Django

Web application framework from python

Own database connector

MVT architecture 🡪 model view template

Model 🡪 database information

View 🡪 at runtime pull data from model , do the processing and send to template

Template 🡪 Basically html , css , js . but it has placeholders for every thing, eg: title , content etc.

Whole application is based on MVT.

Flow 🡪

Home -> home.urls -> urls.py of home -> views.py -> run corresponding function

Important 🡪 Only public files should be kept in static folder. Files with confidential details should not be kept there

We can transfer data form model to index.html with help of context. Eg: to fetch data from database and send it to index.html

python manage.py runserver 🡪 commad use to start the app

python manage.py migrate 🡪 to create tables for admin handle

python manage.py createsuperuser 🡪 to create super user for admin handle

base.html 🡪 we create this file usually to keep the nav bar and footer over here

{% extends 'base.html' %} 🡪 to use the data from base.html

block 🡪 used to change data in base.html

csrf token 🡪 used to enable token that the request if from the website only

to create a model 🡪 create a class for it and import the model in admin.py and register it using 🡪admin.site.register(xyz) command, now add your adds name in settings.py 🡪 installed apps

python manage.py makemigrations 🡪makes a file of changes

python manage.py migrate 🡪 now to make the particular table in Django admin panel

to connect the contact database to front end form 🡪

1. In views.py Import contact 🡪 from home.models import Contact
2. In views.py create and if condition 🡪 if request.method == "POST":

    if request.method == "POST":

        name = request.POST.get('name')

        email = request.POST.get('email')

        phone = request.POST.get('phone')

        desc = request.POST.get('desc')

        serial+1

        contact = Contact(name =name, email = email, phone =phone , desc = desc , date = datetime.today(), serial = serial)

        contact.save()

        messages.success(request, 'Profile details sent')