WEEK10 – 220962446

**Question 1. Design a web site using Django, which is a website directory – A site containing**

**links to other websites. A web page has different categories.**

**• A category table has a name, number of visits, and number of likes.**

**• A page table refers to a category, has a title, URL, and many views.**

**Design a form that populates the above database and displays it.**

**Code 1)**

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, 'webapp/index.html', {'categories': categories})

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, 'webapp/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, 'webapp/add\_page.html', {'form': form})

urls.py

from django.urls import path

from . import views

urlpatterns = [

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

path('add\_category/', views.add\_category, name='add\_category'),

path('add\_page/', views.add\_page, name='add\_page'),

]

models.py

from django.db import models

class Category(models.Model):

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

num\_visits = models.IntegerField(default=0)

num\_likes = models.IntegerField(default=0)

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

return self.name

class Page(models.Model):

category = models.ForeignKey(Category, related\_name='pages', on\_delete=models.CASCADE)

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

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', 'num\_visits', 'num\_likes']

class PageForm(forms.ModelForm):

class Meta:

model = Page

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

base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Website Directory</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

}

header {

background-color: #007bff;

color: white;

padding: 15px;

text-align: center;

}

header h1 {

margin: 0;

}

nav ul {

list-style-type: none;

padding: 0;

}

nav ul li {

display: inline;

margin: 0 15px;

}

nav ul li a {

color: white;

text-decoration: none;

font-weight: bold;

}

nav ul li a:hover {

text-decoration: underline;

}

.container {

max-width: 1200px;

margin: 30px auto;

padding: 0 15px;

}

.category-card {

background-color: white;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

margin-bottom: 30px;

padding: 20px;

}

.category-card h3 {

margin-top: 0;

}

.category-card ul {

list-style-type: none;

padding: 0;

}

.category-card ul li {

margin: 5px 0;

}

.category-card ul li a {

color: #007bff;

text-decoration: none;

}

.category-card ul li a:hover {

text-decoration: underline;

}

footer {

background-color: #333;

color: white;

padding: 15px;

text-align: center;

}

.form-container {

background-color: white;

padding: 30px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

margin: 0 auto;

max-width: 600px;

}

.form-container form input, .form-container form textarea {

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ddd;

border-radius: 4px;

}

.form-container form button {

background-color: #007bff;

color: white;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

.form-container form button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<header>

<h1>Website Directory</h1>

<nav>

<ul>

<li><a href="{% url 'index' %}">Home</a></li>

<li><a href="{% url 'add\_category' %}">Add Category</a></li>

<li><a href="{% url 'add\_page' %}">Add Page</a></li>

</ul>

</nav>

</header>

<div class="container">

{% block content %}

{% endblock %}

</div>

<footer>

<p>&copy; 2025 Website Directory</p>

</footer>

</body>

</html>

add\_category.html

{% extends 'base.html' %}

{% block content %}

<h2>Add Category</h2>

<div class="form-container">

<form method="post">

{% csrf\_token %}

<div>

{{ form.as\_p }}

</div>

<button type="submit">Add Category</button>

</form>

</div>

{% endblock %}

add\_page.html

{% extends 'base.html' %}

{% block content %}

<h2>Add Page</h2>

<div class="form-container">

<form method="post">

{% csrf\_token %}

<div>

{{ form.as\_p }}

</div>

<button type="submit">Add Page</button>

</form>

</div>

{% endblock %}

index.html

{% extends 'base.html' %}

{% block content %}

<h2>Categories</h2>

<div>

{% for category in categories %}

<div class="category-card">

<h3>{{ category.name }}</h3>

<p><strong>Number of visits:</strong> {{ category.num\_visits }}</p>

<p><strong>Number of likes:</strong> {{ category.num\_likes }}</p>

<h4>Pages:</h4>

<ul>

{% for page in category.pages.all %}

<li>

<a href="{{ page.url }}" target="\_blank">{{ page.title }}</a> - Views: {{ page.views }}

</li>

{% endfor %}

</ul>

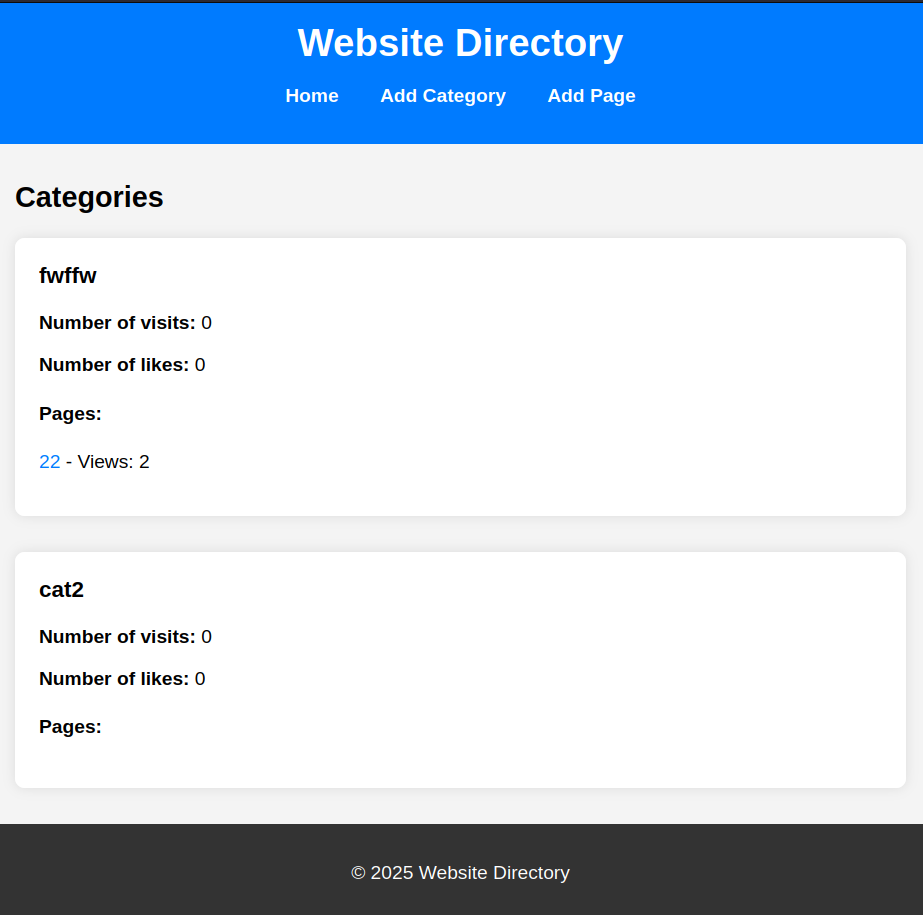
</div>

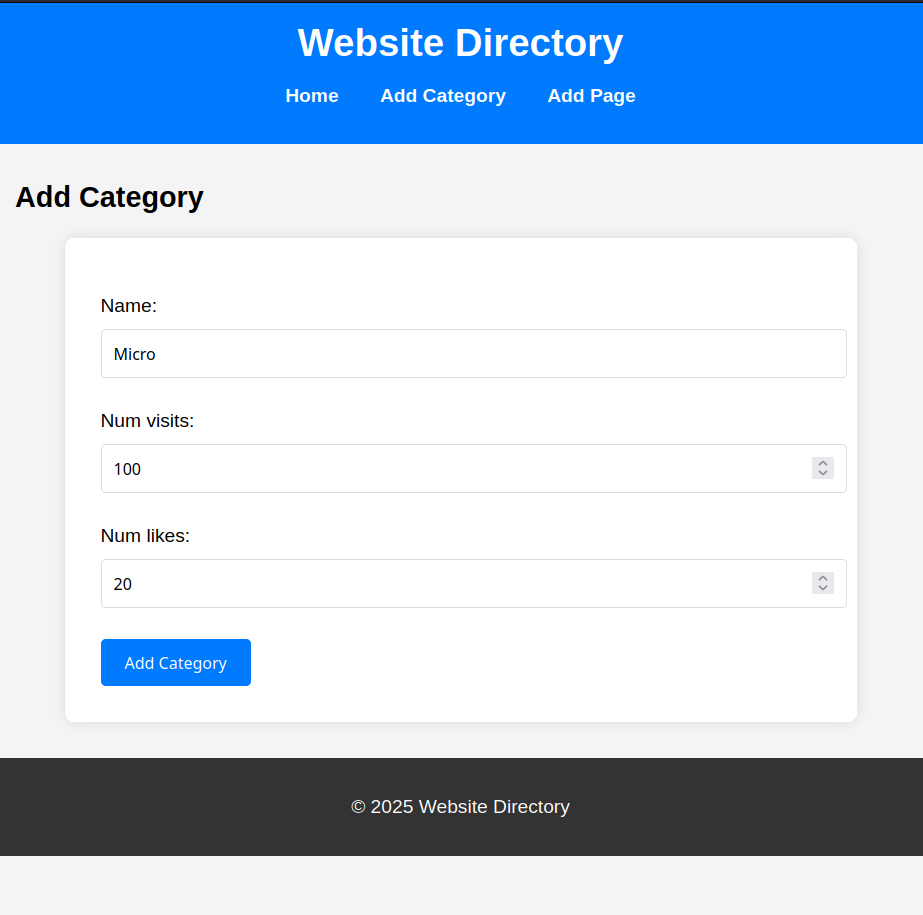
{% endfor %}

</div>

{% endblock %}

Solution 1)





**Question 2) Consider the following tables:**

**WORKS(person-name,Company-name,Salary)**

**LIVES(Person\_name, Street, City)**

**Assume Table data suitably. Design a Django webpage and include an option to**

**insert data into WORKS table by accepting data from the user using TextBoxes.**

**Also, include an option to retrieve the names of people who work for a**

**particular company along with the cities they live in (particular company name**

**must be accepted from the user)**

**Code 2)**

views.py

from django.shortcuts import render, redirect

from .models import Works, Lives

from .forms import WorksForm, CompanyQueryForm

def insert\_data(request):

if request.method == 'POST':

form = WorksForm(request.POST)

if form.is\_valid():

works\_entry = form.save()

lives\_entry = Lives(person\_name=works\_entry.person\_name, street="Default Street", city="Default City")

lives\_entry.save()

return redirect('insert\_data')

else:

form = WorksForm()

return render(request, 'webapp/insert\_data.html', {'form': form})

def company\_query(request):

people = []

if request.method == 'POST':

form = CompanyQueryForm(request.POST)

if form.is\_valid():

company\_name = form.cleaned\_data['company\_name']

print(f"Querying for company: {company\_name}")

works = Works.objects.filter(company\_name=company\_name)

print(f"Found {works.count()} records in Works for company {company\_name}")

for work in works:

print(f"Checking city for: {work.person\_name}")

# Get the city they live in

city = Lives.objects.filter(person\_name=work.person\_name).first()

if city:

print(f"Found city: {city.city} for {work.person\_name}")

people.append({

'person\_name': work.person\_name,

'city': city.city

})

else:

print(f"No city found for {work.person\_name}")

else:

form = CompanyQueryForm()

return render(request, 'webapp/company\_query.html', {'form': form, 'people': people})

urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.insert\_data, name='insert\_data'),

path('company\_query/', views.company\_query, name='company\_query'),

]

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.FloatField()

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

return f"{self.person\_name} works at {self.company\_name}"

class Lives(models.Model):

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

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

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

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

return f"{self.person\_name} lives in {self.city}"

forms.py

from django import forms

from .models import Works

class WorksForm(forms.ModelForm):

class Meta:

model = Works

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

widgets = {

'person\_name': forms.TextInput(attrs={'class': 'form-control'}),

'company\_name': forms.TextInput(attrs={'class': 'form-control'}),

'salary': forms.NumberInput(attrs={'class': 'form-control'}),

}

class CompanyQueryForm(forms.Form):

company\_name = forms.CharField(max\_length=100, widget=forms.TextInput(attrs={'class': 'form-control'}))

base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Company Data</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container">

<header class="py-3">

<h1>Company and City Information</h1>

<nav>

<ul class="nav">

<li class="nav-item"><a class="nav-link" href="{% url 'insert\_data' %}">Insert Data</a></li>

<li class="nav-item"><a class="nav-link" href="{% url 'company\_query' %}">Company Query</a></li>

</ul>

</nav>

</header>

{% block content %}

{% endblock %}

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

Solution 2)

