Lab 10 – Web Programming Lab

Arhaan Girdhar – 220962050

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.

Views.py

```
from django.shortcuts import render
from django.shortcuts import render, redirect
from .forms import InsertWorksForm, SearchForm
from .models import Works, Lives
def insert_works(request):
    if request.method == 'POST':
        form = InsertWorksForm(request.POST)
    if form.is_valid():
        form.save()
        return redirect('insert_works')
    else:
        form = InsertWorksForm()
        return render(request, 'employee/insert_works.html', {'form': form})
def search_people(request):
    results = []
    if request.method == 'POST':
        form = SearchForm(request.POST)
        if form.is_valid():
        company = form.cleaned_data['company_name']
        works_qs = Works.objects.filter(company_name=company)
        for work in works_qs:
```

```
        try:
                lives = Lives.objects.get(person_name=work.person_name)
                results.append({'person_name': work.person_name, 'city': lives.city})
                except Lives.DoesNotExist:
                results.append({'person_name': work.person_name, 'city': 'Unknown'})
        else:
                form = SearchForm()
                return render(request, 'employee/search.html', {'form': form, 'results': results})
```

Urls.py

```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
path('admin/', admin.site.urls),
path('', include('directory.urls')),
]
```

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']
```

Settings.py

```
import os

from pathlib import Path


BASE_DIR = Path(__file__).resolve().parent.parent


SECRET_KEY = 'replace-this-with-a-secure-key'

DEBUG = True

ALLOWED_HOSTS = []


INSTALLED_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'directory',

]


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 = 'websiteDIR.urls'


TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [], # You can add global template directories here.
'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 = 'websiteDIR.wsgi.application'
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'


DATABASES = {
'default': {
'ENGINE': 'django.db.backends.sqlite3',
'NAME': BASE_DIR / 'db.sqlite3',
}
}


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'},
]

LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True
STATIC_URL = '/static/'
```

Models.py

```
from django.db import models

class Category(models.Model):
name = models.CharField(max_length=128, 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, related_name='pages')
title = models.CharField(max_length=128)
url = models.URLField()
views = models.IntegerField(default=0)

