SOFTWARE ENGINEERING INTERN TASK

NAME: SOLOMON KIPKIRUI

DEGREE: BSc GIS AND REMOTE SENSING

TASK: HEALTH INFORMATION SYSTEM

BACKEND DEVELOPMENT

- **Core Technologies**
  - Django 5.1.6
  - PostGIS (PostgreSQL with GIS extensions)
  - Django REST Framework + GIS extensions
  - Leaflet for map rendering

- **Project Structure Overview**

Backend: Django supporting GeoDjango

Database: PostgreSQL with PostGIS for geographic data

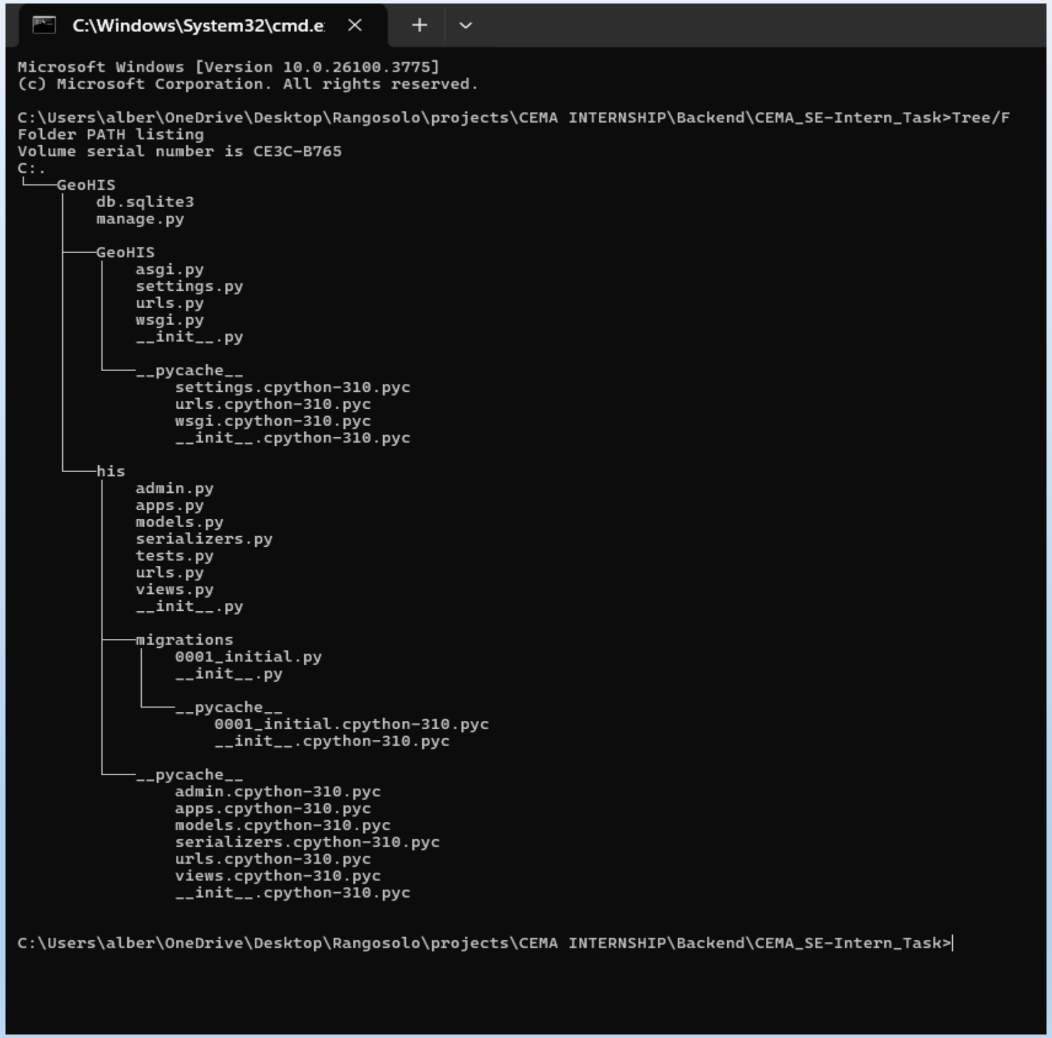
API: RESTful API using Django REST Framework and REST Framework GIS

Frontend: (Vue + Vite + Tailwindcss)

Key Apps: his

BACKEND DEVELOPMENT

File structure for the backend



```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\alber\OneDrive\Desktop\Rangosolo\projects\CEMA INTERNSHIP\Backend\CEMA_SE-Intern_Task>Tree/F
Folder PATH listing
Volume serial number is CE3C-B765
C:.
└───GeoHIS
    │   db.sqlite3
    │   manage.py
    │
    ├───GeoHIS
    │   │   asgi.py
    │   │   settings.py
    │   │   urls.py
    │   │   wsgi.py
    │   │   __init__.py
    │   │
    │   └───__pycache__
    │           settings.cpython-310.pyc
    │           urls.cpython-310.pyc
    │           wsgi.cpython-310.pyc
    │           __init__.cpython-310.pyc
    │
    └───his
        │   admin.py
        │   apps.py
        │   models.py
        │   serializers.py
        │   tests.py
        │   urls.py
        │   views.py
        │   __init__.py
        │
        ├───migrations
        │   │   0001_initial.py
        │   │   __init__.py
        │   │
        │   └───__pycache__
        │           0001_initial.cpython-310.pyc
        │           __init__.cpython-310.pyc
        │
        └───__pycache__
                admin.cpython-310.pyc
                apps.cpython-310.pyc
                models.cpython-310.pyc
                serializers.cpython-310.pyc
                urls.cpython-310.pyc
                views.cpython-310.pyc
                __init__.cpython-310.pyc


C:\Users\alber\OneDrive\Desktop\Rangosolo\projects\CEMA INTERNSHIP\Backend\CEMA_SE-Intern_Task>
```

Django Settings

- One of the most prominent settings is GDAL library path for GIS functions, debug mode on for debugging, and permitted hosts set to local testing. Installed applications are GIS, REST Framework, CORS headers, and Leaflet to process spatial data.

- **DATABASE CONFIGURATION**

The project uses PostgreSQL with
PostGIS to store and query spatial data. The
database is installed locally
with dev credentials to assist geographic
features the project needs.

- **REST FRAMEWORK API SECURITY**

API authentication is disabled for development convenience, and permissions grant open access. CORS is configured to allow requests from the local frontend. Authentication and more restrictive permissions should be turned on in productio

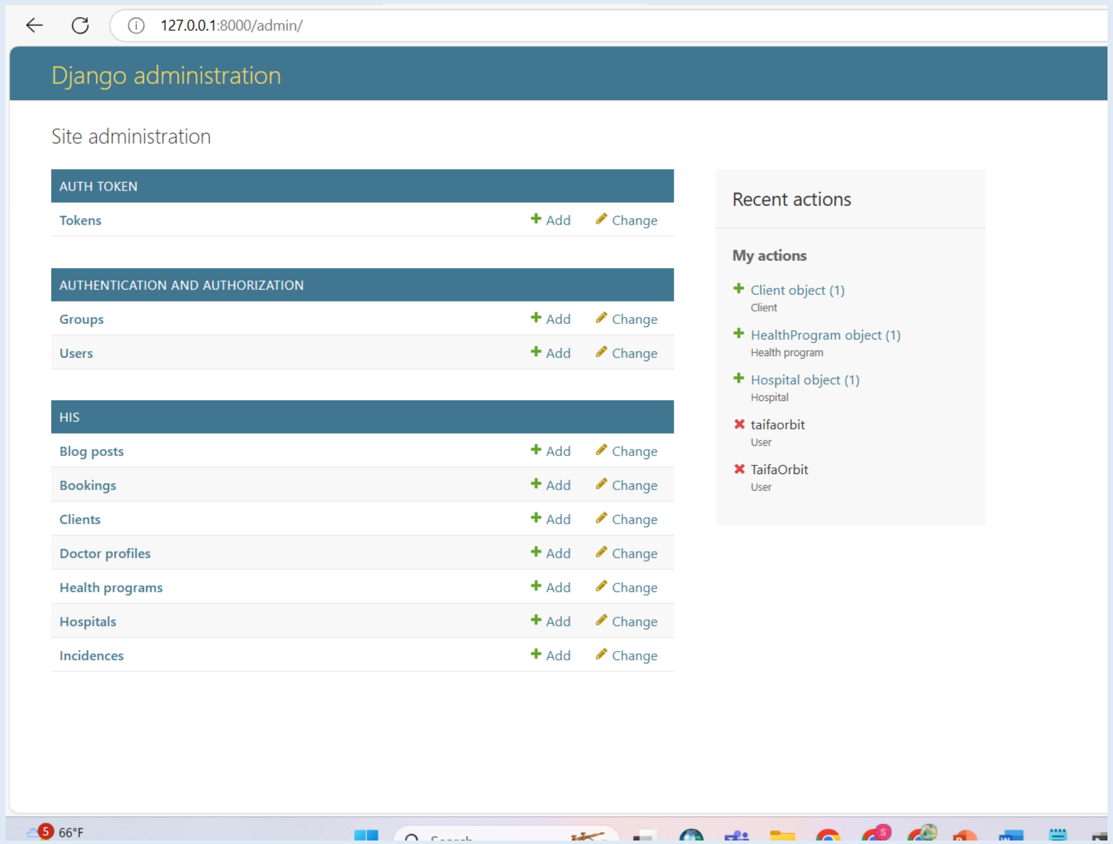
SERIALIZERS

- Serializers convert instances of models to JSON, adding geographic fields as GeoJSON. Hospital, Client, and Incidence, which handle spatial data, are some of the most crucial serializers, along with HealthProgram, DoctorProfile, Booking, and BlogPost.

API ROUTING AND VIEWSETS

- The API uses Django REST Framework's DefaultRouter to automatically create endpoints for hospitals, health programs, physicians, clients, bookings, blog posts, and incidences, providing RESTful access to all primary data models.

RESULTS: ADMIN PAGE


127.0.0.1:8000/admin/

Django administration

Site administration

AUTH TOKEN		
Tokens	Add	Change

AUTHENTICATION AND AUTHORIZATION		
Groups	Add	Change
Users	Add	Change

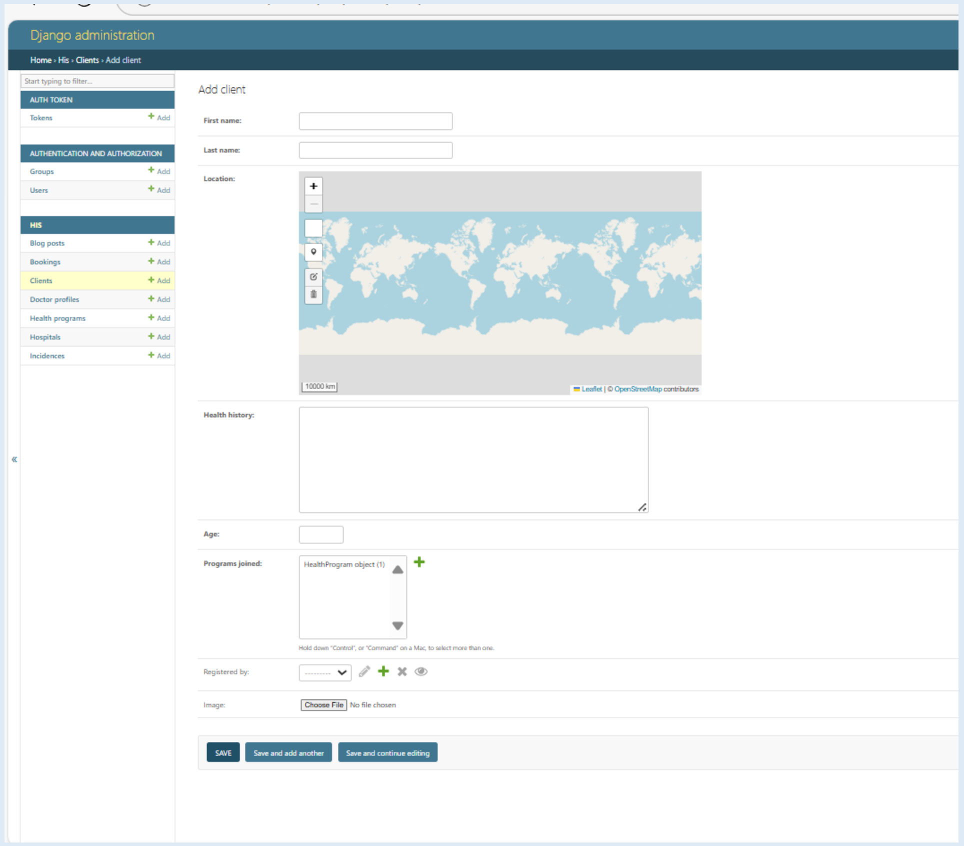
HIS		
Blog posts	Add	Change
Bookings	Add	Change
Clients	Add	Change
Doctor profiles	Add	Change
Health programs	Add	Change
Hospitals	Add	Change
Incidences	Add	Change

Recent actions

My actions

- Client object (1)
  Client
- HealthProgram object (1)
  Health program
- Hospital object (1)
  Hospital
- taifaorbit
  User
- TaifaOrbit
  User

66°F


Django administration

Home › His › Clients › Add client

Start typing to filter...

AUTH TOKEN	
Tokens	Add

AUTHENTICATION AND AUTHORIZATION	
Groups	Add
Users	Add

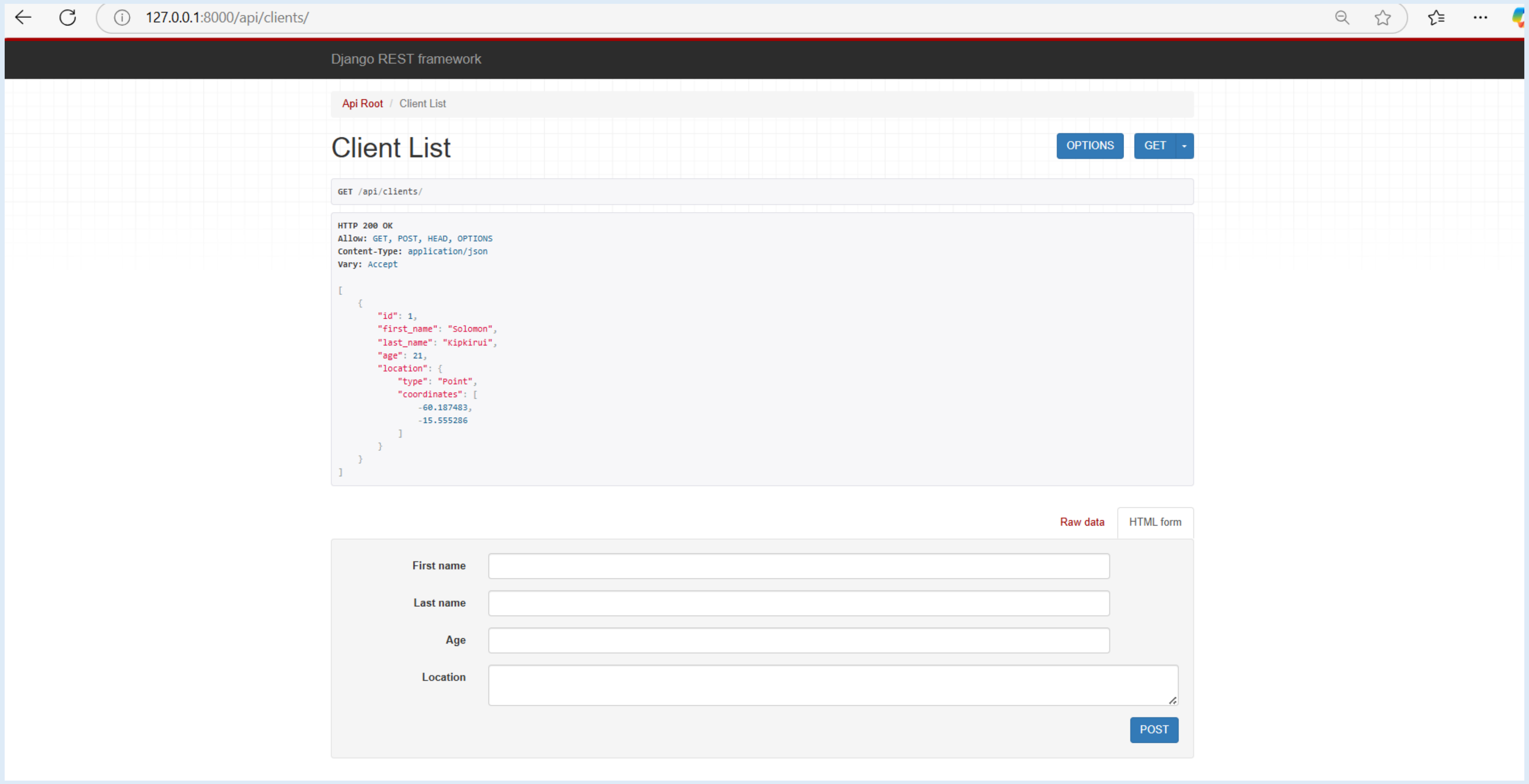
HIS	
Blog posts	Add
Bookings	Add
Clients	Add
Doctor profiles	Add
Health programs	Add
Hospitals	Add
Incidences	Add

Add client

First name:	
Last name:	
Location:	10000 km · Leaflet | © OpenStreetMap contributors
Health history:	
Age:	
Programs joined:	HealthProgram object (1)
	Hold down "Control", or "Command" on a Mac, to select more than one.
Registered by:	---------
Image:	Choose File No file chosen

SAVE · Save and add another · Save and continue editing

RESULTS: CLIENTS API VIEW



Client List

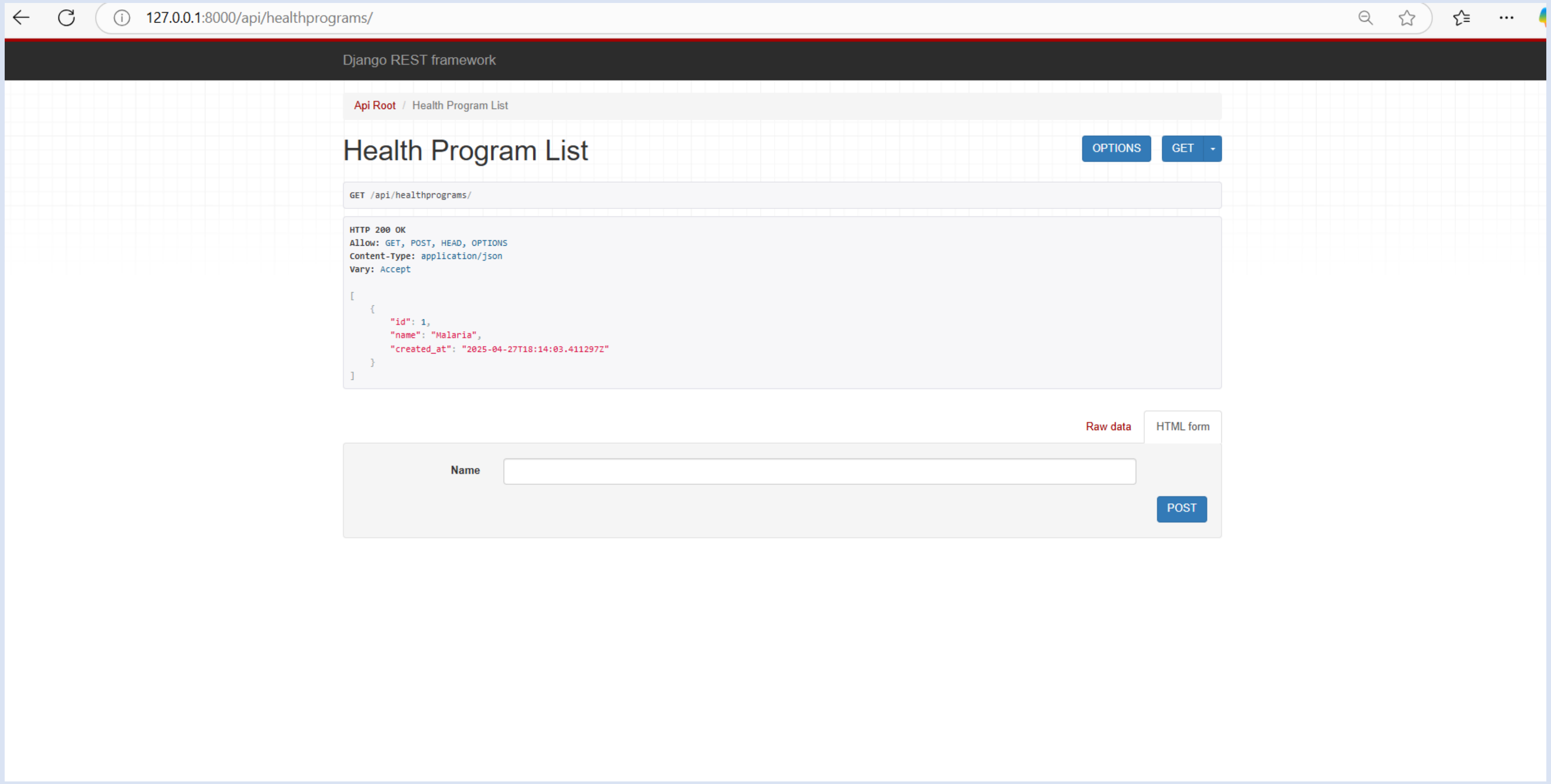
```
GET /api/clients/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "first_name": "Solomon",
        "last_name": "Kipkirui",
        "age": 21,
        "location": {
            "type": "Point",
            "coordinates": [
                -60.187483,
                -15.555286
            ]
        }
    }
]
```

First name	
Last name	
Age	
Location	

RESULTS: CLIENTS API VIEW


127.0.0.1:8000/api/healthprograms/
Django REST framework
Api Root / Health Program List
OPTIONS
GET
Raw data
HTML form
Name
POST

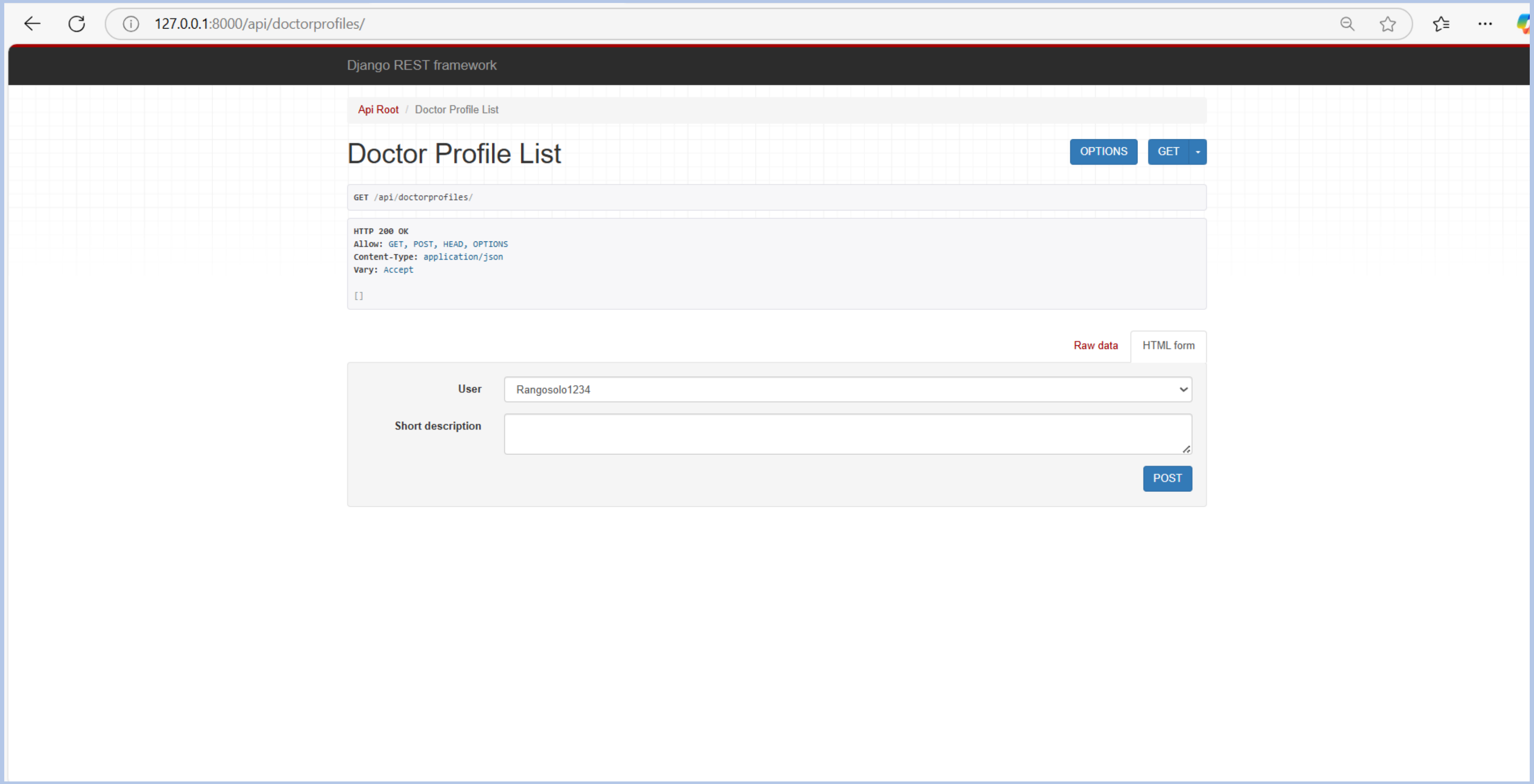
Health Program List

```
GET /api/healthprograms/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "name": "Malaria",
        "created_at": "2025-04-27T18:14:03.411297Z"
    }
]
```

RESULTS: DOCTOR PROFILES API VIEW


127.0.0.1:8000/api/doctorprofiles/
Django REST framework
Api Root / Doctor Profile List
OPTIONS
GET
Raw data
HTML form
POST

Doctor Profile List

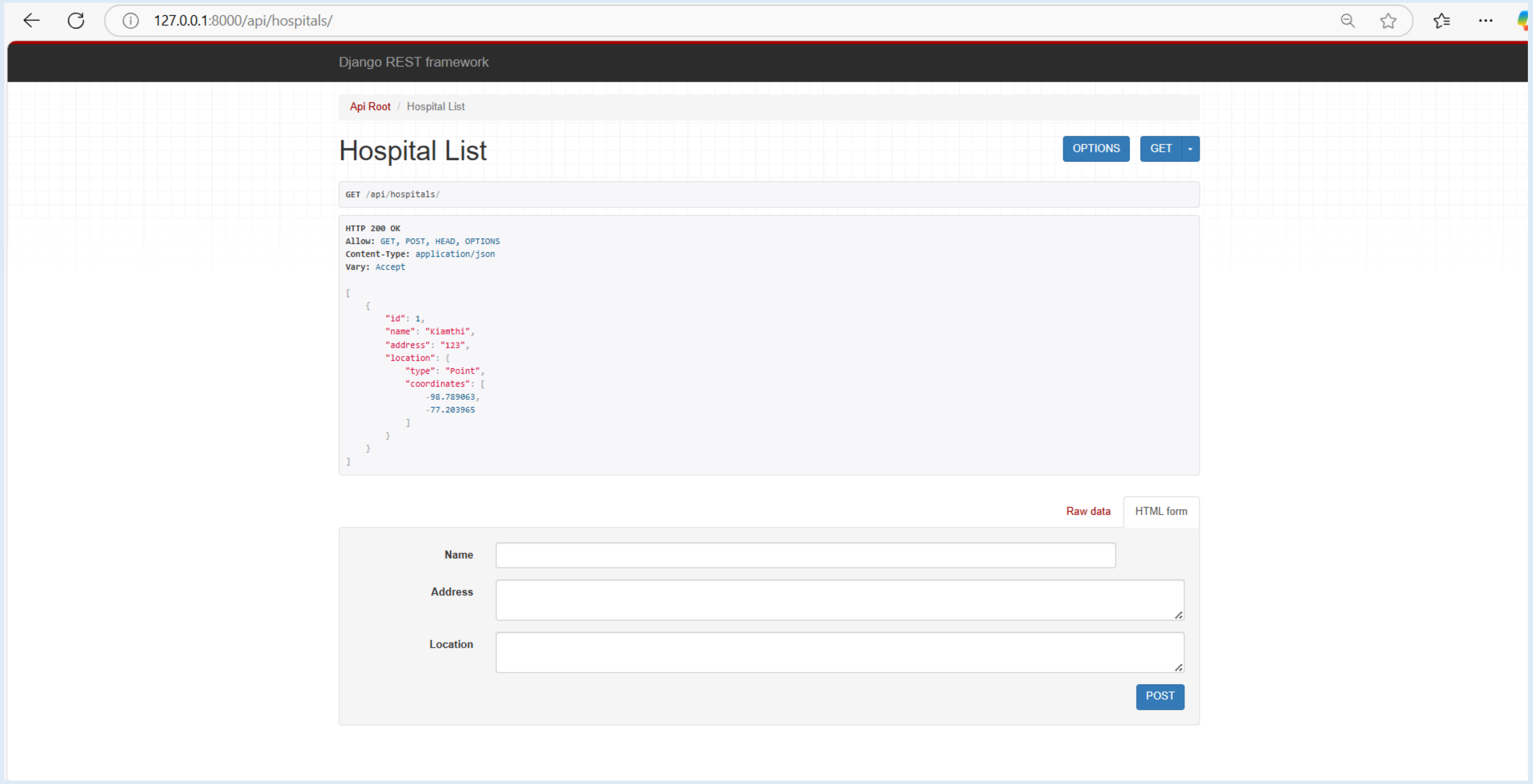
```
GET /api/doctorprofiles/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[]
```

User	Rangosolo1234
Short description	

RESULTS: HOSPITAL LIST API VIEW


127.0.0.1:8000/api/hospitals/

Django REST framework

Api Root / Hospital List

Hospital List

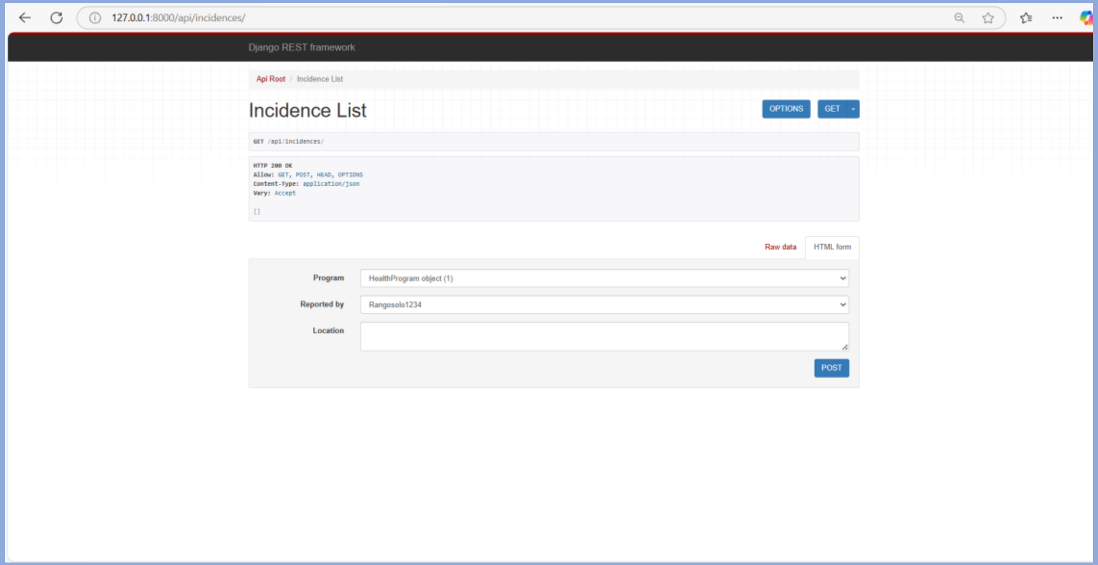
```
GET /api/hospitals/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "name": "Kiamthi",
        "address": "123",
        "location": {
            "type": "Point",
            "coordinates": [
                -98.789063,
                -77.203965
            ]
        }
    }
]
```

Name	
Address	
Location	

RESULTS: INCIDENCES AND BLOGPOSTS API VIEW


127.0.0.1:8000/api/incidences/

Django REST framework

Api Root / Incidence List

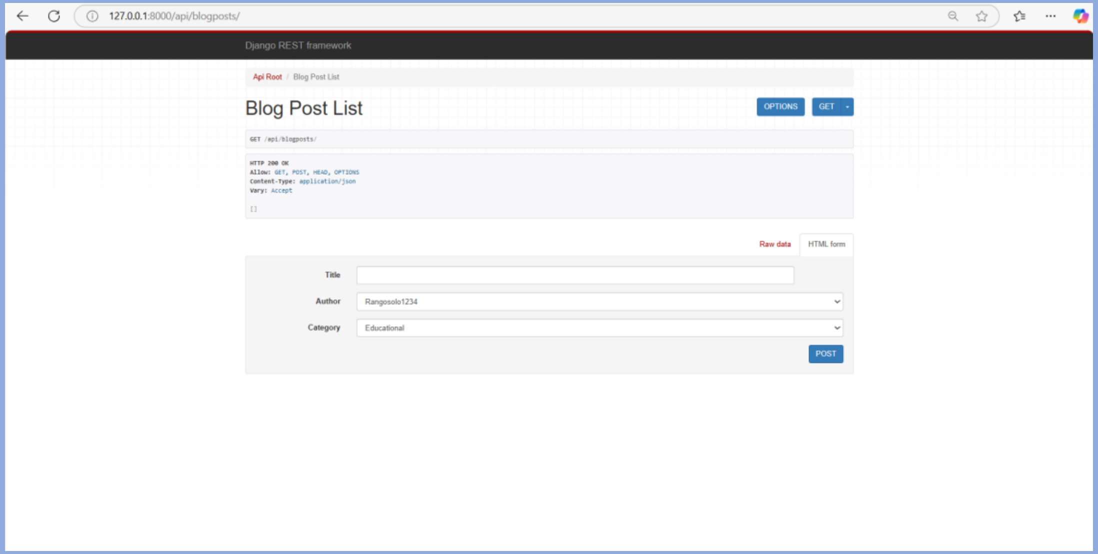
Incidence List

```
GET /api/incidences/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[]
```

Program	HealthProgram object (1)
Reported by	Rangosolo1234
Location	


127.0.0.1:8000/api/blogposts/

Django REST framework

Api Root / Blog Post List

Blog Post List

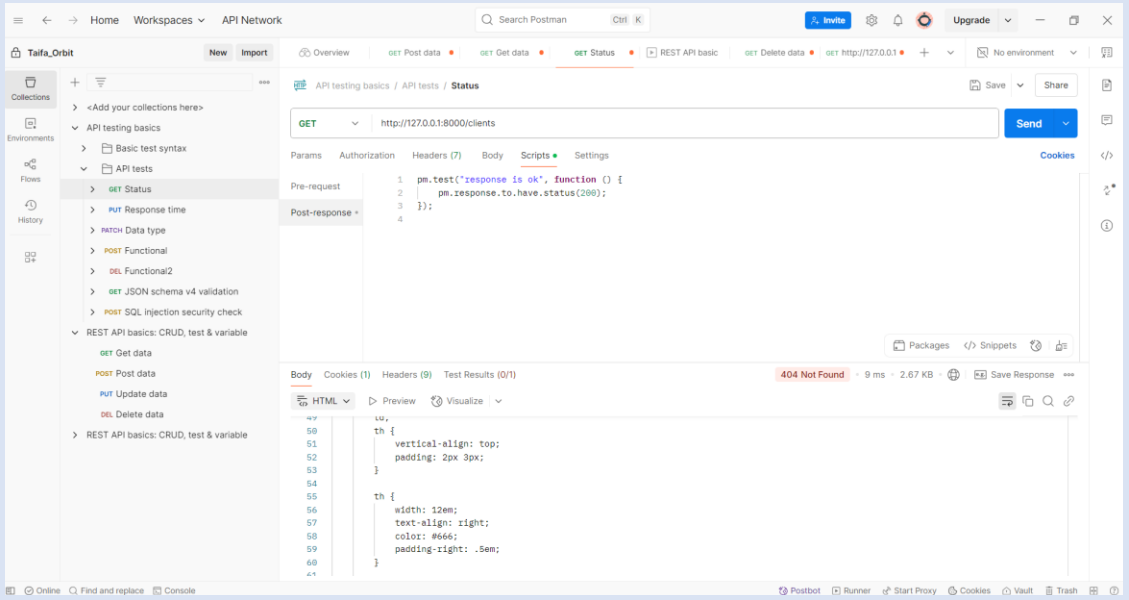
```
GET /api/blogposts/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[]
```

Title	
Author	Rangosolo1234
Category	Educational

TESTING ENDPOINTS WITH POSTMAN



API testing basics / API tests / Status

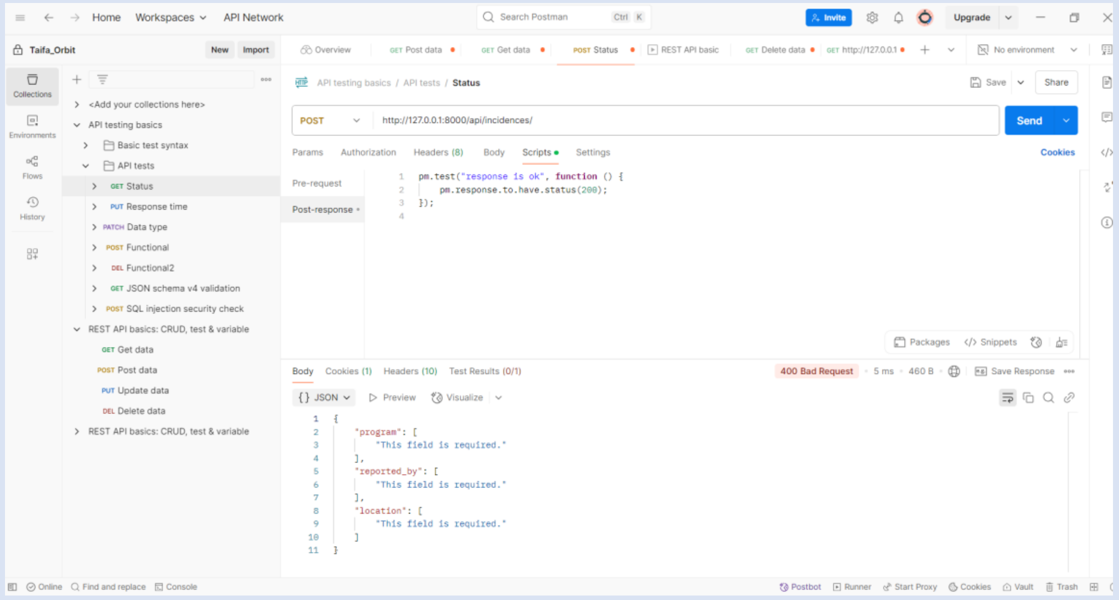
GET http://127.0.0.1:8000/clients

```
pm.test("response is ok", function () {
    pm.response.to.have.status(200);
});
```

404 Not Found · 9 ms · 2.67 KB

```
    td,
    th {
        vertical-align: top;
        padding: 2px 3px;
    }

    th {
        width: 12em;
        text-align: right;
        color: #666;
        padding-right: .5em;
    }
```



API testing basics / API tests / Status

POST http://127.0.0.1:8000/api/incidences/

```
pm.test("response is ok", function () {
    pm.response.to.have.status(200);
});
```

400 Bad Request · 5 ms · 460 B

```
{
    "program": [
        "This field is required."
    ],
    "reported_by": [
        "This field is required."
    ],
    "location": [
        "This field is required."
    ]
}
```