**WEEK 9**

Q1) **views.py**

from django.shortcuts import render, redirect

from .models import Category, Page

from .forms import CategoryForm, PageForm

def index(request):

categories = Category.objects.all()

return render(request, 'directory/index.html', {'categories': categories})

def category\_detail(request, category\_id):

category = Category.objects.get(id=category\_id)

pages = Page.objects.filter(category=category)

return render(request, 'directory/category\_detail.html', {'category': category, 'pages': pages})

def add\_category(request):

if request.method == "POST":

form = CategoryForm(request.POST)

if form.is\_valid():

form.save()

return redirect('index')

else:

form = CategoryForm()

return render(request, 'directory/add\_category.html', {'form': form})

def add\_page(request):

if request.method == "POST":

form = PageForm(request.POST)

if form.is\_valid():

form.save()

return redirect('index')

else:

form = PageForm()

return render(request, 'directory/add\_page.html', {'form': form})

**models.py**

from django.db import models

class Category(models.Model):

name = models.CharField(max\_length=100, unique=True)

visits = models.IntegerField(default=0)

likes = models.IntegerField(default=0)

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

return self.name

class Page(models.Model):

category = models.ForeignKey(Category, on\_delete=models.CASCADE)

title = models.CharField(max\_length=200)

url = models.URLField()

views = models.IntegerField(default=0)

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

return self.title

**forms.py**

from django import forms

from .models import Category, Page

class CategoryForm(forms.ModelForm):

class Meta:

model = Category

fields = ['name', 'visits', 'likes']

class PageForm(forms.ModelForm):

class Meta:

model = Page

fields = ['category', 'title', 'url', 'views']



Q2)

**views.py**from django.shortcuts import render, redirect

from .models import WORKS, LIVES

from .forms import WORKSForm

def index(request):

return render(request, 'employee/index.html')

def add\_employee(request):

if request.method == "POST":

form = WORKSForm(request.POST)

if form.is\_valid():

form.save()

return redirect('index')

else:

form = WORKSForm()

return render(request, 'employee/add\_employee.html', {'form': form})

def search\_employees(request):

employees = None

company\_name = ""

if request.method == "POST":

company\_name = request.POST.get('company\_name')

employees = WORKS.objects.filter(company\_name=company\_name).values('person\_name')

employees = list(employees)

for emp in employees:

lives = LIVES.objects.filter(person\_name=emp['person\_name']).values('city')

if lives:

emp['city'] = lives[0]['city']

else:

emp['city'] = "Unknown"

return render(request, 'employee/search\_employees.html', {'employees': employees, 'company\_name': company\_name})

**forms.py**from django import forms

from .models import WORKS

class WORKSForm(forms.ModelForm):

class Meta:

model = WORKS

fields = ['person\_name', 'company\_name', 'salary']

**models.py**

from django.db import models

class WORKS(models.Model):

person\_name = models.CharField(max\_length=100)

company\_name = models.CharField(max\_length=100)

salary = models.IntegerField()

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

return f"{self.person\_name} - {self.company\_name}"

class LIVES(models.Model):

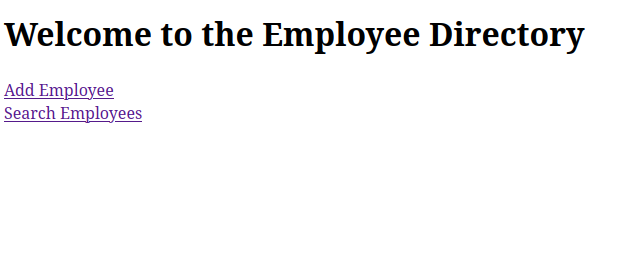
person\_name = models.CharField(max\_length=100)

street = models.CharField(max\_length=255)

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

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

return f"{self.person\_name} – {self.city}"



****

