NAME:AYANIKA PAUL
REG NO. 220905128 LAB 9

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:

q1/urls.py

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
path('admin/', admin.site.urls),
path('', include('website.urls')),
]
```

website/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'),
]
```

website/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']
```

website/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, 'website/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, 'website/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, 'website/add_page.html', {'form': form})
```

website/models.py

```
from django.db import models

class Category(models.Model):
name = models.CharField(max_length=200)
num_visits = models.PositiveIntegerField(default=0)
num_likes = models.PositiveIntegerField(default=0)
def __str__(self):
return self.name
class Page(models.Model):
category = models.ForeignKey(Category, on_delete=models.CASCADE, related_name='pages')
title = models.CharField(max_length=200)
url = models.URLField()
views = models.PositiveIntegerField(default=0)
def __str__(self):
return self.title
```

add_category.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Add Category</title>
</head>
```

```
<body>
<h1>Add a New Category</h1>
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Save</button>
</form>
<a href="{% url 'index' %}">Back to Directory</a>
</body>
</html>
```

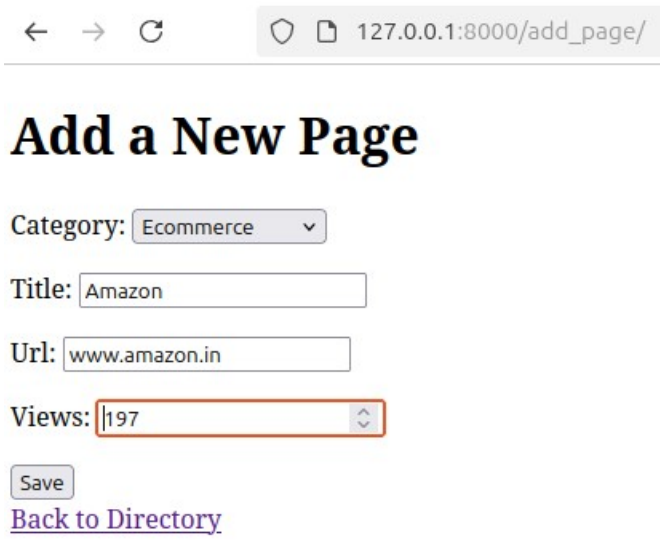
add_page.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Add Page</title>
</head>
<body>
<h1>Add a New Page</h1>
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Save</button>
</form>
<a href="{% url 'index' %}">Back to Directory</a>
</body>
</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Website Directory</title>
</head>
<body>
<h1>Website Directory</h1>
<h2>Categories</h2>
<ul>
{% for category in categories %}
<li>
<h3>{{ category.name }}</h3>
<p>Number of Visits: {{ category.num_visits }}</p>
<p>Number of Likes: {{ category.num_likes }}</p>
```

```
<h4>Pages:</h4>
<ul>
{% for page in category.pages.all %}
<li>
<a href="{{ page.url }}" target="_blank">{{ page.title }}</a>
<p>Views: {{ page.views }}</p>
</li>
{% endfor %}
</ul>
</li>
{% endfor %}
</ul>
<a href="{% url 'add_category' %}">Add Category</a><br>
<a href="{% url 'add_page' %}">Add Page</a>
</body>
</html>
```



127.0.0.1:8000

Website Directory

Categories

- **Ecommerce**

  Number of Visits: 500

  Number of Likes: 400

  Pages:

  - Amazon

    Views: 197

127.0.0.1:8000/add_category/

Add a New Category

Name: Tickets

Num visits: 800

Num likes: 500

Save

Back to Directory

127.0.0.1:8000

Website Directory

Categories

- **Ecommerce**

  Number of Visits: 500

  Number of Likes: 400

  Pages:

  - Amazon

    Views: 197

- **Pharmaceutical**

  Number of Visits: 300

  Number of Likes: 159

  Pages:

  - Pharmeasy

    Views: 270

- **Tickets**

  Number of Visits: 800

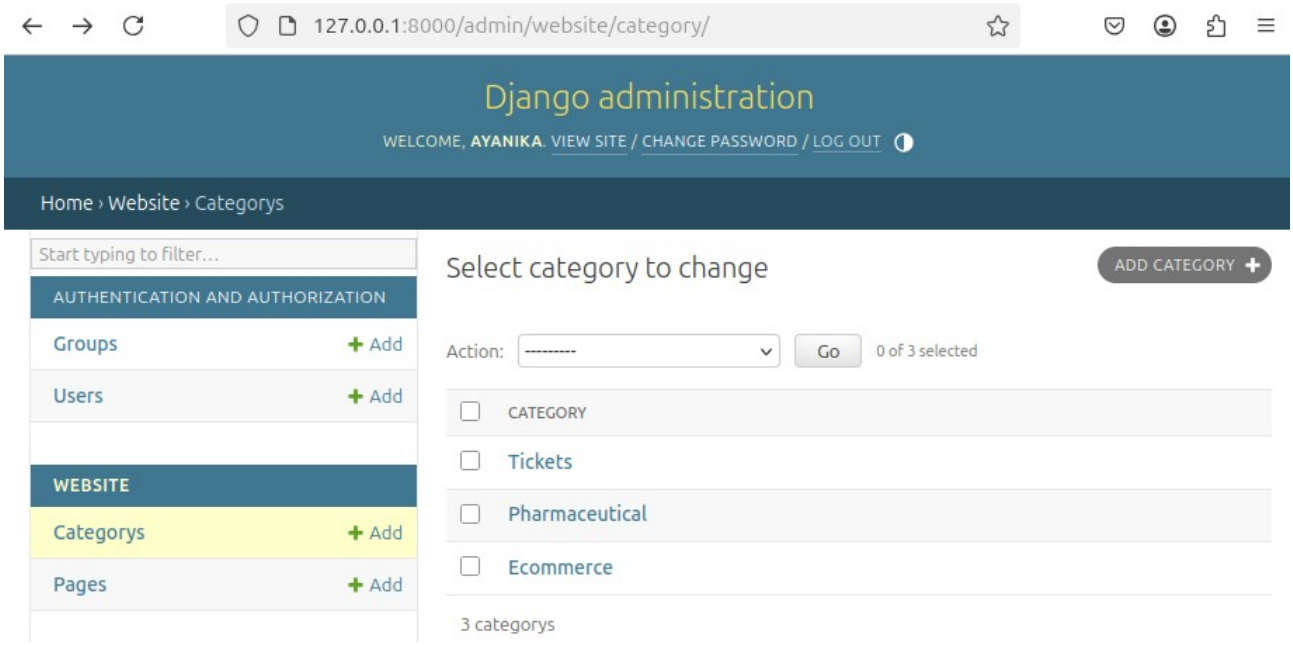
  Number of Likes: 500

  Pages:

  - MakyMyTrip

    Views: 800

Add Category
Add Page



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:

q2/urls.py

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
path('admin/', admin.site.urls),
path('', include('webapp.urls')),
]
```

webapp/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
path('', views.index, name='index'),
path('add/', views.add_work, name='add_work'),
path('add_lives/', views.add_lives, name='add_lives'),
```

```
path('search/', views.search_people, name='search_people'),
]
```

website/forms,py

```
from django import forms
from .models import Works, Lives
class WorksForm(forms.ModelForm):
class Meta:
model = Works
fields = ['person_name', 'company_name', 'salary']
class LivesForm(forms.ModelForm):
class Meta:
model = Lives
fields = ['person_name', 'street', 'city']
```

webapp/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.DecimalField(max_digits=10, decimal_places=2)
def __str__(self):
return f"{self.person_name} works at {self.company_name}"
class Lives(models.Model):
person_name = models.ForeignKey(Works, on_delete=models.CASCADE)
street = models.CharField(max_length=100)
city = models.CharField(max_length=100)
def __str__(self):
return f"{self.person_name.person_name} lives in {self.city}"
```

add_lives.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Add Lives Details</title>
</head>
<body>
<h1>Add Lives Details</h1>
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Save</button>
</form>
</body>
</html>
```

add_success.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Success</title>
</head>
<body>
<h1>Data Saved Successfully!</h1>
<p><a href="{% url 'index' %}">Go back to Home</a></p>
</body>
</html>
```

add_work.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Add Work Details</title>
</head>
<body>
<h1>Add Work Details</h1>
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Save</button>
</form>
</body>
</html>
```

