Q1: 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:

webapp/forms.py:

from django import forms

from .models import Category, Page

class PageForm(forms.ModelForm):

class Meta:

model = Page

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

class CategoryForm(forms.ModelForm):

class Meta:

model = Category

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

webapp/views.py:

from django.shortcuts import render, redirect

from .forms import CategoryForm, PageForm

from .models import Category

def index(request):

if request.method == 'POST':

if 'category\_submit' in request.POST:

category\_form = CategoryForm(request.POST)

if category\_form.is\_valid():

category\_form.save()

else:

page\_form = PageForm(request.POST)

if page\_form.is\_valid():

page\_form.save()

return redirect('index')

categories = Category.objects.all()

return render(request, 'webapp/index.html', {

'categories': categories,

'category\_form': CategoryForm(),

'page\_form': PageForm()

})

webapp/models.py:

from django.db import models

class Category(models.Model):

name = models.CharField(max\_length=255, unique=True)

visits = models.PositiveIntegerField(default=0)

likes = models.PositiveIntegerField(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=255)

url = models.URLField()

views = models.PositiveIntegerField(default=0)

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

return self.title

webapp/templates/webapp/index.html:

<!DOCTYPE html>

<html>

<head>

<title>Website Directory</title>

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

</head>

<body class="bg-light">

<div class="container mt-4">

<div class="row">

<div class="col-md-6">

<div class="card mb-4">

<div class="card-header bg-primary text-white">

<h1 class="card-title h4 mb-0">Add Category</h1>

</div>

<div class="card-body">

<form method="post">

{% csrf\_token %}

<div class="mb-3">

{{ category\_form.as\_p }}

</div>

<button type="submit" name="category\_submit" class="btn btn-primary">Add Category</button>

</form>

</div>

</div>

<div class="card mb-4">

<div class="card-header bg-success text-white">

<h1 class="card-title h4 mb-0">Add Page</h1>

</div>

<div class="card-body">

<form method="post">

{% csrf\_token %}

<div class="mb-3">

{{ page\_form.as\_p }}

</div>

<button type="submit" class="btn btn-success">Add Page</button>

</form>

</div>

</div>

</div>

<div class="col-md-6">

<div class="card">

<div class="card-header bg-dark text-white">

<h1 class="card-title h4 mb-0">Directory</h1>

</div>

<div class="card-body">

{% for category in categories %}

<div class="card mb-3">

<div class="card-header">

<h2 class="h5 mb-0">{{ category.name }}</h2>

</div>

<div class="card-body">

<p class="text-muted small mb-2">Visits: {{ category.visits }} | Likes: {{ category.likes }}</p>

