

Portal for Doctor and patient

Description:

Build REST API using python Django web service to manage doctor and patient queries. Doctors and patients both are treated as a user.

Requirements:

1. Must have Separate login and separate services (Both doctor and patients)
2. Should use the Django user model and store the patient as well as the doctor.
3. Should extend the model for storing the other user-related details.
4. The patient should ask for suggestions for medicines
5. The doctor should be able to respond to the suggestion of patients. (Like prescribing medicines)
6. History of suggestions for both doctor and patient.
7. Proper API restriction should be handled. (Patient should not be able to access Doctor API - Vice versa)
8. One doctor can have many patients.
9. The doctor should be able to give suggestion to any patient even if the patient is not asked any suggestion.

API's

1. Login API
2. Create a User API (Doctor and patient)
3. Medicine suggestion API (Patient)
4. Medicine respond API (Doctor)
5. Medicines history API (Patient)
6. Patients history API (doctor)
7. Health Suggestion to Patient by Doctor API (Doctor)

To Run this Django Project:

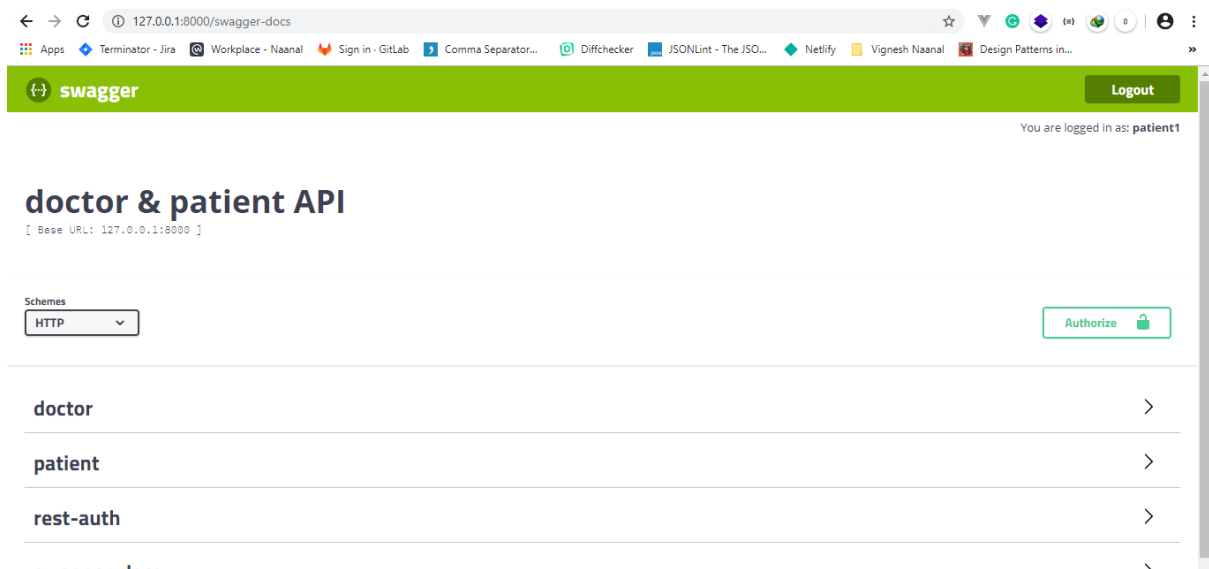
1. Activate a virtual env
2. Type `python manage.py runserver`

I have developed a project in window 10. So this project virtual env run only windows env.

Please verify the virtual environment works correctly or not. if not works virtual env you need to create virtual env first and install a requirement

1. Virtualenv Venv
2. Source `venv/bin/activate` (for ubuntu)
3. `cd Venv/Scripts/` and activate (for windows)
4. `pip install -r requirement.txt`

You can view available api in ' /swagger-docs '



Login APIs

- 1) /rest-auth/login/
- 2) /rest-auth/logout/
- 3) /rest-auth/password/change/
- 4) /rest-auth/registration/
- 5) /rest-auth/user/