def __str__(self):
return self.title
```

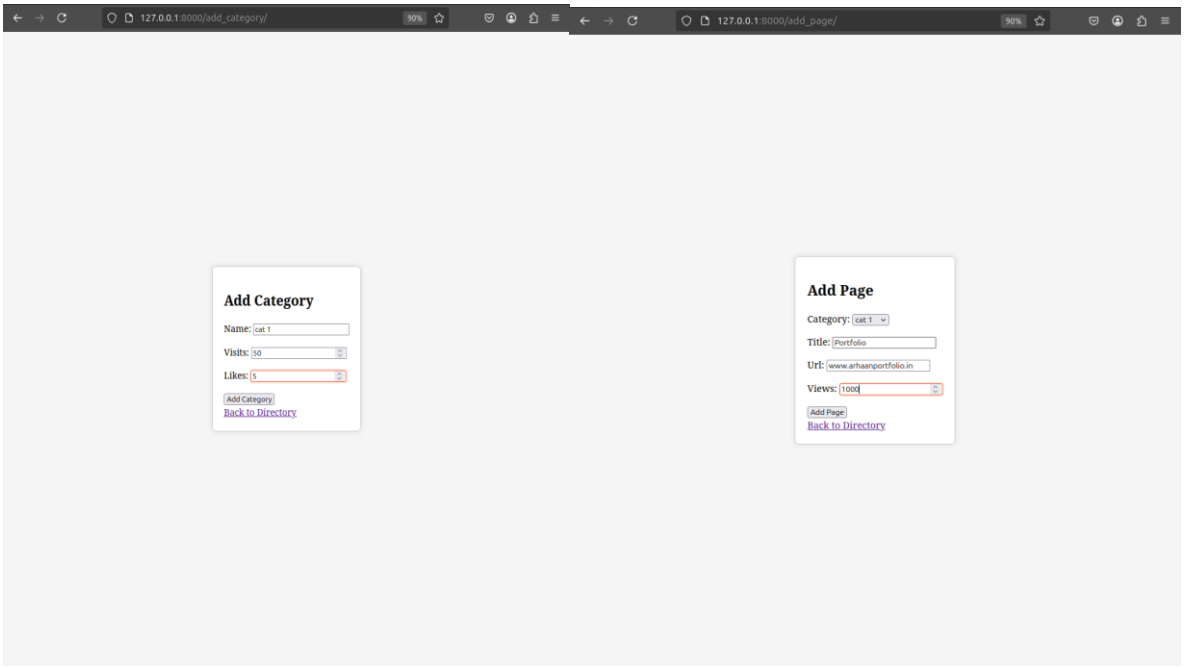
Index.html

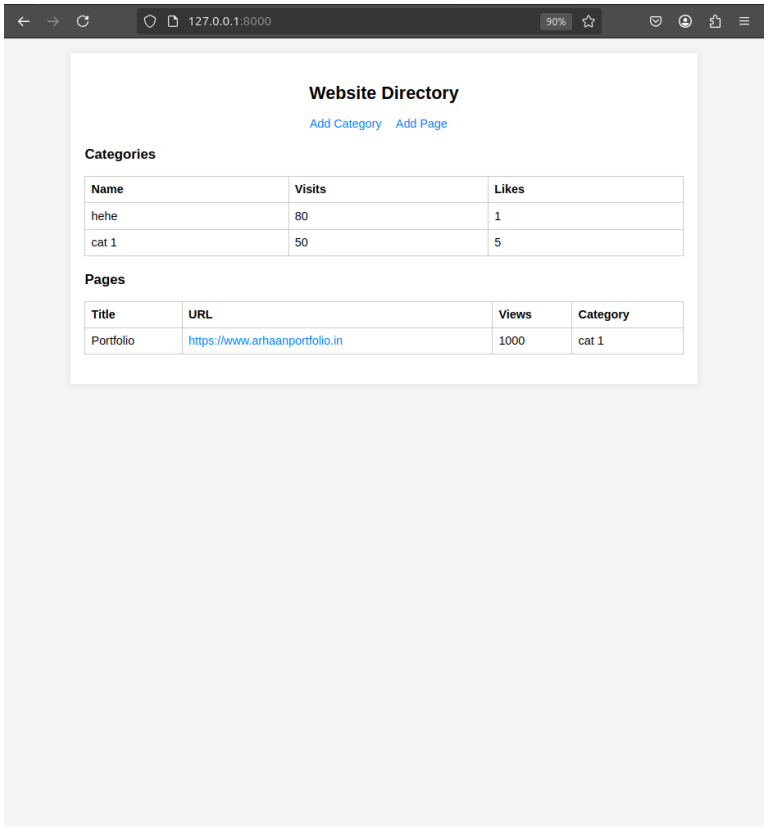
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Website Directory</title>
<style>
body { font-family: Arial, sans-serif; background: #f5f5f5; }
.container {
width: 80%; margin: 20px auto; background: #fff; padding: 20px;
box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
h2 { text-align: center; }
table { width: 100%; border-collapse: collapse; margin-bottom: 20px; }
table, th, td { border: 1px solid #ccc; }
th, td { padding: 8px; text-align: left; }
a { margin-right: 15px; text-decoration: none; color: #007BFF; }
a:hover { text-decoration: underline; }
</style>
</head>
<body>
<div class="container">
<h2>Website Directory</h2>
<div style="text-align: center; margin-bottom: 20px;">
<a href="{% url 'add_category' %}">Add Category</a>
<a href="{% url 'add_page' %}">Add Page</a>
</div>
<h3>Categories</h3>
<table>
```

```
<tr>
<th>Name</th>
<th>Visits</th>
<th>Likes</th>
</tr>
{% for category in categories %}
<tr>
<td>{{ category.name }}</td>
<td>{{ category.visits }}</td>
<td>{{ category.likes }}</td>
</tr>
{% empty %}
<tr>
<td colspan="3" style="text-align:center;">No categories added yet.</td>
</tr>
{% endfor %}
</table>
<h3>Pages</h3>
<table>
<tr>
<th>Title</th>
<th>URL</th>
<th>Views</th>
<th>Category</th>
</tr>
{% for page in pages %}
<tr>
<td>{{ page.title }}</td>
<td><a href="{{ page.url }}" target="_blank">{{ page.url }}</a></td>
```

```
<td>{{ page.views }}</td>

<td>{{ page.category.name }}</td>

</tr>

{% empty %}

<tr>

<td colspan="4" style="text-align:center;">No pages added yet.</td>

</tr>

{% endfor %}

</table>

</div>

</body>

</html>
```

OUTPUT





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).

Views.py

```
from django.shortcuts import render
from django.shortcuts import render, redirect
from .forms import InsertWorksForm, SearchForm
from .models import Works, Lives
def insert_works(request):
    if request.method == 'POST':
        form = InsertWorksForm(request.POST)
    if form.is_valid():
        form.save()
        return redirect('insert_works')
    else:
        form = InsertWorksForm()
        return render(request, 'employee/insert_works.html', {'form': form})
def search_people(request):
    results = []
    if request.method == 'POST':
        form = SearchForm(request.POST)
        if form.is_valid():
        company = form.cleaned_data['company_name']
        works_qs = Works.objects.filter(company_name=company)
        for work in works_qs:
    try:
```

```
                lives = Lives.objects.get(person_name=work.person_name)
                results.append({'person_name': work.person_name, 'city': lives.city})
                except Lives.DoesNotExist:
                results.append({'person_name': work.person_name, 'city': 'Unknown'})
        else:
                form = SearchForm()
                return render(request, 'employee/search.html', {'form': form, 'results': results})
```

Urls.py

```
from django.urls import path
from .views import insert_works, search_people

urlpatterns = [
path('insert/', insert_works, name='insert_works'),
path('search/', search_people, name='search_people'),
]
```

Forms.py

```
from django import forms
from .models import Works

class InsertWorksForm(forms.ModelForm):
        class Meta:
                model = Works
                fields = ['person_name', 'company_name', 'salary']

class SearchForm(forms.Form):
        company_name = forms.CharField(max_length=100, label="Company Name")
```