<ul class="list-group">

{% for page in category.page\_set.all %}

<li class="list-group-item d-flex justify-content-between align-items-center">

<a href="{{ page.url }}" class="btn btn-link btn-sm text-start">

{{ page.title }}

</a>

<span class="badge bg-secondary">Views: {{ page.views }}</span>

</li>

{% endfor %}

</ul>

</div>

</div>

{% endfor %}

</div>

</div>

</div>

</div>

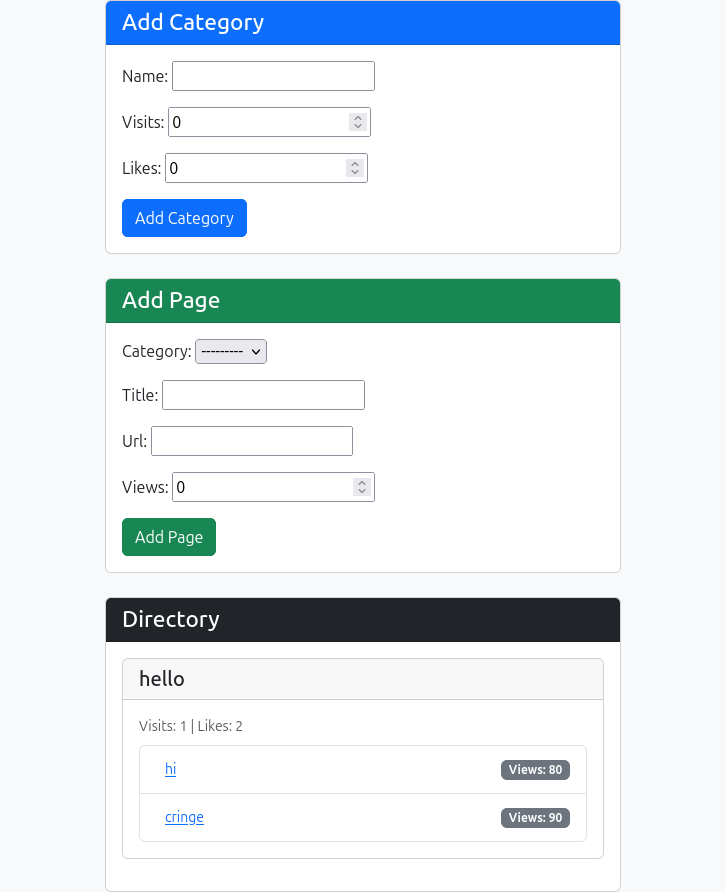
</div>

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

</body>

</html>

Image:



Q2: 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:

webapp/forms.py:

from django import forms

from .models import Works

class WorksForm(forms.ModelForm):

class Meta:

model = Works

fields = '\_\_all\_\_'

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 CompanySearchForm(forms.Form):

company\_name = forms.CharField(

label='Search by Company',

widget=forms.TextInput(attrs={'class': 'form-control'})

)

webapp/views.py:

from django.shortcuts import render

from .forms import WorksForm, CompanySearchForm

from .models import Works, Lives

from django.contrib import messages

def index(request):

if request.method == 'POST':

if 'submit\_works' in request.POST:

works\_form = WorksForm(request.POST)

if works\_form.is\_valid():

works\_form.save()

messages.success(request, 'Data inserted successfully!')

elif 'submit\_search' in request.POST:

search\_form = CompanySearchForm(request.POST)

if search\_form.is\_valid():

company = search\_form.cleaned\_data['company\_name']

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

results = []

for emp in employees:

try:

lives = Lives.objects.get(person\_name=emp.person\_name)

city = lives.city

except Lives.DoesNotExist:

city = 'Unknown'

results.append({

'name': emp.person\_name,

'company': emp.company\_name,

'salary': emp.salary,

'city': city

})

return render(request, 'webapp/index.html', {

'works\_form': WorksForm(),

'search\_form': CompanySearchForm(),

'results': results,

'searched\_company': company

})

return render(request, 'webapp/index.html', {

'works\_form': WorksForm(),

'search\_form': CompanySearchForm()

})

webapp/models.py:

from django.db import models

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 self.person\_name

class Works(models.Model):

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

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

salary = models.DecimalField(max\_digits=10, decimal\_places=2)

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

return self.person\_name

webapp/templates/webapp/index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Employee Locator</title>

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

</head>

<body class="container mt-4">

<h2 class="mb-4">Employee Management</h2>

<div class="card mb-4">

<div class="card-header bg-primary text-white">

Insert Employee Work Details

</div>

<div class="card-body">

<form method="post">

{% csrf\_token %}

{{ works\_form.as\_p }}

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

</form>

</div>

</div>

<div class="card mb-4">

<div class="card-header bg-success text-white">

Search Employees by Company

</div>

<div class="card-body">

<form method="post">

{% csrf\_token %}

{{ search\_form.as\_p }}

<button type="submit" name="submit\_search" class="btn btn-success">Search</button>

</form>

</div>

</div>

{% if messages %}

{% for message in messages %}

<div class="alert alert-{{ message.tags }}">{{ message }}</div>

{% endfor %}

{% endif %}

{% if results %}

<div class="card">

<div class="card-header bg-info text-white">

Results for company: {{ searched\_company }}

</div>

<div class="card-body">

<table class="table table-striped">

<thead>

<tr>

<th>Name</th>

<th>Company</th>

<th>Salary</th>

<th>City</th>

</tr>

</thead>

<tbody>

{% for emp in results %}

<tr>

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

<td>{{ emp.company }}</td>

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

<td>{{ emp.city }}</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

{% endif %}

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

</body>

</html>

Image:

