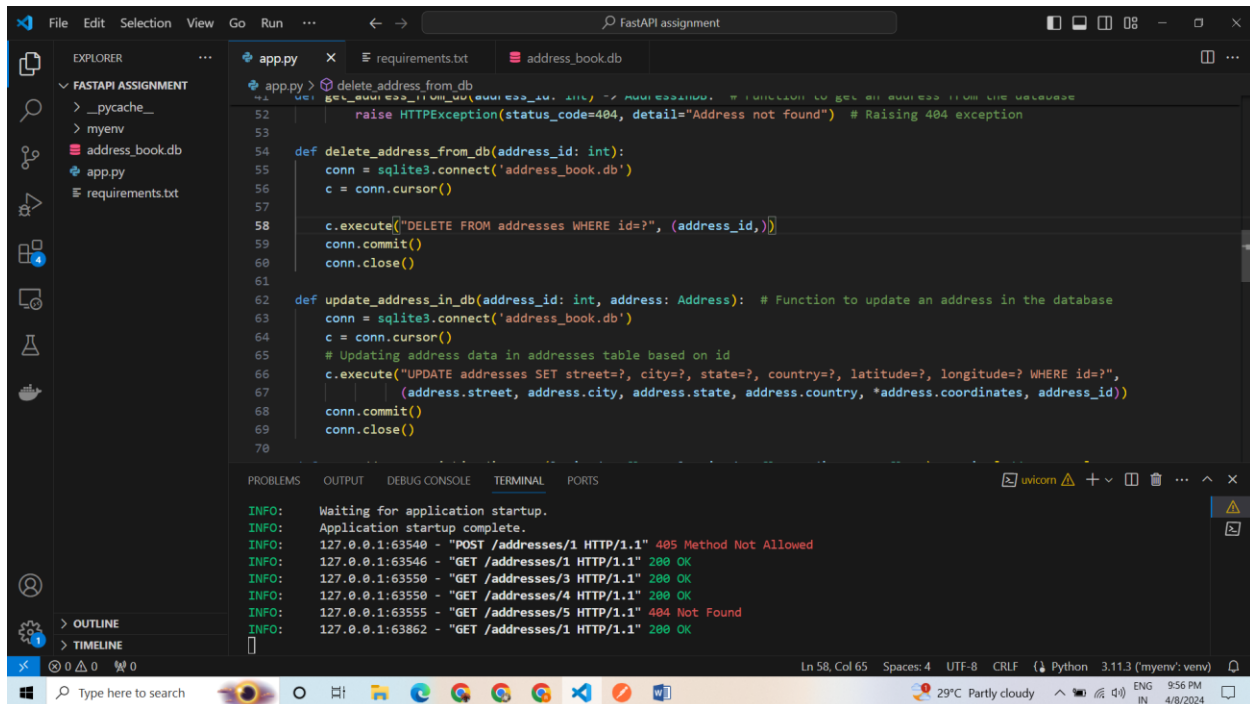


To run a FastAPI application

Directory structure



```
app.py
def delete_address_from_db(address_id: int):
    raise HTTPException(status_code=404, detail="Address not found") # Raising 404 exception

def delete_address_from_db(address_id: int):
    conn = sqlite3.connect('address_book.db')
    c = conn.cursor()

    c.execute("DELETE FROM addresses WHERE id=?", (address_id,))
    conn.commit()
    conn.close()

def update_address_in_db(address_id: int, address: Address): # Function to update an address in the database
    conn = sqlite3.connect('address_book.db')
    c = conn.cursor()
    # Updating address data in addresses table based on id
    c.execute("UPDATE addresses SET street=?, city=?, state=?, country=?, latitude=?, longitude=? WHERE id=?",
              (address.street, address.city, address.state, address.country, *address.coordinates, address_id))
    conn.commit()
    conn.close()
```

```
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:63540 - "POST /addresses/1 HTTP/1.1" 405 Method Not Allowed
INFO: 127.0.0.1:63546 - "GET /addresses/3 HTTP/1.1" 200 OK
INFO: 127.0.0.1:63550 - "GET /addresses/4 HTTP/1.1" 200 OK
INFO: 127.0.0.1:63550 - "GET /addresses/5 HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:63555 - "GET /addresses/1 HTTP/1.1" 200 OK
```