index.html

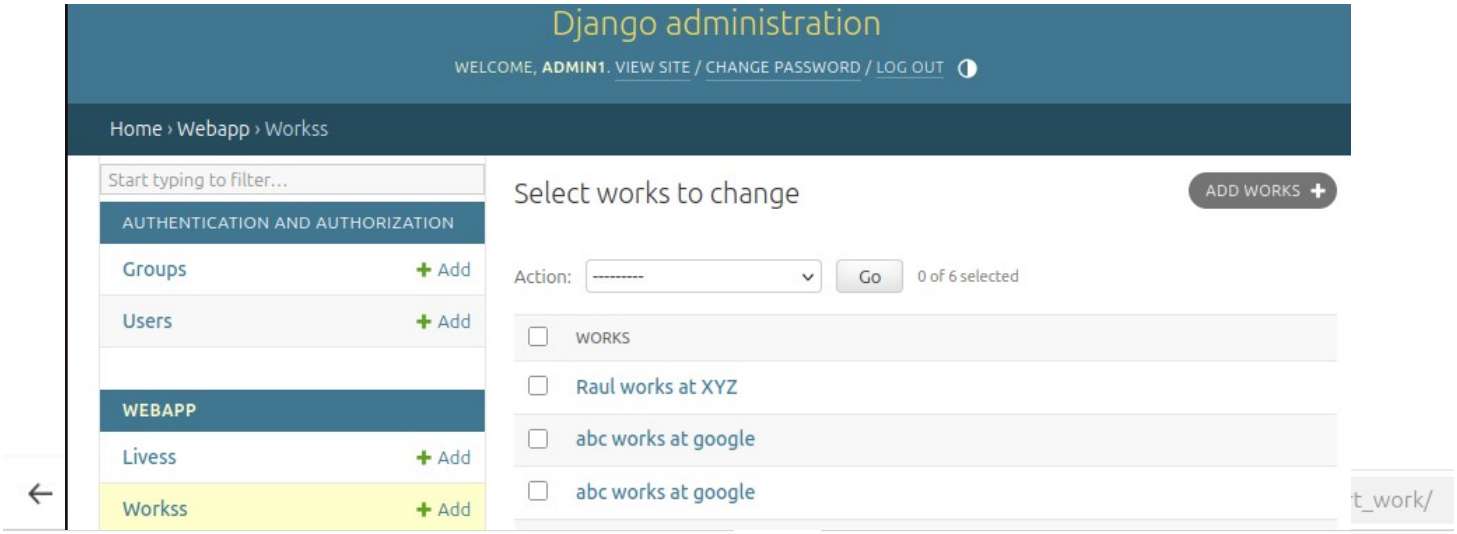
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Home</title>
</head>
<body>
<h1>Welcome to the Employee Directory</h1>
<p><a href="{% url 'add_work' %}">Add Work Details</a></p>
<p><a href="{% url 'add_lives' %}">Add Lives Details</a></p>
<p><a href="{% url 'search_people' %}">Search People by Company</a></p>
</body>
</html>
```

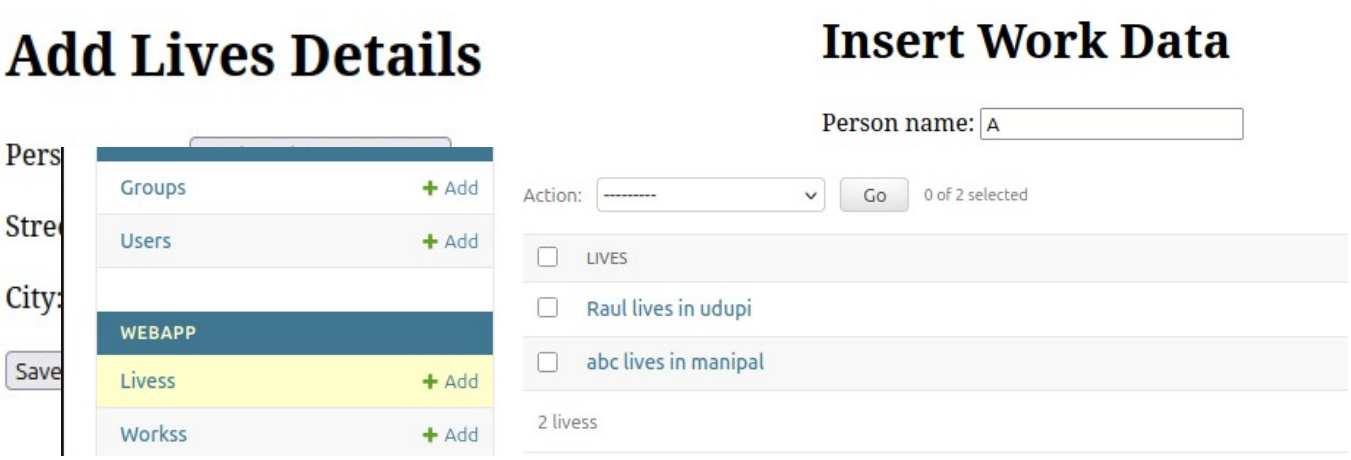
search_people.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Search People</title>
</head>
<body>
<h1>Search People by Company</h1>
<form method="post">
{% csrf_token %}
<label for="company_name">Company Name:</label>
<input type="text" name="company_name" required>
<button type="submit">Search</button>
</form>
</body>
</html>
```

search_people.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Search Results</title>
</head>
<body>
<h1>Search Results</h1>
{% if data %}
<ul>
{% for item in data %}
<li>{{ item.name }} - {{ item.city }}</li>
{% endfor %}
</ul>
{% else %}
<p>No results found.</p>
{% endif %}
</body>
</html>
```


Django administration
WELCOME, ADMIN1. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Webapp › Workss
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
WEBAPP
Livess + Add
Workss + Add
Select works to change
ADD WORKS +
Action: --------- Go 0 of 6 selected
WORKS
Raul works at XYZ
abc works at google
abc works at google


Add Lives Details
Groups + Add
Users + Add
WEBAPP
Livess + Add
Workss + Add
Insert Work Data
Person name: A
Action: --------- Go 0 of 2 selected
LIVES
Raul lives in udupi
abc lives in manipal
2 livess


Search People by Company
Company Name: google
Search


Search Results
abc - manipal