Patient Suggestions APIs

For Patient

- 1) /patient/suggestion-request/

For Doctor

- 2) /patient/suggestion-response/

For Doctor and Patient

- 3) /patient/patient-doctor-suggestion-history/

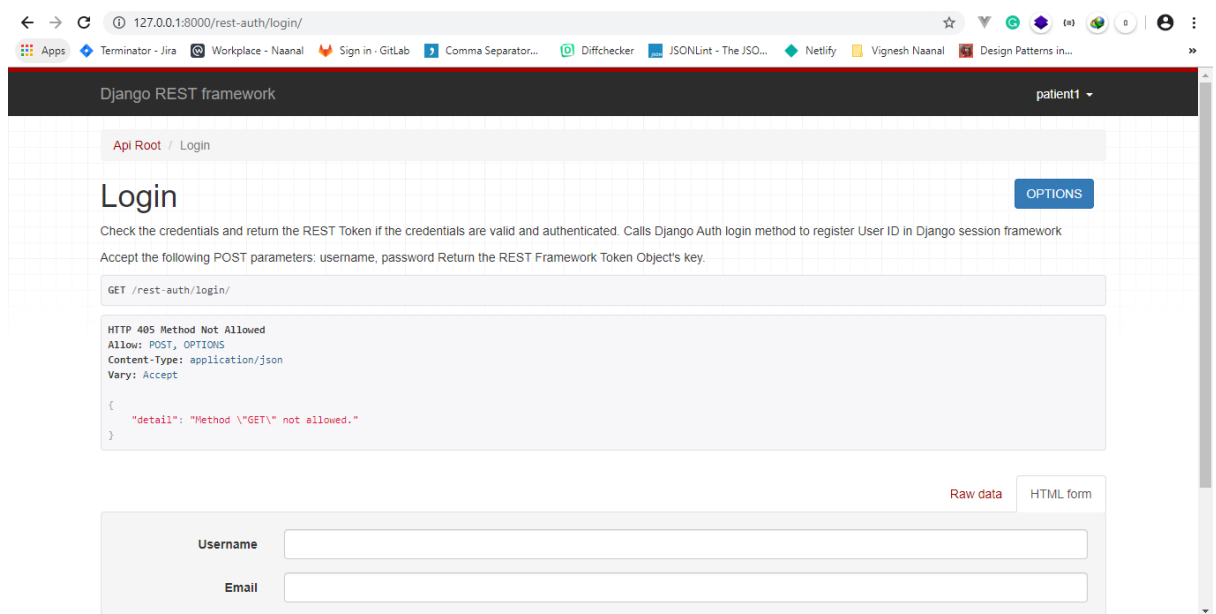
Doctor Suggestions

- 1) doctor/suggestion-by-doctor
- 2) doctor/suggestion-to-patient

Login APIs

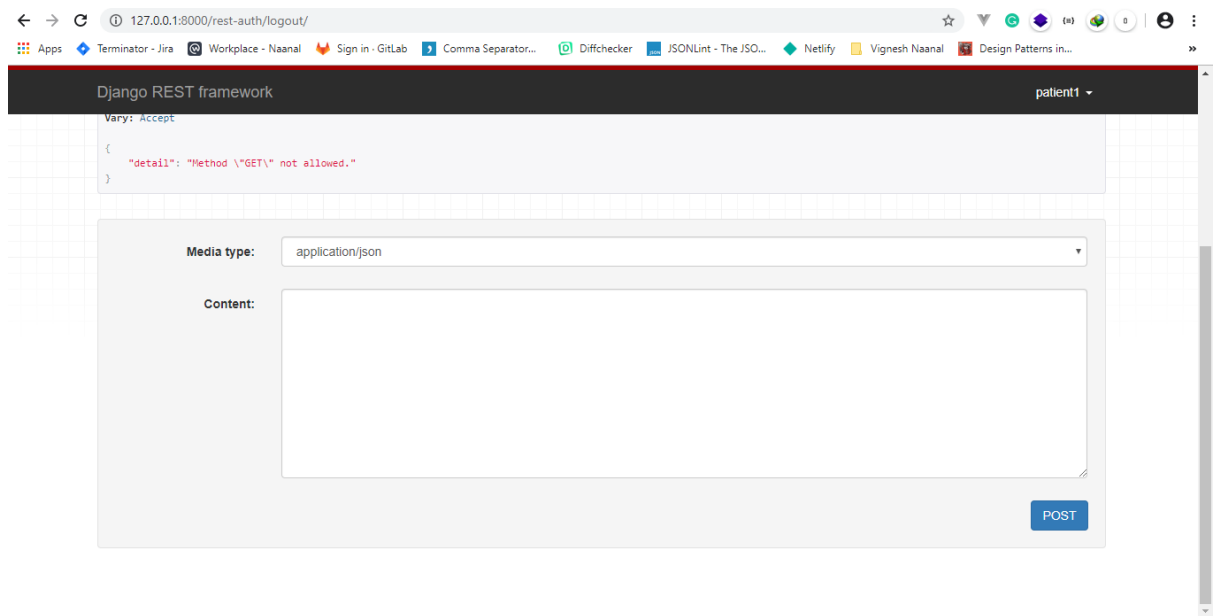
API = ‘ /rest-auth/login/ ’

- This API used to login a User’s like patient and doctor.
- Both user can use this api for login



API = ‘ /rest-auth/logout/ ’

- Used for logout purpose

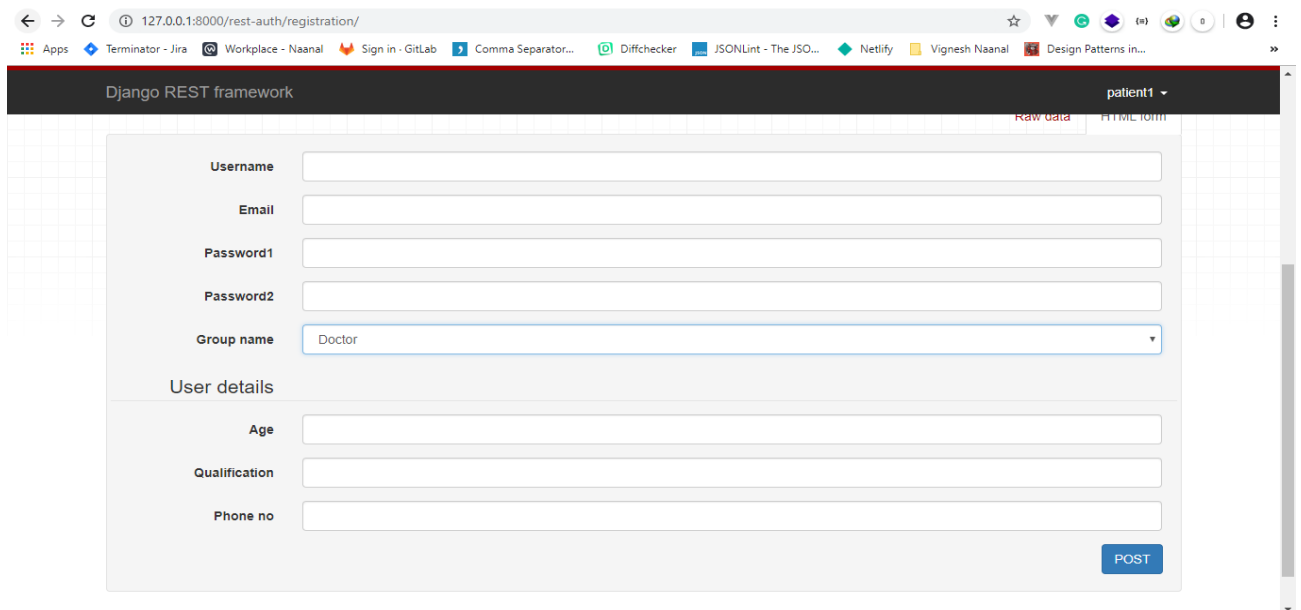


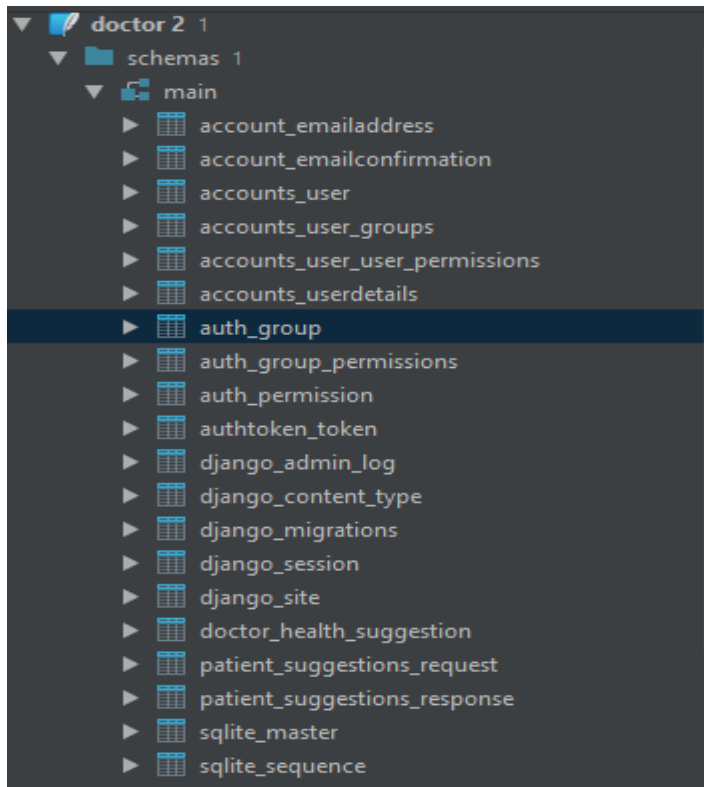
API = /rest-auth/registration/

- For register a user. You can register a user with one group permission, Patient or Doctor
- User can extend the user details

Tables:

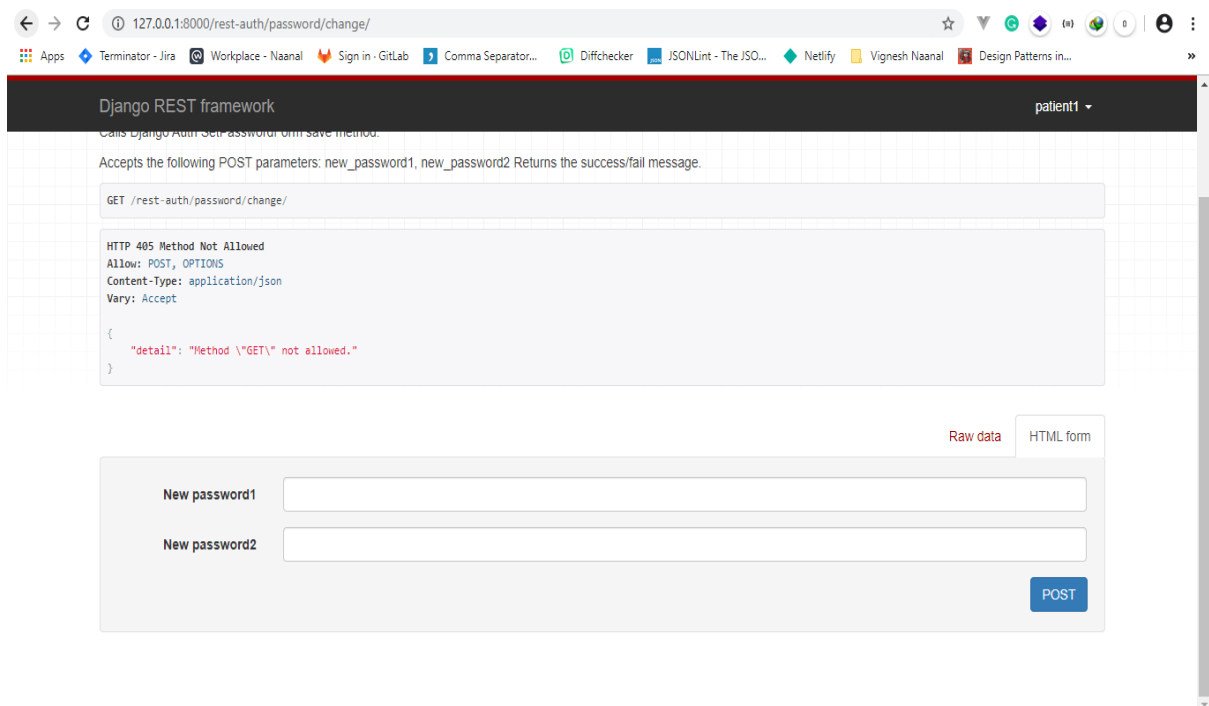
1. ‘accounts_user’ this table is the primary for user login and password.
2. ‘accounts_userdetails’ this table used for extend the user details.
3. ‘ auth_group ‘ store the group name- patient, Doctor
4. ‘ accounts_user_group ‘ the group is mapped to user.





API = /rest-auth/password/change/

- This api used for logged user can change a password



API = rest-auth/user/

- This api can used for user can change a extra details of user. Like age, phone number and education

Django REST framework patient3 ▾

Api Root / User Details

User Details OPTIONS GET ▾

Reads and updates UserModel fields Accepts GET, PUT, PATCH methods.
Default accepted fields: username, first_name, last_name Default display fields: pk, username, email, first_name, last_name Read-only fields: pk, email
Returns UserModel fields.

GET /rest-auth/user/

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

{
  "pk": 14,
  "username": "patient3",
  "email": "",
  "first_name": "",
  "last_name": "",
  "user_details": {
    "age": 25,
    "qualification": "MCA",
    "phone_no": 7898523641
  }
}
```

Raw data HTML form

Username
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name

Last name

User details

Age

Qualification

Phone no

PUT

Patient Suggestions APIs

For Patient

API = /patient/suggestion-request/

- This api can used for patient only. Don't have a access for doctors.
- Patient view own request only. Don't view other patient request suggestions
- Patient can ask for a medicine suggestion. Medicine_name, disease_info, remarks.
- Patient can't delete a request directly. Instant of use disable an request using is_disable flag.
- Any doctor can reply to the patient request suggestions.

127.0.0.1:8000/patient/suggestion-request/

Django REST framework patient1

```
{  "medicine_name": "Dolo 650",  "disease_info": "can I use for headache?",  "remarks": "is 650 power ok for my age",  "created_user_name": "patient1",  "is_disable": false}
```

Raw data HTML form

Medicine name

Disease info

Remarks

Is disable ☐

POST

API = /patient/suggestion-response/

- Any doctors can response a patient suggestion request
- Doctor can only view own suggestions.
- This api can used for doctors only. Don't have an access for patients.
- Doctor can't delete a request directly. Instant of use disable a request using is_disable flag.

Patient Suggestions Response Li x +

127.0.0.1:8000/patient/suggestion-response/

Django REST framework doctor1

```
{  "medicine_name": "Dolo 650",  "doctor_response": "Yes, you can use it. Dolo 650 is a drug that relieves pain and fever. The active ingredient in this drug is paracetamol. It acts by increasing",  "suggestion_request": 1,  "created_user_name": "doctor1",  "is_disable": false,  "medicine_name": "Dolo 650"}
```

Raw data HTML form

Doctor response

Suggestion request

Is disable ☐

POST

API = patient/patient-doctor-suggestion-history/

- This API used for history of patient and doctor suggestion history
- Can access an API only authenticated members. Both doctors and patient can access this API.
- This suggestion history API, this API data is must be in particular condition
Request made by me (or) response made by me.

Patient view

The screenshot shows a web browser at the URL `127.0.0.1:8000/patient/patient-doctor-suggestion-history/`. The page title is "Suggestion History List" and the user is logged in as "patient1". The API endpoint is `GET /patient/patient-doctor-suggestion-history/`. The response is an HTTP 200 OK with `Allow: GET, HEAD, OPTIONS` and `Content-Type: application/json`. The JSON response contains a list of two suggestion objects:

```
[
  {
    "medicine_name": "Paracetamol",
    "disease_info": "what kind of medicine this?",
    "remarks": null,
    "created_datetime": "2020-03-28T06:47:39.736528Z",
    "doctor_response": []
  },
  {
    "medicine_name": "Dolo 650",
    "disease_info": "Can I use for headache?",
    "remarks": "Is 650 power ok for my age?",
    "created_datetime": "2020-03-27T10:56:47.667888Z",
    "doctor_response": [
      {
        "doctor_response": "Yes, you can use it. Dolo 650 is a drug that relieves pain and fever. The active ingredient in this drug is paracetamol. It acts by inhibiting the production of prostaglandins, which are chemicals that cause pain and inflammation.",
        "created_by_username": "doctor1",
        "created_datetime": "2020-03-27T11:37:25.134444Z"
      },
      {
        "doctor_response": "Age limit for Dolo 650\r\nThis product is not recommended for use in children younger than 6 years due to an increased risk of serious side effects.",
        "created_by_username": "doctor1",
        "created_datetime": "2020-03-27T11:37:25.134444Z"
      }
    ]
  }
]
```

Doctor View

The screenshot shows the same API endpoint `127.0.0.1:8000/patient/patient-doctor-suggestion-history/` but from the perspective of a doctor, logged in as "doctor1". The response is an HTTP 200 OK with the same headers. The JSON response contains the same list of two suggestion objects, but the first object's `doctor_response` is now an array containing two responses from "doctor1":

