**“OFFICE EMPLOYEE MANAGEMENT SYSTEM(WEB APPLICATION)”**

**FIRST STEP(CREATE A PROJECT)**

* **CODE:-**

1. **Create a project cmd:- Django-admin startproject office\_emp\_proj.**
2. **Change the directory cmd:- cd office\_emp\_proj.**
3. **Create a app insde the create project cmd:- python manage.py startapp emp\_app**

**SECEOND STEP**

**Open “setting.py” in your project and past the code(at first remove all the code)**

"""

Django settings for office\_emp\_proj project.

Generated by 'django-admin startproject' using Django 3.1.6.

For more information on this file, see

https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.1/ref/settings/

"""

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'bux%-f7(%\*+3$hcm@tl&uchu+irr\*a9j)%-=nxy$s#@pd$sc4%'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'emp\_app',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'office\_emp\_proj.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'office\_emp\_proj.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

# Password validation

# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/3.1/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC\_URL = '/static/'

**In this project we can create two urls.py one is by default have in the project and another is created in app:-**

**URLS.PY(DY DEFAULT):-**

"""office\_emp\_proj URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

https://docs.djangoproject.com/en/3.1/topics/http/urls/

Examples:

Function views

1. Add an import: from my\_app import views

2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other\_app.views import Home

2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

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

path('',include('emp\_app.urls'))

]

URLS.PY(Which is created in the app by you):-

"""office\_emp\_proj URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

https://docs.djangoproject.com/en/3.1/topics/http/urls/

Examples:

Function views

1. Add an import: from my\_app import views

2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other\_app.views import Home

2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path, include

from . import views

urlpatterns = [

path('',views.index, name = 'index'),

path('war',views.war, name = 'warning'),

path('all\_emp',views.all\_emp, name = 'all\_emp'),

path('add\_emp',views.add\_emp, name = 'add\_emp'),

path('remove\_emp',views.remove\_emp, name = 'remove\_emp'),

path('remove\_emp/<int:emp\_id>',views.remove\_emp, name = 'remove\_emp'),

path('filter\_emp',views.filter\_emp, name = 'filter\_emp'),

]

NOW WE CAN CREATE A MODEL WHICH IS PLACED IN THE YOUR CREATED APP:-

from django.db import models

# Create your models here.

class Department(models.Model):

name = models.CharField(max\_length=100, null=False)

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

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

return self.name

class Role(models.Model):

name = models.CharField(max\_length=100, null=False)

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

return self.name

class Employee(models.Model):

first\_name = models.CharField(max\_length=100, null=False)

last\_name = models.CharField(max\_length=100, null=100)

dept = models.ForeignKey(Department, on\_delete=models.CASCADE)

salary = models.IntegerField(default=0)

bonus = models.IntegerField(default=0)

role = models.ForeignKey(Role, on\_delete=models.CASCADE)

phone = models.IntegerField(default=0)

hire\_date = models.DateField()

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

return "%s %s %s" %(self.first\_name, self.last\_name, self.phone)

AFTER YOU CRETED THE MODEL YOU CAN DO SOME NECCERY WORK LIKE MAKEMIGRATION, MIGRATE,AND CREATE A SUPERUSER, AFTER THAT WE CAN CREATE A “TEMPLATES” FILE INSIDE THE CREATED YOUR APP NOW WE CAN CREATE SOME HTML FILE,

ADD\_EMP.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">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.rtl.min.css" integrity="sha384-DOXMLfHhQkvFFp+rWTZwVlPVqdIhpDVYT9csOnHSgWQWPX0v5MCGtjCJbY6ERspU" crossorigin="anonymous">

<title>Add Employee</title>

</head>

<body>

<div class="container">

<h1>Add An Employee </h1>

<hr>

<form action="/add\_emp" method="post">

{% csrf\_token %}

<label for="first\_name" >First Name</label><br>

<input type="text" id="first\_name" name="first\_name" value="**{{** first\_name **}}**"><br>

<label for="last\_name" >Last Name</label><br>

<input type="text" id="last\_name" name="last\_name" value="**{{** last\_name **}}**"><br>

<label for="salary" >Salary</label><br>

<input type="text" id="salary" name="salary" value="**{{** salary **}}**"><br>

<label for="dept" >Department</label><br>

<input type="number" id="dept" name="dept" value="**{{** dept **}}**"><br>

<label for="role" >Role</label><br>

<input type="number" id="role" name="role" value="**{{** role **}}**"><br>

<label for="bonus" >Bonus</label><br>

<input type="number" id="bonus" name="bonus" value="**{{** bonus **}}**"><br>

<label for="phone" >Phone Number</label><br>

<input type="number" id="phone" name="phone" value="**{{** phone **}}**"><br>

<hr>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V" crossorigin="anonymous"></script>

-->

</body>

</html>

ALL\_EMP.HTML:-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.rtl.min.css" integrity="sha384-DOXMLfHhQkvFFp+rWTZwVlPVqdIhpDVYT9csOnHSgWQWPX0v5MCGtjCJbY6ERspU" crossorigin="anonymous">

<title>View All Employee</title>

</head>

<body>

<div class="container">

<h1>All Employee Details </h1>

<hr>

<table class="table table-dark table-striped">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">First Name</th>

<th scope="col">Last Name</th>

<th scope="col">Salary</th>

<th scope="col">Bonus</th>

<th scope="col">Phone Number</th>

<th scope="col">Roles</th>

<th scope="col">Department</th>

<th scope="col">Locations</th>

<th scope="col">Hiredates</th>

</tr>

</thead>

{% for emp in emps %}

<tbody>

<tr>

<th scope="row">**{{**emp.id**}}**</th>

<td>**{{**emp.first\_name**}}**</td>

<td>**{{**emp.last\_name**}}**</td>

<td>**{{**emp.salary**}}**</td>

<td>**{{**emp.bonus**}}**</td>

<td>**{{**emp.phone**}}**</td>

<td>**{{**emp.role.name**}}**</td>

<td>**{{**emp.dept.name**}}**</td>

<td>**{{**emp.dept.location**}}**</td>

<td>**{{**emp.hire\_date**}}**</td>

</tr>

</tbody>

{% endfor %}

</table>

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V" crossorigin="anonymous"></script>

-->

</body>

</html>

FILTER\_EMP.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">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.rtl.min.css" integrity="sha384-DOXMLfHhQkvFFp+rWTZwVlPVqdIhpDVYT9csOnHSgWQWPX0v5MCGtjCJbY6ERspU" crossorigin="anonymous">

<title>Filter Employee</title>

</head>

<body>

<div class="container">

<h1>Filter Employee Details</h1>

<hr>

<form action="/filter\_emp" method="post">

{% csrf\_token %}

<label for="first\_name" >First Name/Last Name</label><br>

<input type="text" id="name" name="name" value="**{{** name **}}**"><br>

<label for="dept" >Department</label><br>

<input type="text" id="dept" name="dept" value="**{{** dept **}}**"><br>

<label for="role" >Role</label><br>

<input type="text" id="role" name="role" value="**{{** role **}}**"><br>

<hr>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V" crossorigin="anonymous"></script>

-->

</body>

</html>

INDEX.HTML:-

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.rtl.min.css" integrity="sha384-DOXMLfHhQkvFFp+rWTZwVlPVqdIhpDVYT9csOnHSgWQWPX0v5MCGtjCJbY6ERspU" crossorigin="anonymous">

<title>Office Employee Management System</title>

</head>

<body>

<div class="container">

<h1>Office Employee Management System</h1>

<hr>

<a class="btn btn-primary" href="/all\_emp" role="button">View All Employee</a>

<a class="btn btn-secondary" href="/add\_emp" role="button">Add an Employee</a>

<a class="btn btn-success" href="/remove\_emp" role="button">Remove an Employee</a>

<a class="btn btn-success" href="/filter\_emp" role="button">Filter Employee Details</a>

<a class="btn btn-danger" href="/war" role="button">About</a>

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V" crossorigin="anonymous"></script>

-->

</body>

</html>

REMOVE\_EMP.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">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.rtl.min.css" integrity="sha384-DOXMLfHhQkvFFp+rWTZwVlPVqdIhpDVYT9csOnHSgWQWPX0v5MCGtjCJbY6ERspU" crossorigin="anonymous">

<title>Remove Employee</title>

</head>

<body>

<div class="container">

<h1>Remove An Employee </h1>

<hr>

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" data-bs-toggle="dropdown" aria-expanded="false">

Choose Employee To Be Removed

</button>

<ul class="dropdown-menu">

{% for emp in emps %}

<li><a class="dropdown-item" href="/remove\_emp/**{{**emp.id**}}**">**{{**emp.first\_name**}}** **{{**emp.last\_name**}}**</a></li>

{% endfor %}

</ul>

</div>

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V" crossorigin="anonymous"></script>

-->

</body>

</html>

WARNING.HTML:-

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.rtl.min.css" integrity="sha384-DOXMLfHhQkvFFp+rWTZwVlPVqdIhpDVYT9csOnHSgWQWPX0v5MCGtjCJbY6ERspU" crossorigin="anonymous">

<title>Office Employee Management System</title>

</head>

<body>

<div class="container">

<h1>Office Employee Management System</h1>

<hr>

<h1> What Is An Employee Management System? </h1>

<p>

An employee management system or EMS is a tool that helps improve employee satisfaction and productivity to help a company achieve their overall goals. These tools help monitor, assess and control employees' working hours and efficiently utilise human resources. It ensures that HR efficiently manages each employee's payroll and disburses salaries on time. An EMS securely stores and manages the personal and work-related details of employees. This makes it easier for the managers to store and access relevant data when needed.

</p>

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V" crossorigin="anonymous"></script>

-->

</body>

</html>

NOW WE CAN WRITE FUNCTION IN ABOUT VIEWS.PY

VIEWS.PY:-

from django.shortcuts import render, HttpResponse

from .models import Employee, Role, Department

from datetime import datetime

from django.db.models import Q

# Create your views here.

def index(request):

return render(request,'index.html')

def all\_emp(request):

emps = Employee.objects.all()

context = {

'emps': emps

}

print(context)

return render(request,'all\_emp.html', context)

def add\_emp(request):

if request.method == 'POST':

first\_name = request.POST['first\_name']

last\_name = request.POST['last\_name']

salary = int(request.POST['salary'])

bonus = int(request.POST['bonus'])

phone = int(request.POST['phone'])

dept = int(request.POST['dept'])

role = int(request.POST['role'])

new\_emp = Employee(first\_name=first\_name, last\_name=last\_name, salary=salary, bonus=bonus, phone=phone, dept\_id=dept, role\_id=role, hire\_date= datetime.now())

new\_emp.save()

return HttpResponse('Employee Added Successfully')

elif request.method=='GET':

return render(request,'add\_emp.html')

else:

return HttpResponse('an Exception Occures! Employee has been added')

def remove\_emp(request, emp\_id = 0):

if emp\_id:

try:

emp\_to\_be\_removed = Employee.objects.get(id=emp\_id)

emp\_to\_be\_removed.delete()

return HttpResponse("Employee Removed Successfully")

except:

return HttpResponse("Please Enter A Valid Employee ID")

emps = Employee.objects.all()

context = {

'emps': emps

}

return render(request,'remove\_emp.html', context)

def war(request):

return render(request,'warning.html')

def filter\_emp(request):

if request.method == 'POST':

name = request.POST['name']

dept = request.POST['dept']

role = request.POST['role']

emps = Employee.objects.all()

if name:

emps = emps.filter(Q(first\_name\_\_icontains = name) | Q(last\_name\_\_icontains = name))

if dept:

emps = emps.filter(dept\_\_name\_\_icontains = dept)

if role:

emps = emps.filter(role\_\_name\_\_icontains = role)

context = {

'emps': emps

}

return render(request, 'all\_emp.html', context)

elif request.method == 'GET':

return render(request,'filter\_emp.html')

else:

return HttpResponse('An Exception Occured')

**-:THANK YOU:-**