==========
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.
==========
Views.py:

```
from django.shortcuts import render, redirect
from .forms import CategoryForm, PageForm
from .models import Category, Page

def index(request):
    return render(request, 'directory/index.html')

def add_category(request):
    if request.method == 'POST':
        form = CategoryForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('list_categories')
    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('list_pages')
    else:
        form = PageForm()
    return render(request, 'directory/add_page.html', {'form': form})

def list_categories(request):
    categories = Category.objects.all()
    return render(request, 'directory/list_categories.html', {'categories': categories})

def list_pages(request):
    pages = Page.objects.all()
    return render(request, 'directory/list_pages.html', {'pages': pages})
```

==========
Models.py:

```
from django.db import models

# Create your models here.
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
```

==========

list_categories.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Categories List</title>
</head>
<body>
  <h1>Categories</h1>
  <ul>
    {% for category in categories %}
      <li>{{ category.name }} - Visits: {{ category.visits }}, Likes: {{ category.likes }}</li>
    {% empty %}
      <li>No categories yet.</li>
    {% endfor %}
  </ul>
  <a href="{% url 'add_category' %}">Add a New Category</a>
</body>
</html>
```

==========

list_pages.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Pages List</title>
</head>
<body>
  <h1>Pages</h1>
  <ul>
    {% for page in pages %}
      <li>
        <strong>{{ page.title }}</strong> ({{ page.views }} views) -
        Category: {{ page.category.name }} -
        <a href="{{ page.url }}" target="_blank">{{ page.url }}</a>
      </li>
    {% empty %}
      <li>No pages yet.</li>
    {% endfor %}
  </ul>
```

```
    <a href="{% url 'add_page' %}">Add a New Page</a>
</body>
</html>
```

==========

add_pages.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Add Page</title>
</head>
<body>
    <h1>Add a New Page</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Page</button>
    </form>
    <a href="{% url 'list_pages' %}">Back to Pages</a>
</body>
</html>
```

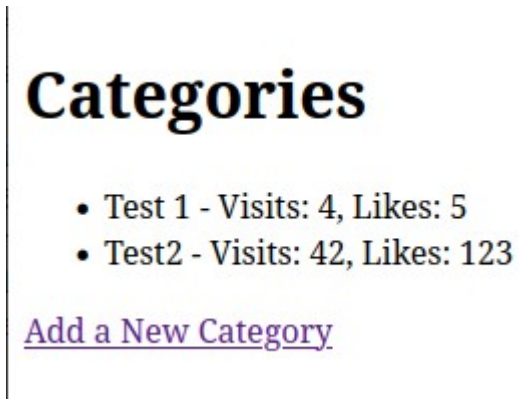
==========

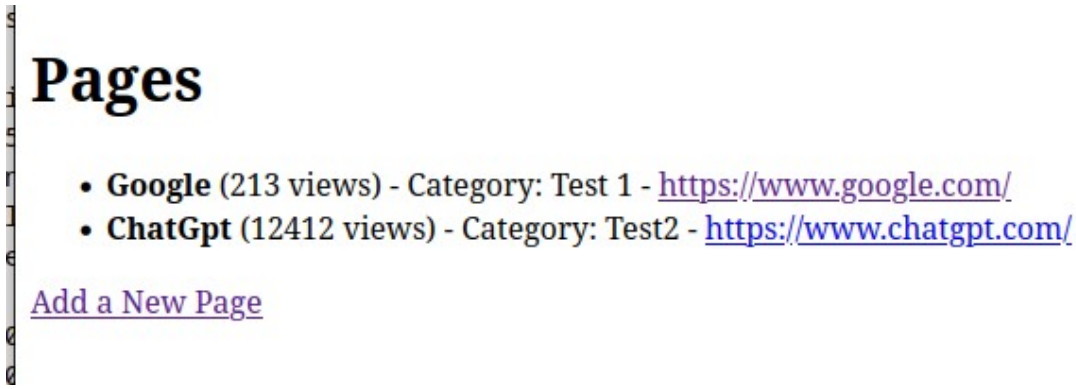
add_category.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Add Category</title>
</head>
<body>
    <h1>Add a New Category</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Category</button>
    </form>
    <a href="{% url 'list_categories' %}">Back to Categories</a>
</body>
</html>
```

==========

Output:





==========

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).

==========

Views.py:

```
from django.shortcuts import render, redirect
from .forms import WorksForm, CompanyQueryForm
from .models import Works, Lives

def index(request):
    return render(request, 'directory/index.html')

# View to insert a record into WORKS
def insert_works(request):
    if request.method == 'POST':
        form = WorksForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('insert_works')  # Redirect to the same page (or elsewhere)
    else:
        form = WorksForm()
    return render(request, 'directory/insert_works.html', {'form': form})

# View to query people working for a company and the cities they live in
def query_company(request):
```

```
    results = None
    if request.method == 'POST':
        form = CompanyQueryForm(request.POST)
        if form.is_valid():
            company_name = form.cleaned_data['company_name']
            works_entries = Works.objects.filter(company_name=company_name)
            results = []
            # For each person working at the given company, try to find their city from the Lives table.
            for work in works_entries:
                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 = CompanyQueryForm()
    return render(request, 'directory/query_company.html', {'form': form, 'results': results})
```

==========

insert_works.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Insert into WORKS</title>
</head>
<body>
    <h1>Insert Record into WORKS</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Insert</button>
    </form>
    <p>
        <a href="{% url 'query_company' %}">Query Company Data</a>
    </p>
</body>
</html>
```

==========

query_company.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Query Company Data</title>
</head>
<body>
    <h1>Retrieve People Working for a Company</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Search</button>
    </form>
```

```
    {% if results %}
      <h2>Results:</h2>
      <ul>
        {% for entry in results %}
          <li>{{ entry.person_name }} lives in {{ entry.city }}</li>
        {% endfor %}
      </ul>
    {% elif results is not none %}
      <p>No records found.</p>
    {% endif %}

    <p>
      <a href="{% url 'insert_works' %}">Insert into WORKS</a>
    </p>
</body>
</html>
```

==========

Output: