IT Lab 7: Databases

Name: Rajvardhan Reddy
Roll Number: 19
Section: B
Batch: B1
Registration Number: 180905093

P1)

settings.py:

```
"""

Django settings for week8v2 project.

Generated by 'django-admin startproject' using Django 3.2.

For more information on this file, see

https://docs.djangoproject.com/en/3.2/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.2/ref/settings/

"""

from pathlib import Path

import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.

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

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET_KEY = 'django-insecure-j3%$lxa5-10w#zi+a=k5!z8bpj32!ttx5f%r!z0+h8#k_s8u-n'

# SECURITY WARNING: don't run with debug turned on in production!
```

```
DEBUG = True
ALLOWED_HOSTS = ['127.0.0.1']
# Application definition
INSTALLED_APPS = [
    'prob4.apps.Prob4Config',
    'prob3.apps.Prob3Config',
    'prob2.apps.Prob2Config',
    'prob1.apps.Prob1Config',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
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 = 'week8v2.urls'
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,'templates')],
        '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 = 'week8v2.wsgi.application'
# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'itlabweek8v2',
        'USER' : 'itlabuser',
        'PASSWORD' : 'incorrect',
        'HOST' : 'localhost'
    }
}
# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators
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',
    },
]
# Internationalization
# https://docs.djangoproject.com/en/3.2/topics/i18n/
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
```

```
USE_TZ = True
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/
STATIC_URL = '/static/'
# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

models.py:

```
from django.db import models
class Category(models.Model):
    name = models.CharField(max_length=100,primary_key=True)
    numberOfVisits = models.IntegerField()
    numberOfLikes = models.IntegerField()
# Create your models here
class Page(models.Model):
    category = models.CharField(max_length=100)
    title = models.CharField(max_length=100)
    url = models.URLField(primary_key=True)
    view = models.IntegerField()
```

forms.py:

```
from prob1.models import Category
from django import forms
class CategoryForm(forms.Form):
    name = forms.CharField(max_length=100)
    numberOfVisits = forms.IntegerField()
    numberOfLikes = forms.IntegerField()
class PageForm(forms.Form):
    category = forms.CharField(max_length=100)
    title = forms.CharField(max_length=100)
    url = forms.URLField()
    view = forms.IntegerField()
```

views.py:

```
from django.shortcuts import render
from .forms import CategoryForm,PageForm
from .models import Category,Page
# Create your views here.
def home(request):
    return render(request,'prog1.html')
def category(request):
    form1 = CategoryForm()
    form = CategoryForm(request.POST)
    if form.is_valid():
        name = form.cleaned_data["name"]
        nov = form.cleaned_data["numberOfVisits"]
        nol = form.cleaned_data["numberOfLikes"]
        Category.objects.create(name = name, numberOfVisits = nov,numberOfLikes = nol)
    return render(request,'prog1p1.html',{"form":form1})
def page(request):
    form1 = PageForm()
    form = PageForm(request.POST)
    if form.is_valid():
        category = form.cleaned_data['category']
        title = form.cleaned_data['title']
        url = form.cleaned_data['url']
        view = form.cleaned_data['view']
        Page.objects.create(category = category,title = title,url = url,view = view)
    return render(request,'prog1p2.html',{"form":form1})
def display(request):
    pages = Page.objects.all()
    categories = Category.objects.all()
    return render(request,'prog1p3.html',{"pages":pages,"categories":categories})
```

urls.py:

```
from django.urls import path
from . import views
urlpatterns = [
    path('',views.home,name="home"),
    path('category',views.category,name="category"),
    path('page',views.page,name = "page"),
    path('display',views.display,name="display") ]
```

0001_initial.py:

```
# Generated by Django 3.2 on 2021-05-26 13:30
from django.db import migrations, models
class Migration(migrations.Migration):
    initial = True
    dependencies = [
    ]
    operations = [
        migrations.CreateModel(
            name='Category',
            fields=[
                ('name', models.CharField(max_length=100, primary_key=True, serialize=False)),
                ('numberOfVisits', models.IntegerField()),
                ('numberOfLikes', models.IntegerField()),
            ],
        ),
        migrations.CreateModel(
            name='Page',
            fields=[
                ('category', models.CharField(max_length=100)),
                ('title', models.CharField(max_length=100)),
                ('url', models.URLField(primary_key=True, serialize=False)),
                ('view', models.IntegerField()),
            ],
        ),
    ]
```

urls.py:

```
from django.contrib import admin
from django.urls import path,include
urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('prob1.urls'))
    #path('',include('prob2.runserverurls'))
    #path('',include('prob3.urls'))
    #path('',include('prob4.urls'))
]
```

prog1.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <a href="{% url 'category' %}">Enter Information to category table</a><br>
    <a href="{% url 'page'}">Enter Information to Page table</a><br>
    <a href="{% url 'display'}">Display Category table and page table</a><br>
</body>
</html>
```

prog1p1.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form action="category" method="POST">
        {% csrf_token %}
        <table>
            {{form.as_table}}
        </table>
        <input type="submit" value="insert">
    </form><br>
    <a href="{% url 'home' %}">Go back to home</a>
</body>
</html>
```

prog1p2.html:

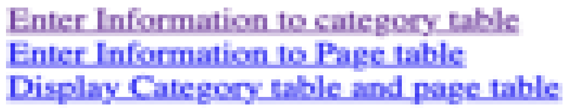
```
<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Document</title>

</head>

<body>

    <form action="page" method="POST">

        {% csrf_token %}

        <table>

            {{form.as_table}}

        </table>

        <input type="submit" value="insert">

    </form>

    <a href="{% url 'home' %}">back to home</a>

</body>

</html>
```

prog1p3.html:

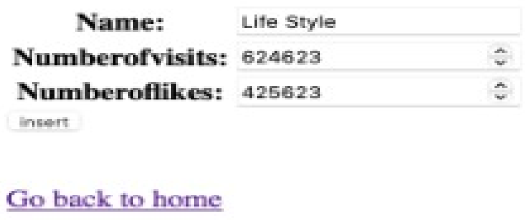
```
<html lang="en">

<head>

    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Document</title>

</head>

<body>

    <h1>Category Table:</h1><br>

    <table>

        <thead>

            <td>Name</td>

            <td>Number of Visits</td>

            <td>Number of likes</td>

        </thead>

        {% for category in categories %}

        <tr>

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

            <td>{{category.numberOfVisits}}</td>

            <td>{{category.numberOfLikes}}</td>

        </tr>

        {% endfor %}

    </table>

    <br>

    <h1>Page table</h1>

    <table>
```

```
        <thead>
            <td>Category</td>
            <td>Title</td>
            <td>URL</td>
            <td>View</td>
        </thead>
        {% for page in pages %}
        <tr>
            <td>{{page.category}}</td>
            <td>{{page.title}}</td>
            <td>{{page.url}}</td>
            <td>{{page.view}}</td>
        </tr>
        {% endfor %}
    </table>
    <br>
    <a href="{% url 'home' %}">back to home</a>
</body>
</html>
```

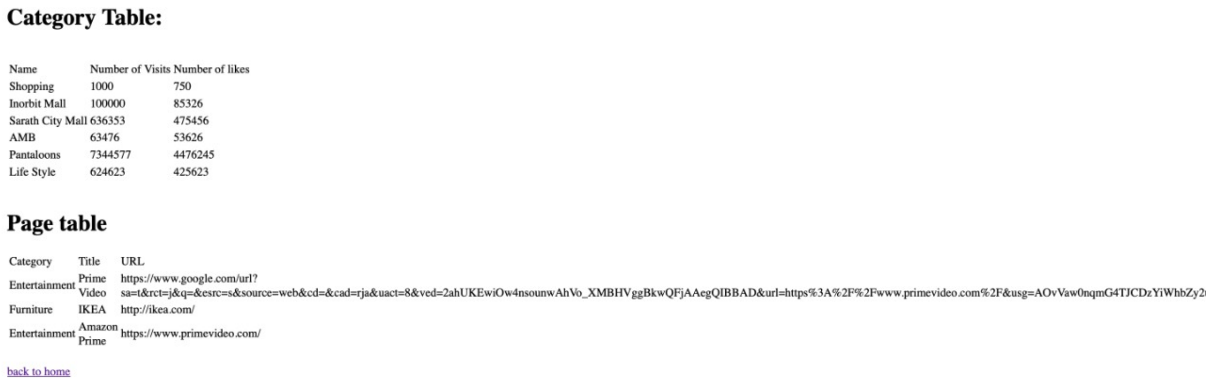
Output:



Enter Information to category table
Enter Information to Page table
Display Category table and page table



Name:	Life Style
Numberofvisits:	624623
Numberoflikes:	425623

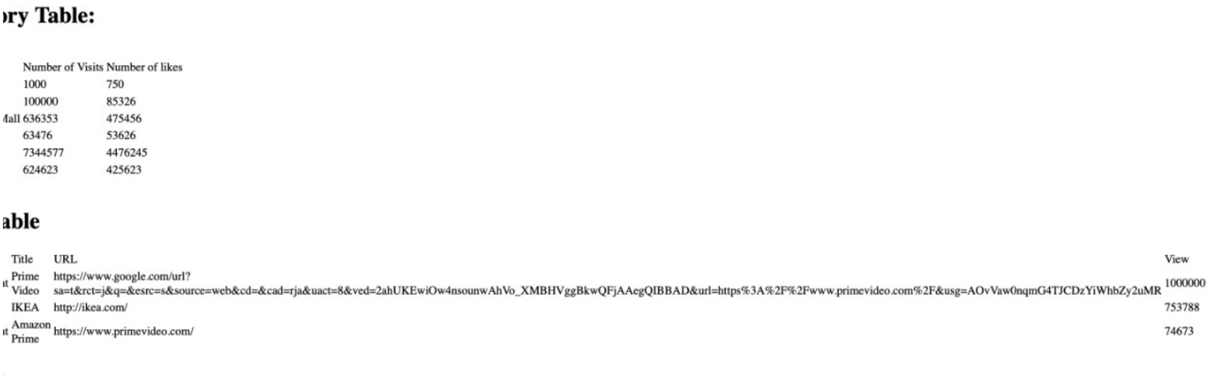


Category Table:

Name	Number of Visits	Number of likes
Shopping	1000	750
Inorbit Mall	100000	85326
Sarath City Mall	636353	475456
AMB	63476	53626
Pantaloons	7344577	4476245
Life Style	624623	425623

Page table

Category	Title	URL
Entertainment	Prime Video	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiOw4nsounwAhVo_XMBHVggBkwQFjAAegQIBBAD&url=https%3A%2F%2Fwww.primevideo.com%2F&usg=AOvVaw0nqmG4TJCDzYiWhbZy2
Furniture	IKEA	http://ikea.com/
Entertainment	Amazon Prime	https://www.primevideo.com/



ory Table:

	Number of Visits	Number of likes
	1000	750
	100000	85326
ıall	636353	475456
	63476	53626
	7344577	4476245
	624623	425623

able

	Title	URL	View
ıt	Prime Video	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiOw4nsounwAhVo_XMBHVggBkwQFjAAegQIBBAD&url=https%3A%2F%2Fwww.primevideo.com%2F&usg=AOvVaw0nqmG4TJCDzYiWhbZy2uMR	1000000
	IKEA	http://ikea.com/	753788
ıt	Amazon Prime	https://www.primevideo.com/	74673

P2)

models.py:

```
from django.db import models

from django.db.models.fields.related import ForeignKey

# Create your models here.

class Works(models.Model):

    name = models.CharField(max_length=100)

    company = models.CharField(max_length=100)

    salary = models.IntegerField()

class Lives(models.Model):

    name = models.CharField(max_length=100)

    street = models.CharField(max_length=200)

    city = models.CharField(max_length=50)
```

forms.py:

```
from django import forms


class Employee(forms.Form):

    name = forms.CharField(max_length=100)

    company = forms.CharField(max_length=100)

    salary = forms.IntegerField()

    street = forms.CharField(max_length=200)

    city = forms.CharField(max_length=50)

class Company(forms.Form):
    company = forms.CharField(max_length=100)
```

views.py:

```
from django.shortcuts import render

from .models import Works,Lives

from .forms import Employee,Company

# Create your views here.

def home(request):

    return render(request,'prog2.html')

def portal(request):

    form = Employee()

    form1 = Employee(request.POST)

    if form1.is_valid():

        name = form1.cleaned_data['name']

        company = form1.cleaned_data['company']

        salary = form1.cleaned_data['salary']

        street = form1.cleaned_data['street']

        city = form1.cleaned_data['city']

        Works.objects.create(name=name,company=company,salary=salary)

        Lives.objects.create(name=name,street=street,city=city)

    return render(request,'prog2p1.html',{"form":form})

def search(request):

    form = Company()

    form1 = Company(request.POST)
    if form1.is_valid():
        company = form1.cleaned_data["company"]
        employa = Works.objects.all().filter(company = company)
        employees = []
        for e in employa:
            employees.append(Lives.objects.get(name = e.name))
        return render(request,"prog2p2.html",{"form":form1,"employees":employees})
    return render(request,"prog2p2.html",{"form":form})
```

urls.py:

```
from django.urls import path
from . import views
urlpatterns = [
    path('',views.home,name="home"),
    path('portal',views.portal,name="portal"),
    path('search',views.search,name="search")
]
```

0001_initial.py:

```
# Generated by Django 3.2 on 2021-05-26 13:30
from django.db import migrations, models
class Migration(migrations.Migration):
    initial = True
    dependencies = [
    ]
    operations = [
        migrations.CreateModel(
            name='Lives',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=100)),
                ('street', models.CharField(max_length=200)),
                ('city', models.CharField(max_length=50)),
            ],
        ),
        migrations.CreateModel(
            name='Works',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=100)),
                ('company', models.CharField(max_length=100)),
                ('salary', models.IntegerField()),
            ],
        ),
    ]
```

urls.py:

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

urlpatterns = [

    path('admin/', admin.site.urls),

   #path('',include('prob1.urls'))

    path('',include('prob2.runserverurls'))

    #path('',include('prob3.urls'))

    #path('',include('prob4.urls'))

]
```

prog2.html:

```
<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Document</title>

</head>

<body>

    <a href="{% url 'portal' %}">update employee portal</a><br>

    <a href="{% url 'search' %}">Find the employee list of a company</a>

</body>

</html>
```

prog2p1.html:

```
<html lang="en">
```

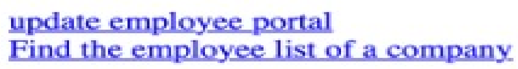
```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form action="portal" method="POST">
        {% csrf_token %}
        <table>
            {{form.as_table}}
        </table><br>
        <input type="submit" value="insert">
    </form>
    <a href="{% url 'home' %}">Go back to home</a>
</body>
</html>
```

prog2p2.html:

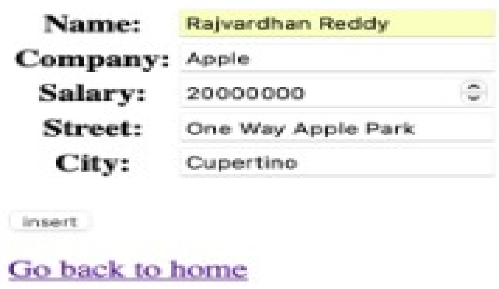
```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

</head>

<body>

    <form action="search" method="POST">

        {% csrf_token %}

        {{form}}

        <br>

        <input type="submit" value="search">

    </form><br>

    <table>

        <thead>

            <td>name</td>

            <td>city</td>

        </thead>

        {% for employee in employees %}

        <tr>

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

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

        </tr>

        {% endfor %}

    </table>

    <a href="{% url 'home' %}">Go back to home</a>

</body>
```

</html>

Output:

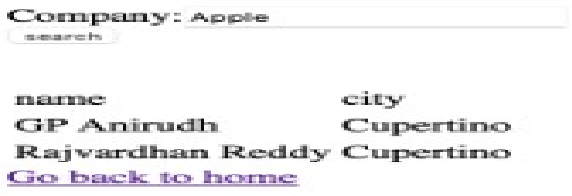


update employee portal
Find the employee list of a company



Name: Rajvardhan Reddy
Company: Apple
Salary: 20000000
Street: One Way Apple Park
City: Cupertino

Go back to home



Company: Apple

```
name                city
GP Anirudh          Cupertino
Rajvardhan Reddy    Cupertino
Go back to home
```

P3)

models.py:

from django.db import models

from django.db.models.aggregates import Count

# Create your models here.

```
class Publisher(models.Model):
    name = models.CharField(max_length=100)
    street = models.CharField(max_length=200)
    city = models.CharField(max_length=50)
    state = models.CharField(max_length=50)
    country = models.CharField(max_length=50)
    site = models.URLField()
class Au(models.Model):
    fname = models.CharField(max_length=100)
    lname = models.CharField(max_length=100)
    em = models.EmailField()
class Book(models.Model):
    title = models.CharField(max_length=200)
    pdate = models.DateField()
    authors = models.ManyToManyField(Au)
    publisher = models.ForeignKey(Publisher,on_delete=models.CASCADE)
```

forms.py:

```
from django import forms
class PublisherForm(forms.Form):
    name = forms.CharField(max_length=100)
    street = forms.CharField(max_length=200)
    city = forms.CharField(max_length=50)
    state = forms.CharField(max_length=50)
    country = forms.CharField(max_length=50)

    site = forms.URLField()

class AuthorForm(forms.Form):

    fname = forms.CharField(max_length=100, label="first name")

    lname = forms.CharField(max_length=100, label="last name")

    email = forms.EmailField()

class BookForm(forms.Form):

    title = forms.CharField(max_length=200)

    pdate = forms.DateField(label="publication date")

    pname = forms.CharField(max_length=100,label="Publisher name")

    anames = forms.CharField(max_length=400,label="Enter first names of authors by space seperation")

class BookSearch(forms.Form):

    title = forms.CharField(max_length=200)

class AuthorSearch(forms.Form):

    fname = forms.CharField(max_length=100, label="enter the first name")

class PublisherSearch(forms.Form):

    name = forms.CharField(max_length=100)
```

