// Bich Tasna3 Project Django

// Create a Virtual Environment

Etape 1 : virtualenv ismelit7ibtsamih or python -m venv ismelit7ibtsamih

// Activate Virtual Environmen

Etape 2 : Temchi lil scripts tikteb activate cd scripts /activate

// Install Django

Etape 3 : pip install django

// Start a New Django Project

Etape 4 : django-admin startproject project

//Run the Development Server

Etape 5 : python manage.py runserver //bich t5adem serveur

// Creating a Django App

// Start a New App

Etape 1 : python manage.py startapp ismelit7ib3lih

Etape 2 : Add App to INSTALLED\_APPS in project/settings.py:

Etape 3 : **Configure URL Routing** In project/urls.py: | In ism folder/urls.py:

// urls tasna3 folder manuelement

Etape 4 : **Creating Views** In ism folder/views.py

// Bich ya9ralek eli ktibtou fil base de donner wala ta3mel modifier 7aja lezim !!!

// Apply Migrations

Etape Existe : python manage.py migrate

// idha ken 5rajlek message fil cmd

System check identified no issues (0

silenced).

You have 18 unapplied migration(s).

Your project may not work properly until you

apply the migrations for app(s): admin, auth,

contenttypes, sessions.

// Additional Commands Django local database relation

// bich ta3ti ism w password lil

Etape 1 : Create a Superuser : python manage.py createsuperuser

//bich t7ot base de donner fi django

//Crée les fichiers de migration basés sur les changements dans les modèles

Etape 2 : Make Migrations and Migrate : python manage.py makemigrations

// Applique ces migrations à la base de données

python manage.py migrate

//**Rôle :** Collecter les fichiers statiques de toutes les applications et les stocker dans un répertoire centralisé.

Etape 3 : Collect Static Files : python manage.py collectstatic

database mte3ik bich matkassarch rassek barcha !!!

// Ism admin email ladhari@gmail.com

// password aziz52995033

// cree une Projet next js

Etape 1 : **Creating a Next.js Project : npx create-next-app@latest ismproject**

Etape 2 : Adding a New Function Component :

Etape 3 : **Creating a Link to Your New Page bich tasna3 link thizek lil page**

Etape 4 : **Metadata**

// COMMENT FAIRE UNE RELATION ENTRE DJANGO ET NEXT JS

Etape 0 : Tasna3 dossier ism Projet fi wistou Dossier Backend dossier frontend

Etape 1 : virtualenv ismelit7ibtsamih

Etape 2 : Temchi lil scripts tikteb activate cd scripts /activate

Etape 3 : pip install django

Etape 4 : django-admin startproject Myproject

Etape 5 : pip install djangorestframework

Etape 6 : zid rest\_framework lil INSTALLED\_APPS fi settings.py

Etape 7 : T7ot base de donner mte3ik mta3 local exemple ... chma3neha Serializers ??

Etape 8 : Tasna3 Views w URLs ista3mel ViewSets Routers

urls

ViewSets

Tasna3 dossier

Ismou views.py

# myapp/serializers.py

from rest\_framework import serializers

from .models import Book

class BookSerializer(serializers.ModelSerializer):

class Meta:

model = Book

fields = ['id', 'title', 'author', 'published\_date']

# myapp/models.py

from django.db import models

class Book(models.Model):

title = models.CharField(max\_length=100)

author = models.CharField(max\_length=100)

published\_date = models.DateField()

def \_\_str\_\_(self):

return self.title

# myapp/urls.py

from django.urls import path, include

from rest\_framework.routers import DefaultRouter

from .views import BookViewSet

router = DefaultRouter()

router.register(r'books', BookViewSet)

urlpatterns = [

path('', include(router.urls)),

]

# myapp/views.py

from rest\_framework import viewsets

from .models import Book

from .serializers import BookSerializer

class BookViewSet(viewsets.ModelViewSet):

queryset = Book.objects.all()

serializer\_class = BookSerializer

Etape 09 : npx create-next-app@latest my-next-app

Etape 10 : Pour Django API Utilise fetch ou axios getServerSideProps ???

Etape 11 : pip install django-cors-headers

Etape 12 : CORS\_ALLOWED\_ORIGINS Zidha

Etape 13 : python manage.py runserver Pour Django et npm run dev pour next js

myproject/settings.py

INSTALLED\_APPS = [

...

'rest\_framework',

'myapp',

]

django-admin startproject myproject

cd myproject

python manage.py startapp myapp

pip install djangorestframework

django-admin startproject myproject

cd myproject

python manage.py startapp myapp

pip install djangorestframework

# myapp/serializers.py:

from rest\_framework import serializers

from .models import Item

class ItemSerializer(serializers.ModelSerializer):

class Meta:

model = Item

fields = '\_\_all\_\_'

# myapp/views.py

from rest\_framework import viewsets

from .models import Item

from .serializers import ItemSerializer

class ItemViewSet(viewsets.ModelViewSet):

queryset = Item.objects.all()

serializer\_class = ItemSerializer

# myapp/urls.py

from django.urls import path, include

from rest\_framework.routers import DefaultRouter

from .views import ItemViewSet

router = DefaultRouter()

router.register(r'items', ItemViewSet)

urlpatterns = [

path('', include(router.urls)),

]

# myproject/urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('api/', include('myapp.urls')),

]

# myapp/views.py

from rest\_framework import viewsets

from .models import Item

from .serializers import ItemSerializer

class ItemViewSet(viewsets.ModelViewSet):

queryset = Item.objects.all()

serializer\_class = ItemSerializer

import { useEffect, useState } from 'react';

export default function Home() {

const [items, setItems] = useState([]);

useEffect(() => {

fetch('http://localhost:8000/api/items/')

.then(response => response.json())

.then(data => setItems(data));

}, []);

return (

<div>

<h1>Items</h1>

<ul>

{items.map(item => (

<li key={item.id}>{item.name}</li>

))}

</ul>

</div>

);

}

**pip install django-cors-headers**

INSTALLED\_APPS = [

...

'corsheaders',

]

MIDDLEWARE = [

...

'corsheaders.middleware.CorsMiddleware',

]

CORS\_ALLOWED\_ORIGINS = [

"http://localhost:3000", # العنوان المحلي لتطبيق Next.js

]