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 self.person_name


class Lives(models.Model):
        person_name = models.CharField(max_length=100)
        street = models.CharField(max_length=200)
        city = models.CharField(max_length=100)
        def __str__(self):
                return self.person_name
```

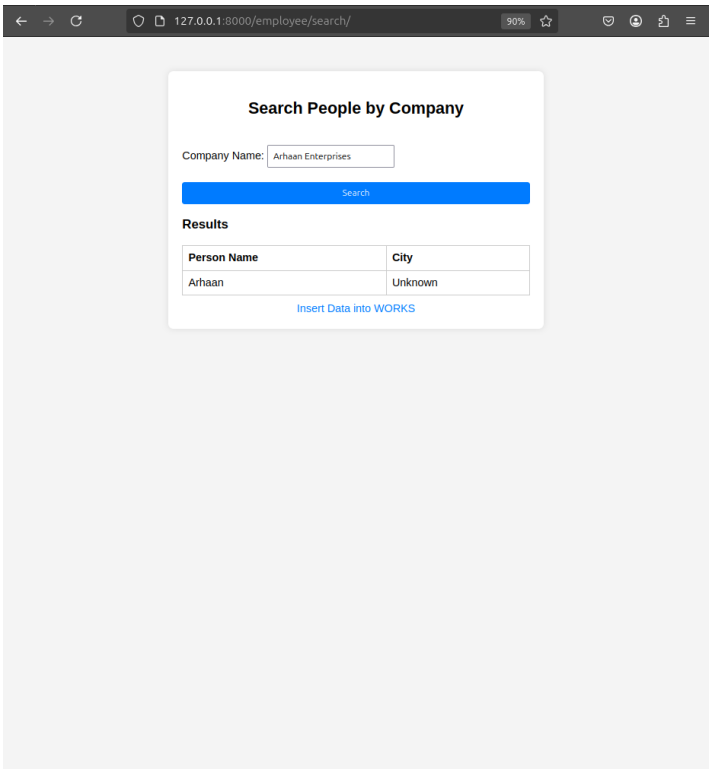
Index.html

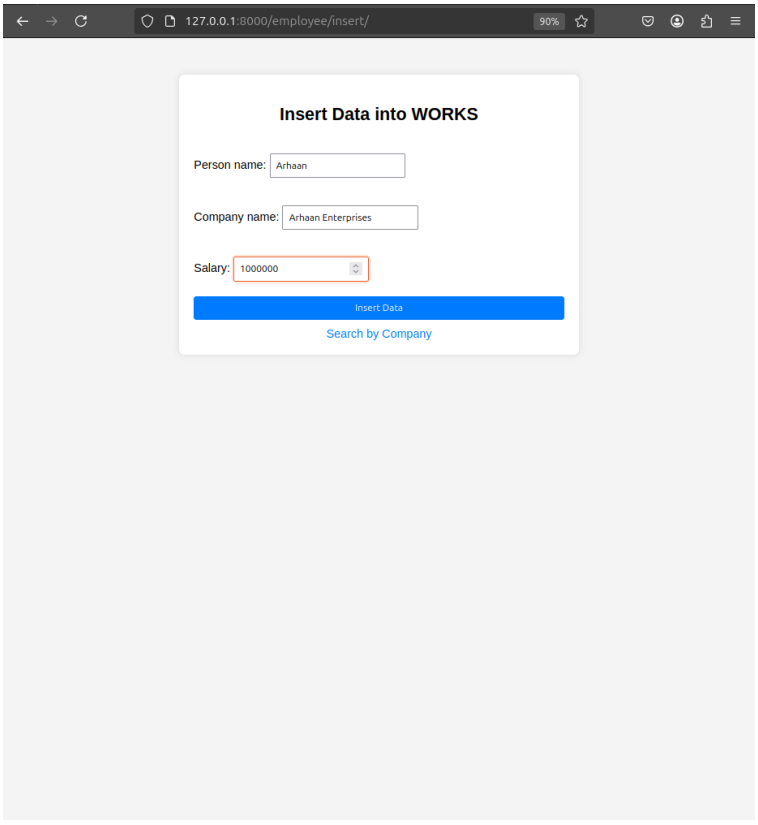
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Search People by Company</title>
<style>
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.container { width: 50%; margin: 50px auto; padding: 20px; background: #fff; border-radius: 8px; box-shadow: 0 0 10px rgba(0,0,0,0.1); }
h2 { text-align: center; }
form { display: flex; flex-direction: column; }
label { margin-top: 10px; }
input, button { padding: 8px; margin-top: 5px; }
button { background: #007BFF; color: #fff; border: none; border-radius: 4px; cursor: pointer; }
```

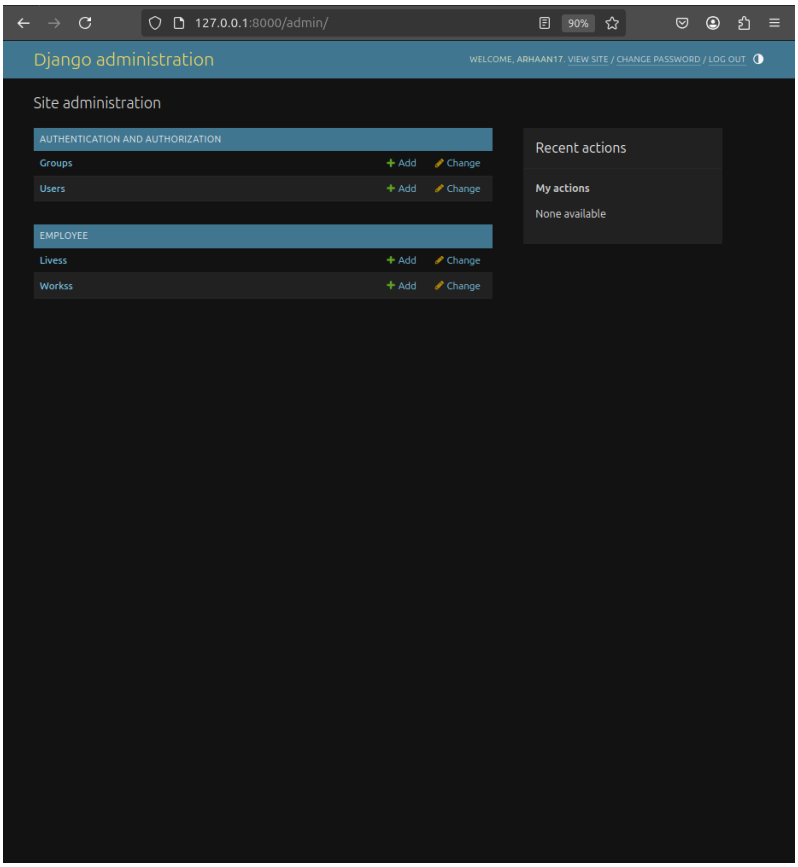
```
button:hover { background: #0056b3; }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { border: 1px solid #ccc; padding: 8px; text-align: left; }
a { margin-top: 10px; text-align: center; display: block; color: #007BFF; text-decoration: none; }
a:hover { text-decoration: underline; }
</style>
</head>
<body>
<div class="container">
<h2>Search People by Company</h2>
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Search</button>
</form>
{% if results %}
<h3>Results</h3>
<table>
<tr>
<th>Person Name</th>
<th>City</th>
</tr>
{% for result in results %}
<tr>
<td>{{ result.person_name }}</td>
<td>{{ result.city }}</td>
</tr>
{% endfor %}
</table>
```