To running FastAPI we have use this command

Terminal: - `uvicorn app:app --reload`

Here are the API endpoints:

Create Address:

Endpoint: `POST /addresses/`

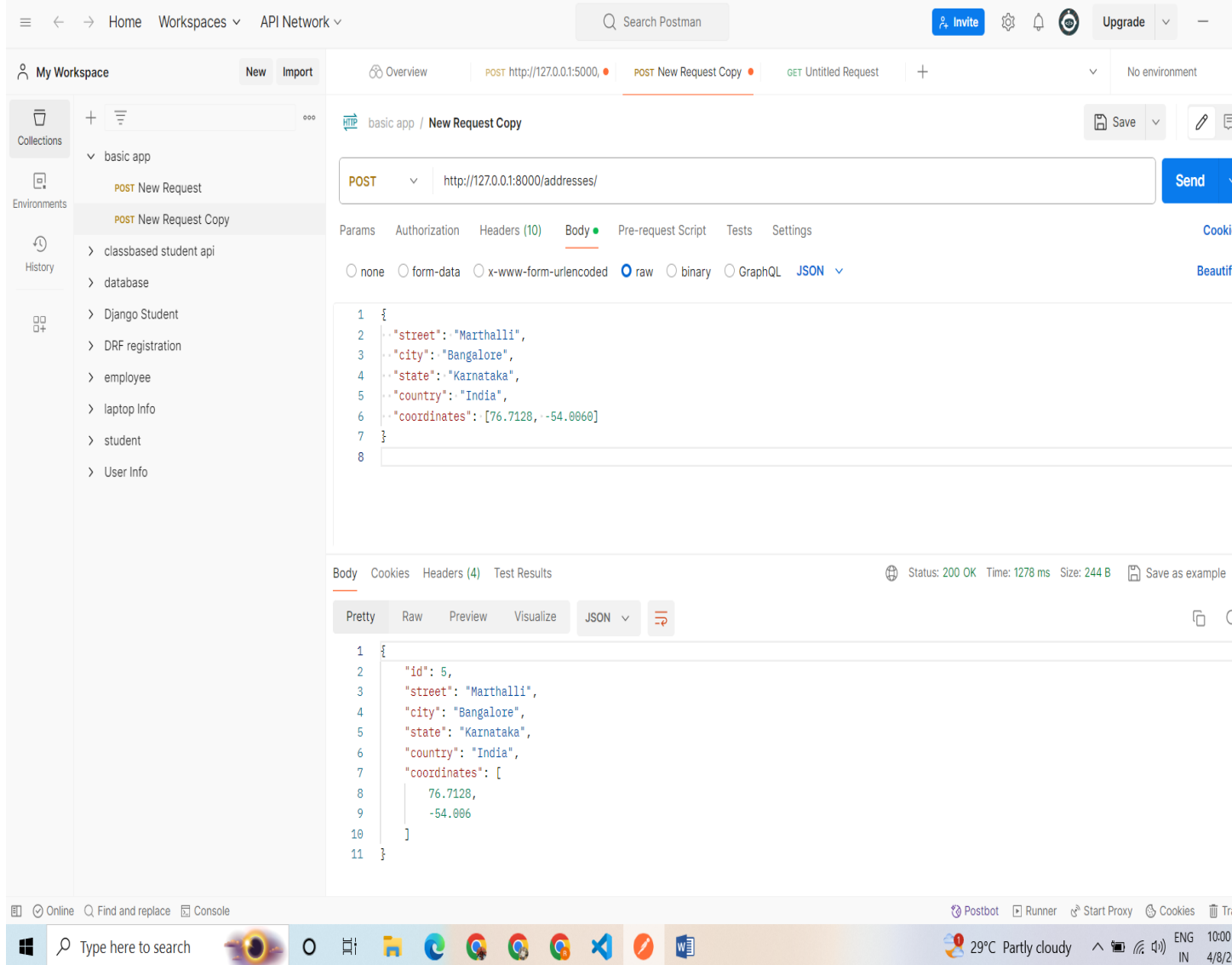
Actual api - `http://127.0.0.1:8000/addresses/`

Request Body: JSON containing address details (street, city, state, country, coordinates)

Json format data

```
{  
  "street": "Marthalli",  
  "city": "Bangalore",  
  "state": "Karnataka",  
  "country": "INDIA",  
  "coordinates": [76.7128, -54.0060]  
}
```

Response: JSON containing the created address details with assigned ID



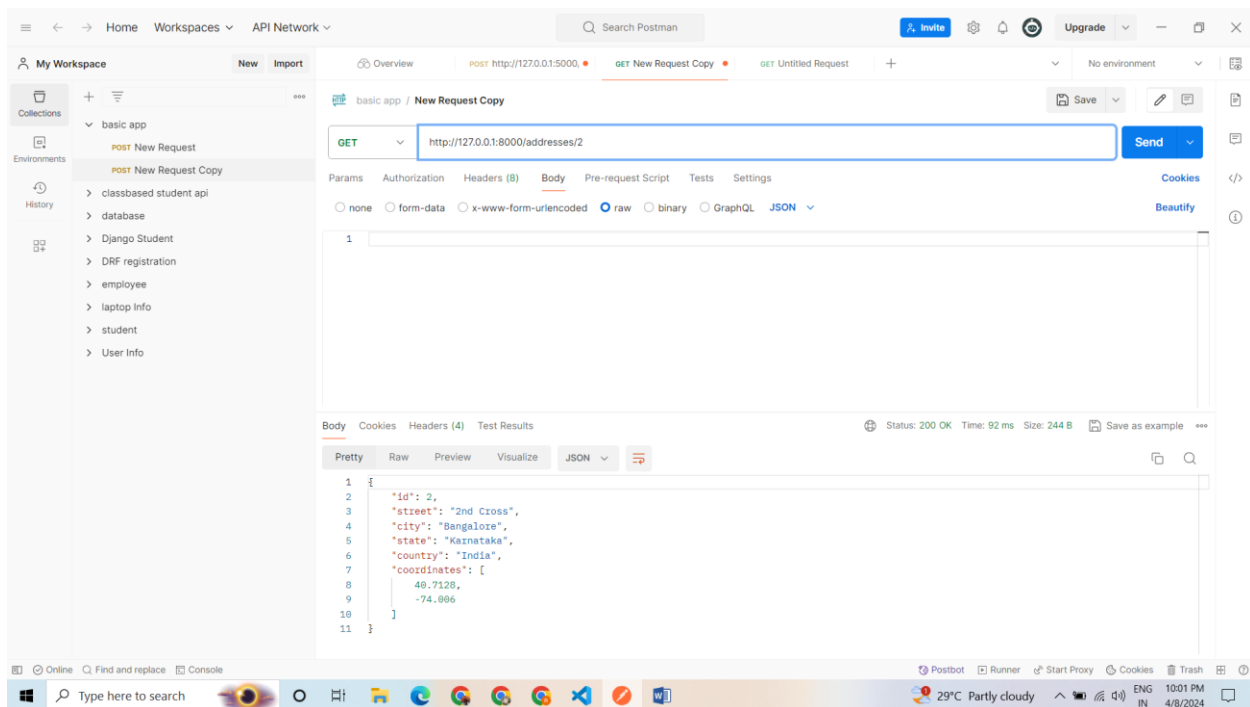
Read Address:

Endpoint: GET /addresses/{address_id}

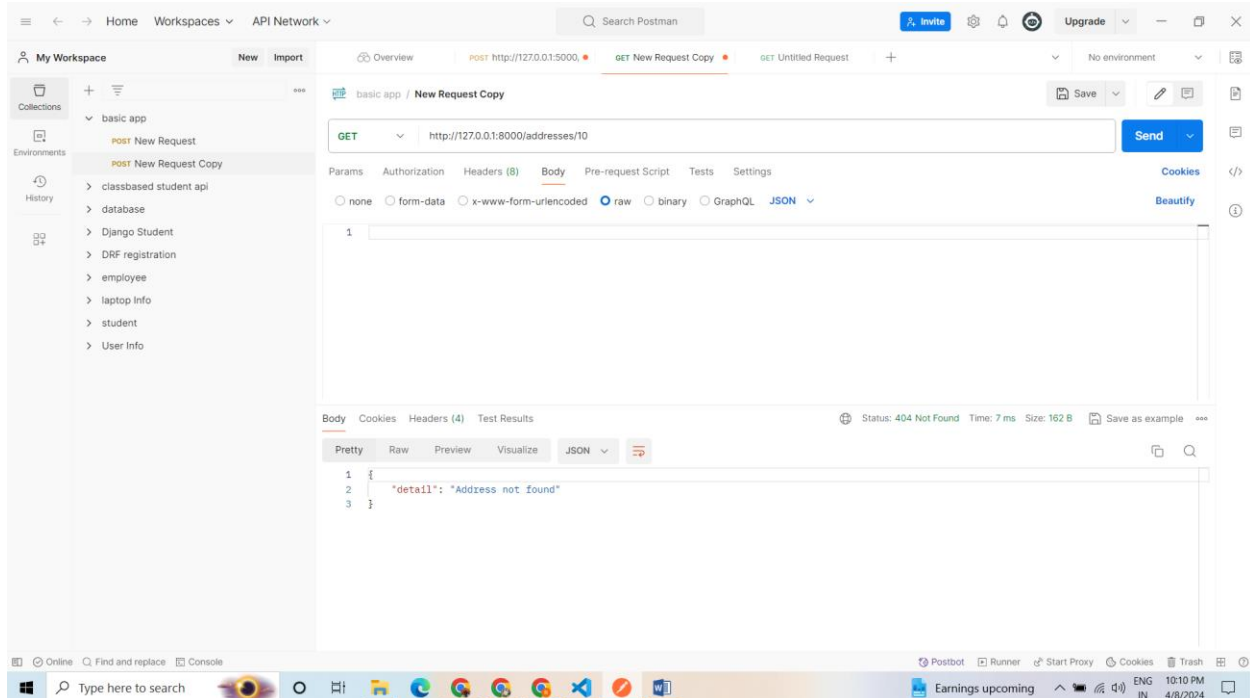
Actual api - http://127.0.0.1:8000/addresses/2

Path Parameter: address_id - ID of the address to retrieve

Response: JSON containing the address details



Data validation when address id is not existing



Update Address:

Endpoint: PUT /addresses/{address_id}

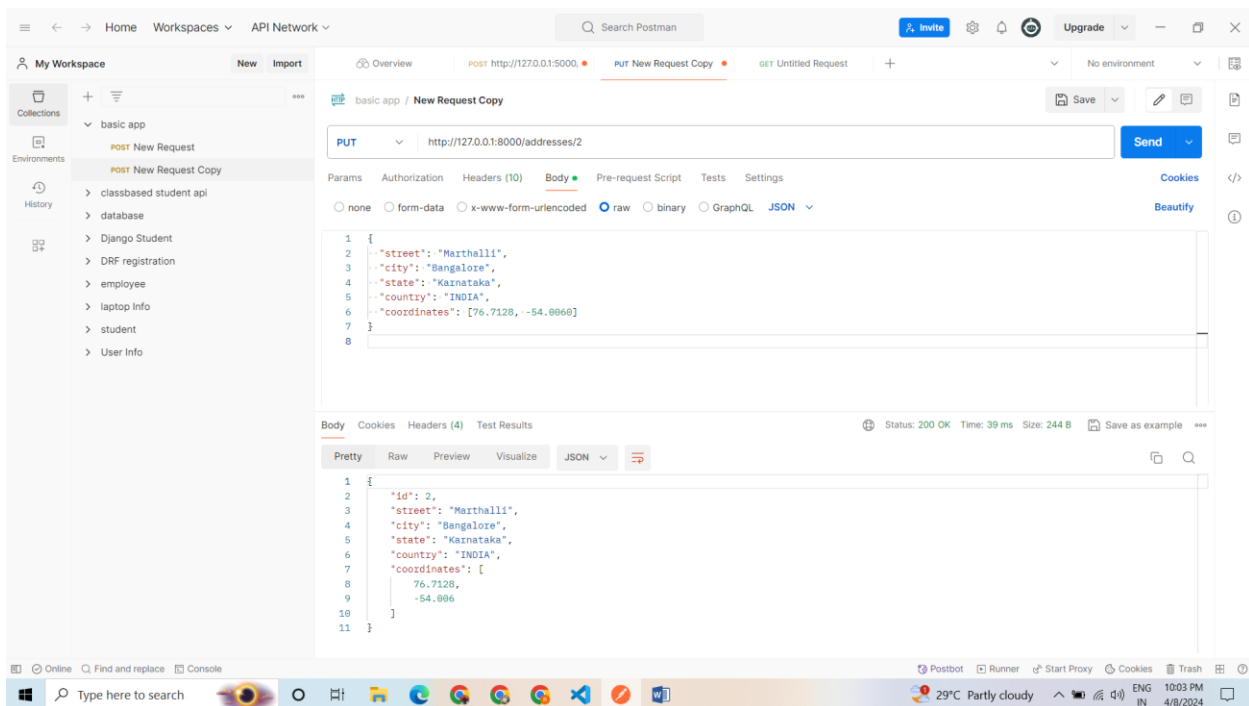
Actual api - http://127.0.0.1:8000/addresses/2

Path Parameter: address_id - ID of the address to update

Request Body: JSON containing updated address details (street, city, state, country, coordinates)

Here I have updated India to INDIA

Response: JSON containing the updated address details



Delete Address:

Endpoint: DELETE /addresses/{address_id}

<http://127.0.0.1:8000/addresses/1> (deleting record 1)

Path Parameter: address_id - ID of the address to delete

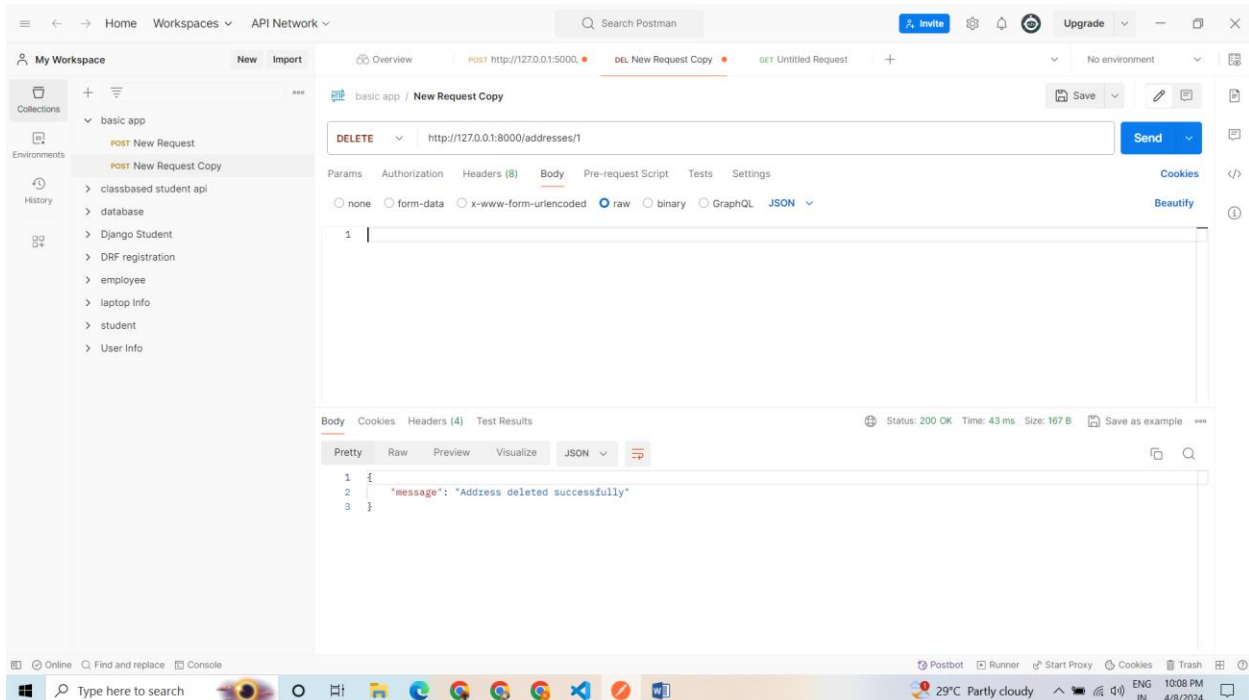
Response: JSON with a message confirming successful deletion

Get Addresses Within Radius:

Endpoint: GET /addresses/

Query Parameters: latitude, longitude, distance

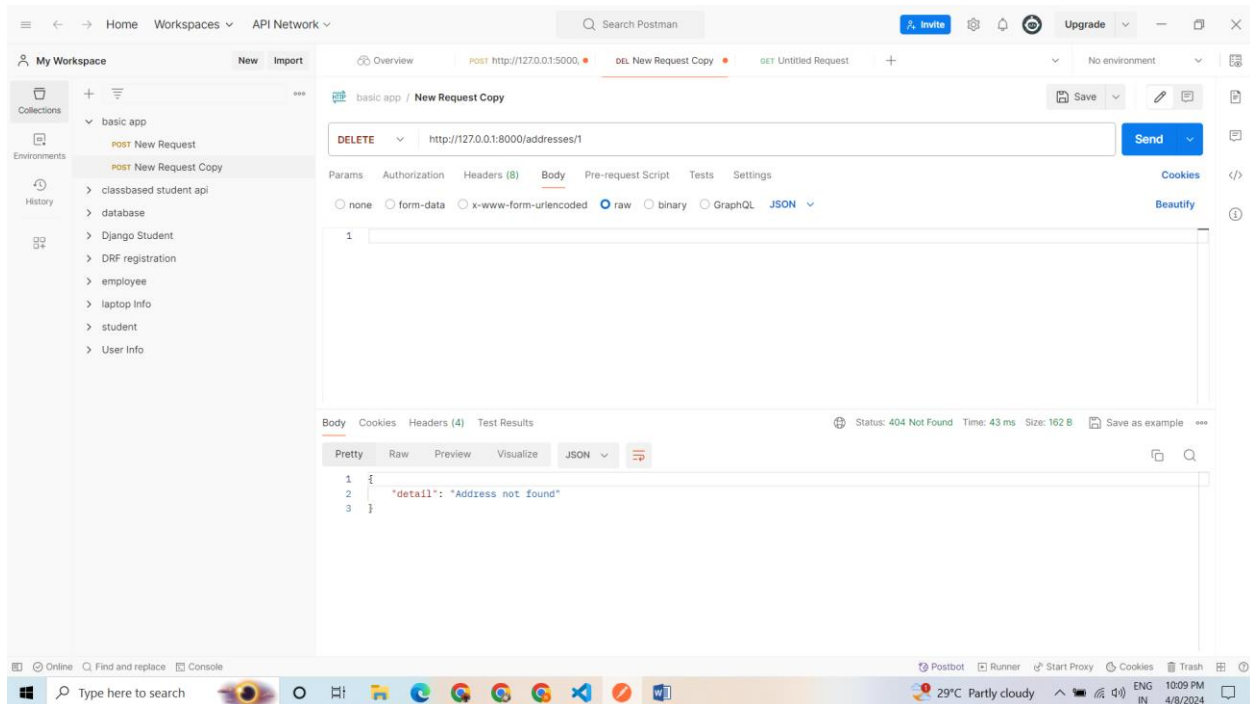
Response: JSON array containing addresses within the specified distance from the provided location coordinates



After deletion again hitting same api

<http://127.0.0.1:8000/addresses/1>

Data validation



5. Get Addresses Within Radius

- Send a GET request to `http://127.0.0.1:8000/addresses/` with query parameters latitude, longitude, and distance.
- For example, to get addresses within a radius of 10 kilometers from a location with latitude 40.7128 and longitude -74.0060, send a GET request to

`http://127.0.0.1:8000/addresses/?latitude=76.7128&longitude=-54.0060&distance=10`

basic app / New Request Copy

GET

http://127.0.0.1:8000/addresses/?latitude=76.7128&longitude=-54.0060&distance=10

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautiful

1

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 35 ms

Size: 365 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

1

[

2

{

3

"id": 2,

4

"street": "Marthalli",

5

"city": "Bangalore",

6

"state": "Karnataka",

7

"country": "INDIA",

8

"coordinates": [

9

76.7128,

10

-54.006

11

]

12

},