```
[
  {
    "medicine_name": "Dolo 650",
    "disease_info": "Can I use for headache?",
    "remarks": "Is 650 power ok for my age?",
    "created_datetime": "2020-03-27T10:56:47.667888Z",
    "doctor_response": [
      {
        "doctor_response": "Yes, you can use it. Dolo 650 is a drug that relieves pain and fever. The active ingredient in this drug is paracetamol. It acts by inhibiting the production of prostaglandins, which are chemicals that cause pain and inflammation.",
        "created_by_username": "doctor1",
        "created_datetime": "2020-03-27T11:37:25.134444Z"
      },
      {
        "doctor_response": "Age limit for Dolo 650\r\nThis product is not recommended for use in children younger than 6 years due to an increased risk of serious side effects.",
        "created_by_username": "doctor1",
        "created_datetime": "2020-03-27T11:58:16.808678Z"
      }
    ]
  },
  {
    "medicine_name": "Paracetamol",
    "disease_info": "what kind of medicine this?",
    "remarks": null,
    "created_datetime": "2020-03-28T06:47:39.736528Z",
    "doctor_response": []
  }
]
```

Doctor Suggestions

API = doctor/suggestion-by-doctor

- This API used for doctor give a Health Suggestion to Patient.
- Patient can't access this API
- Doctor should select a Patient and enter a health tips for patient and submit a suggestion

The screenshot shows a web browser window displaying the Django REST framework API interface. The URL bar shows the address `127.0.0.1:8000/doctor/suggestion-by-doctor/`. The page title is "Doctor Health Suggestion By Doctor List". Below the title, there is a search bar with the text `GET /doctor/suggestion-by-doctor/`. The response status is "HTTP 200 OK" with headers: `Allow: GET, POST, HEAD, OPTIONS`, `Content-Type: application/json`, and `Vary: Accept`. The response body is an empty array `[]`. Below the response, there is a form with the following fields: "Suggestion patient id" (a dropdown menu with "patient1" selected), "Health tips" (a text input field), and "Is disable" (a checkbox). A "POST" button is located at the bottom right of the form. The interface also includes tabs for "Raw data" and "HTML form".

API = doctor/suggestion-to-patient

- This API is used for patient can view a doctor advice to patient
- Only patient can access this API
- This API is read only

The screenshot shows a web browser window displaying the Django REST framework API interface. The URL bar shows the address `127.0.0.1:8000/doctor/suggestion-to-patient/`. The page title is "Doctor Health Suggestion For Patient List". Below the title, there is a search bar with the text `GET /doctor/suggestion-to-patient/`. The response status is "HTTP 200 OK" with headers: `Allow: GET, HEAD, OPTIONS`, `Content-Type: application/json`, and `Vary: Accept`. The response body is a JSON object:

```
[{"suggestion_patient_name": "patient1", "created_datetime": "2020-03-27T15:16:28.733319Z", "health_tips": "don't take this medicine.\r\nAcetaminophen (Tylenol)\r\nCommon names for Acetaminophen include Tylenol, Mapap, and Feverall. This drug is regularly", "is_disable": false}]
```

. The interface also includes tabs for "Raw data" and "HTML form".

User Name and Passwords

Admin:

Username: vignesh

Password :vignesh

Doctor Logins:

Username: doctor1

Password: vigneshd1

Username: doctor2

Password: vigneshd2

Username: doctor3

Password: vigneshd3

Patient Logins:

Username: patient1

Password: vigneshp1

Username: patient2

Password: vigneshp2

Username: patient3

Password: vigneshp3