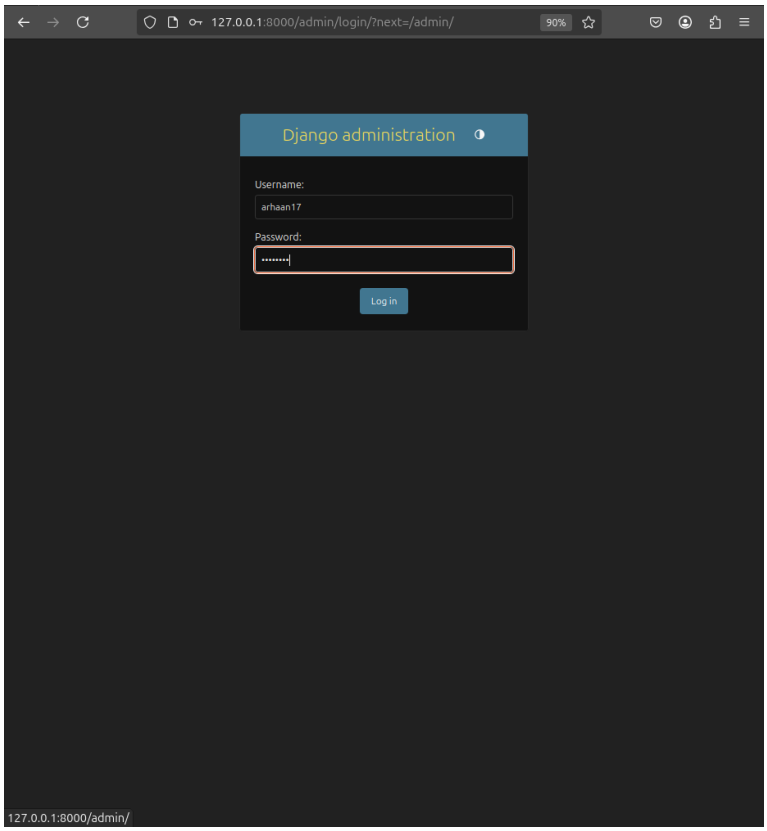
```
{% endif %}

<a href="{% url 'insert_works' %}">Insert Data into WORKS</a>

</div>

</body>

</html>
```

OUTPUT


127.0.0.1:8000/employee/search/
90%
Search People by Company
Company Name:
Arhaan Enterprises
Search
Results
Person Name
City
Arhaan
Unknown
Insert Data into WORKS


127.0.0.1:8000/employee/insert/
90%
Insert Data into WORKS
Person name:
Arhaan
Company name:
Arhaan Enterprises
Salary:
1000000
Insert Data
Search by Company


127.0.0.1:8000/admin/
90%
Django administration
WELCOME, ARHAAN17. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
Users + Add Change
EMPLOYEE
Livess + Add Change
Workss + Add Change
Recent actions
My actions
None available


127.0.0.1:8000/admin/login/?next=/admin/
90%
Django administration
Username:
arhaan17
Password:
Log in
127.0.0.1:8000/admin/