Setup

Use python 3.11 or greater ya

1. Activate venv

```
python -m venv venv
```

2. Install requirements

```
pip install -r req.txt
```

3. Run Django Migrate. This creates a db locally on your pc

```
python manage.py migrate
```

4. Create your local superuser

```
python manage.py createsuperuser
```

5. Start Django

```
python manage.py runserver
```

Creating your custom object model

1. Open the models.py file (`templateApp\templateProject\models.py`)
2. Create the class with the fields
3. We can add relation such as user who created this object by adding

```
user = models.ForeignKey(User, on_delete=models.CASCADE,
related_name="users")
```

4. After you finish the fields, we need to migrate

```
python manage.py makemigrations templateProject
```

5. It would return an error like

```
AttributeError: 'function' object has no attribute 'generate'
```

6. We need to alter the migrations python file

```
default=nanoid.generate.generate,

# change to
default=nanoid.generate,
```

7. Then we can migrate

```
python manage.py migrate
```

8. Now check out the Django Admin Console

```
python manage.py runserver
```

9. Go to http://localhost:8000/admin/
10. Now noticed that there are no "Case" object option
11. Need to add it to the admin site in the url (templateApp\templateProject\urls.py), add the following code so it looks like

```
from django.contrib import admin
from django.urls import path

from templateProject.models import Case


admin.site.register(Case)
urlpatterns = [
    path("admin/", admin.site.urls),
]
```

12. Go back to admin portal and you can create some Case from the console

Creating Views

1. Now create a serializer for the Case object
2. Create a view set
    1. We define some classes for authentication and permissions

    2. There are some filtering and ordering classes as well

3. Register the view in the urls, your urls.py should look like

```python
from django.contrib import admin
from django.urls import path, include
from rest_framework import routers
from templateProject.models import Case, CaseViewSet


admin.site.register(Case)

router = routers.DefaultRouter()
router.register(prefix=r"cases", viewset=CaseViewSet, basename="cases")

urlpatterns = [
    path("admin/", admin.site.urls),
    path(r"api/auth/", include("knox.urls")),
    path("", include(router.urls)),
]
```

Using the Endpoint

1. Generate token

```bash
curl --request POST \
  --url http://localhost:8000/api/auth/login/ \
  --header 'Authorization: Basic ZXZvdGlhbnVzOmV2b3RpYW51cw==' \
  --header 'User-Agent: insomnia/10.1.0'
```

Response

```json
{
  "expiry": "2024-10-25T12:09:59.484114Z",
  "token": "870a26bdbcbadf902d29c0f493b09548b779971aaeff41e071c27ec1f27c0b83"
}
```

2. Use Token to query

```bash
curl --request GET \
  --url 'http://localhost:8000/cases?ordering=description&limit=15&offset=0' \
  --header 'Authorization: Token
  870a26bdbcbadf902d29c0f493b09548b779971aaeff41e071c27ec1f27c0b83' \
  --header 'User-Agent: insomnia/10.1.0'
```

Response

```json
{
    "count": 2,
    "next": null,
    "previous": null,
    "results": [
        {
            "case_id": "041UO6y2WgL9JhpRt0h16",
            "description": "bbb",
            "created_at": "2024-10-25T02:14:41.290425Z",
            "field_custom": true,
            "updated_at": "2024-10-25T02:14:41.290425Z"
        },
        {
            "case_id": "m685S9S3_Vpq-HMvPmyBx",
            "description": "zz",
            "created_at": "2024-10-25T02:11:07.763527Z",
            "field_custom": true,
            "updated_at": "2024-10-25T02:11:07.764530Z"
        }
    ]
}
```

3. Ordering and search (filtering)

   - Putting a `-` in front of the field will reverse the
   - Putting a `search` query you can get the `description` field

```
curl --request GET \
  --url 'http://localhost:8000/cases?
search=z&ordering=description&limit=15&offset=0' \
  --header 'Authorization: Token
870a26bdbcbadf902d29c0f493b09548b779971aaeff41e071c27ec1f27c0b83' \
  --header 'User-Agent: insomnia/10.1.0'
```