1)creer virtualenv: python -m venv 'name'

2)cd nom

2)activer l'env Scripts\activate

3) install django pip install django

5)creer le projet: django-admin startproject nameproj

6) cd nameproj

7) py.manage.py startapp esmapp

7)run server py manage.py runserver

projet->setting.py->installed app-> "nomapp.apps.NomappConfig"

exemple "employee.apps.EmployeeConfig"

**projet->urls** from django.urls import path (hethy mawjouda)

from django.urls.config import include

path('',include('esmapp.urls') //path('',include('employee.urls')

app->new file urls.py from django.urls import path

from . import views

urlpatterns =[

path('emp/', views.emp, name='emp'), # Ajoutez '/' à la fin pour les URL

path('show/', views.show, name='show'),

path('edit/<int:id>/', views.edit, name='edit'),

path('update/<int:id>/', views.update),

path('delete/<int:id>/', views.destroy),]

**new folder feee app** templates

setting.py ->templates ->**Dirs**  [os.path.join(BASE\_DIR,)],

import os

from pathlib import Path //hethy mawjouda

**fel app tasna3 file : forms.py from django import forms**

**from .models import Employee**

**class EmployeeForm(forms.ModelForm):**

**class Meta:**

**model=Employee**

**fields="\_\_all\_\_"**

**app->views.py** from django.shortcuts import render,redirect

from .models import Employee

from .forms import EmployeeForm

**models.py:** from django.db import models

class Employee(models.Model):

eid = models.CharField(max\_length=20)

ename = models.CharField(max\_length=20)

email = models.EmailField()

econtact = models.CharField(max\_length=20)

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

return "%s" % (self.ename) # Affiche le nom d'employee

class Meta:

db\_table = "employee" # Nom de la table dans la base de données

**tsajel models //mithel lzm tkoun fel projet crud** : python manage.py makemigrations

python manage.py migrate

**admin.py from django.contrib import admin**

**from .models import Employee**

**admin.site.register(Employee)**

////////creer un employees dans la base admin avant de create superuser avec :

/// python manage.py createsuperuser

from django.contrib import admin

from .models import Employee

admin.site.register(Employee)

**hajet plus**

**db.sql : base de donnéé creer par django**

**migrate: bch t7ot jadwel fee base de données**

**code . : y7l vsc**

**setting:::::::::::::::::[**

**->ssecret key**

**->installed apps ayy app tet9ayed ghadi**

**->templates :frontend**

**]**

**urls[**

**path('gro**

[**www.fc.com/group**](http://www.fc.com/group)

**]**

**app 3ibara page [**

**views yohkom fee hethom koll yab3th w y9bl des données**

**models; base de donne**

**templates : front end**

**urls ena nasn3ou**

**admin teb3 createur de site]**

**8) creer app py manage.py startapp nomapp**

**9) tsajel app proj----setting----unstalled apps 'appname.apps.NameConfig',**

**proj----urls------ /emp path,include**

**tasna3 urls.py**

**----urlpatterns path('page/',include('appname.urls'))**

**10) from django.http httpResponse ki yod5ol yal9a toul heeloo wrld**

**def c(request):**

**return httpResponse('hello world')**

**11) views creer une fonction def ind (request):**

**12)urls app from django.urls import path**

**from . import views ( . :urls bjanb views)**

**urlpatterns = [**

**path('esm elli baad slach yaani fonction yomkin tkoun fargha yaani page initiale', views.esmfonction ,name='esmfonction'),**

**]**

**urls app path(' ',include('appname.urls')) yhlha toul mnghir nktb chy**

**urls projet path('in/',include('appname.urls')) page/in**

**13)views (concernce fonction) from django.shortcuts import render**

**14)app/ bara new folder----templates**

**15)settings.py import os**

**----templates---'DIRS':[ os.path.join(BASE\_DIR,'templates')]**

**//en cas templates bara naaaml folder besm app fee wost templates**

**16) kol haja fee templates torbtha bel view**

**hetha chnwa tthaher**

**17)slice:::: ya5ou zouz asemi khw**

**18) {%tag%}**

**{%endtag%}**

**heritage de couleur::: base.html creer une page(templates) <style> background:black<style>**

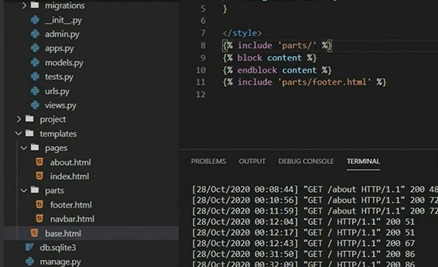
**inex.html {%extends' base.html'%} mayothhor maaaha chy ekher tktb maaaaha!!!!!!!!!!!!!!**

**donc lzm nktha feee blog::::: {% block name%}**

**<h1>hbhjgfdf**

**{%endblock%}**

**19)**



**{% include 'nom app/footer.html'%} nhothha taht elli nhb**

**20)for / if :::::::::::::::::::::::::::::::::::::::**

**{% if name=='ali' %} {% for x in name%}**

**<p>{{x}}</p>**

**{ % elif%} {%endfor%}**

**21)dans le projet en cree un dossier static: en ajout folder pour css/image/js/...**

**setting STATIC\_ROOT=os.path.join(BASE\_DIR,'static')**

**sitting ----- static----**

**STATICFILES\_DIRS=[**

**os.path.join(BASE\_DIR,'static')**

**py managep.py collectstatic thew static bara wahdha/w ayy ta8yir ta3mlou**

**22) {%load static%}**

**<link rel="stylesheet" href="{% static ' masar :css/style.css''%}"**

**<span><a href="{% url 'emp' %} >aboutt(thez lpage about**

**23)app---models**



**image---->sna3na folder (upload\_to**



**null=true ya3ni mouch lzm n3amrou najm naaaml save mnghir man3amer**

**blank=true mouvh matlub**

**24) choice**



**python manage.py makemigration //kol manaaaml colonne jdida**

**pythone manage.py migrate**

**25)**

**ki taaaaml class:**

**admin app-----> admin.site.register(esm class)**

**26)**

**from dateTime import dateTime**

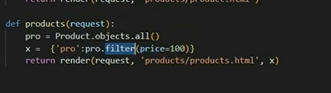


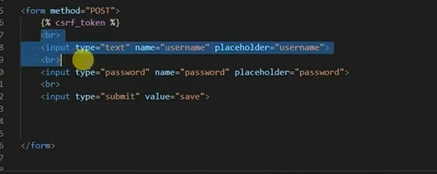
**27)**



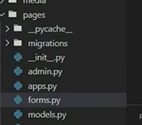


**tjib hajet hasseb price:**





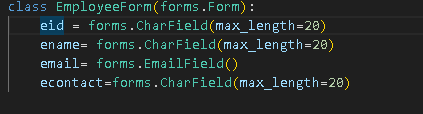
**1)app----> creer forms.py**



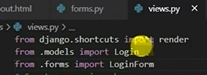
**forms.py**

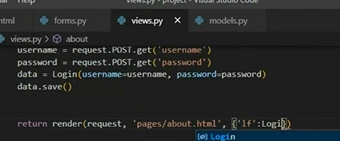






**views.py**





**23)**

**]**

**{%endif%}**

**8)**

from django.shortcuts import render,redirect

from .models import Employee

from .forms import EmployeeForm

def emp(request):

    if request.method == "POST":

        form = EmployeeForm(request.POST)

        if form.is\_valid():# Si le formulaire est valide

            form.save()#on enregistre les données si la form valide

            return redirect("/show")

    else:

        form = EmployeeForm() # Si la méthode HTTP n'est pas POST

    return render(request, 'index.html', {'form': form})

def show(request):

    employees = Employee.objects.all() # Récupère tous les employés dans la base de données

    return render(request, "show.html", {'employees': employees})# Rendu du template 'show.html' en passant la liste des employés

def edit(request, id):

    employee=Employee.objects.get(id=id) # Récupère l'employé avec l'ID spécifié dans l'URL

    return render(request,'edit.html',{'employee':employee})# Rendu du template 'edit.html' en passant l'employé récupéré

def update(request, id):

    employee = Employee.objects.get(id=id)# Récupère l'employé avec l'ID

     # Crée un formulaire avec les données soumises dans la requête POST et les valeurs existantes de l'employé

    form = EmployeeForm(request.POST, instance=employee)

    if form.is\_valid():

        form.save()

        return redirect("/show")

    return render(request, 'edit.html', {'employee': employee})

def destroy(request, id):

    employee = Employee.objects.get(id=id)# Récupère l'employé avec l'ID

    employee.delete()  # Supprime l'employé

    return redirect("/show")

////////////////////////////////////////////////////

Edit.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Update Employee</title>

    {% load static %} <!--charger les ficher static-->

    <link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body>

    <form method="POST" action="/update/{{ employee.id }}/">

        {% csrf\_token %}<!--Cela permet de protéger votre application contre les attaques CSRF-->

        <div>

            <h3>Update Details</h3>

        </div>

        <div>

            <p>Employee Id:</p>

            <div>

                <input type="text" name="eid" value="{{ employee.eid }}" required /><!-- insérer dynamiquement la valeur d'un champ d'un modèle-->

            </div>

        </div>

        <div>

            <p>Employee Name:</p>

            <div>

                <input type="text" name="ename" value="{{ employee.ename }}" required />

            </div>

        </div>

        <div>

            <p>Employee Email:</p>

            <div>

                <input type="email" name="eemail" value="{{ employee.eemail }}" required />

            </div>

        </div>

        <div>

            <p>Employee Contact:</p>

            <div>

                <input type="text" name="econtact" value="{{ employee.econtact }}" required />

            </div>

        </div>

        <div>

            <button type="submit">Update</button>

        </div>

    </form>

</body>

</html>

/////////////////////////////

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    {% load static %}

    <link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body>

   <form method="POST" action="/emp/">

    {% csrf\_token %}

    <div>

        <h3>Enter Details</h3>

        <div>

            <p>Employee Id:</p> {{ form.eid }}

        </div>

        <div>

            <p>Employee Name:</p> {{ form.ename }} <!--form.ename est utilisée pour afficher un champ spécifique d'un formulaire-->

        </div>

        <div>

            <p>Employee Email:</p> {{ form.eemail }}

        </div>

        <div>

            <p>Employee Contact:</p> {{ form.econtact }}

        </div>

        <div>

            <br>

            <button type="submit">Submit</button>

        </div>

    </div>

</form>

</body>

</html>

////////////////////

Show.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Employee Record</title>

    {% load static %}

    <link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body>

    <table border="1">

        <thead>

            <tr>

                <th>Employee ID</th>

                <th>Employee Name</th>

                <th>Employee Email</th>

                <th>Employee Contact</th>

                <th>Action</th>

            </tr>

        </thead>

        <tbody>

            {% for employee in employees %} <!--foreach employees hiya lista jaya mn fonction show-->

            <tr>

                <td>{{ employee.eid }}</td>  <!-- Affiche l'ID de l'employé -->

                <td>{{ employee.ename }}</td> <!-- Affiche le nom de l'employé -->

                <td>{{ employee.eemail }}</td> <!-- Affiche l'email de l'employé -->

                <td>{{ employee.econtact }}</td> <!-- Affiche le contact de l'employé -->

                <td>

                    <a href="/edit/{{ employee.id }}">Edit</a> <!-- Lien pour éditer l'employé avec son ID -->

                    <a href="/delete/{{ employee.id }}">Delete</a> <!-- Lien pour supprimer l'employé avec son ID -->

                </td>

            </tr>

        {% empty %}

            <tr>

                <td colspan="5">No records found</td> <!-- Message lorsque la liste est vide -->

            </tr>

        {% endfor %}

        </tbody>

    </table>

    <br><br>

    <button><a style="color:white;" href="/emp">Add New Record</a></button>

</body>

</html>