views.py :

```
from django.shortcuts import render

from .forms import AuthorForm,PublisherForm,BookForm,AuthorSearch,PublisherSearch,BookSearch

from .models import Au,Publisher,Book

# Create your views here.
```

```
def home(request):
    return render(request,'prog3.html')
def publisherEntry(request):
    form = PublisherForm()
    form1 = PublisherForm(request.POST)
    if form1.is_valid():
        name = form1.cleaned_data["name"]
        street = form1.cleaned_data["street"]
        city = form1.cleaned_data["city"]
        state = form1.cleaned_data["state"]
        country = form1.cleaned_data["country"]
        site = form1.cleaned_data["site"]
        Publisher.objects.create(name = name,street = street,city = city,state = state,country = country,site = site)
    return render(request,'prog3p1.html',{"form":form})
def authorEntry(request):
    form = AuthorForm()
    form1 = AuthorForm(request.POST)
    if form1.is_valid():
        fname = form1.cleaned_data["fname"]
        lname = form1.cleaned_data["lname"]
        email = form1.cleaned_data["email"]
        Au.objects.create(fname = fname,lname = lname,em = email)
    return render(request,'prog3p2.html',{"form":form})
```

```
def bookEntry(request):
    form = BookForm()
    form1 = BookForm(request.POST)
    if form1.is_valid():
        a = form1.cleaned_data
        title = a["title"]
        pdate = a["pdate"]
        pname = a["pname"]
        anames = a["anames"].split()
        print(anames)
        publisher = Publisher.objects.get(name = pname)
        authors = []
        book = Book(title = title,pdate = pdate,publisher = publisher)
        book.save()
        for i in anames:
            a = Au.objects.get(fname = i)
            book.authors.add(a)
        book.save()
    return render(request,'prog3p3.html',{"form":form})
def searchBook(request):
    form = BookSearch()
    form1 = BookSearch(request.POST)
    if form1.is_valid():
        title = form1.cleaned_data["title"]
        book = Book.objects.get(title = title)
        return render(request,'prog3p4.html',{"form":form1,"book":book})
    return render(request,'prog3p4.html',{"form":form})
def searchAuthor(request):
    form = AuthorSearch()
    form1 = AuthorSearch(request.POST)
    if form1.is_valid():
        fname = form1.cleaned_data["fname"]
        author = Au.objects.get(fname = fname)
        return render(request,'prog3p5.html',{"form":form1,"author":author})
    return render(request,'prog3p5.html',{"form":form})
def searchPublisher(request):
    form = PublisherSearch()
    form1 = PublisherSearch(request.POST)
    if form1.is_valid():
        name = form1.cleaned_data["name"]
        publisher = Publisher.objects.get(name = name)
        return render(request,'prog3p6.html',{"form":form1,"publisher":publisher})
    return render(request,'prog3p6.html',{"form":form})
```

urls.py:

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path('',views.home,name="home"),
    path('publisherEntry',views.publisherEntry,name="publisherEntry"),
    path('authorEntry',views.authorEntry,name="authorEntry"),
    path('bookEntry',views.bookEntry,name="bookEntry"),
    path('searchBook',views.searchBook,name="searchBook"),
    path('searchAuthor',views.searchAuthor,name="searchAuthor"),
    path('searchPublisher',views.searchPublisher,name="searchPublisher"),
]
```

0001_inital.py:

```
# Generated by Django 3.2 on 2021-05-26 13:30
from django.db import migrations, models
import django.db.models.deletion
class Migration(migrations.Migration):
    initial = True
    dependencies = [
    ]
    operations = [
        migrations.CreateModel(
            name='Au',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('fname', models.CharField(max_length=100)),
                ('lname', models.CharField(max_length=100)),
                ('em', models.EmailField(max_length=254)),
            ],
        ),
        migrations.CreateModel(
            name='Publisher',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=100)),
                ('street', models.CharField(max_length=200)),
                ('city', models.CharField(max_length=50)),
                ('state', models.CharField(max_length=50)),
                ('country', models.CharField(max_length=50)),
                ('site', models.URLField()),
            ],
        ),
        migrations.CreateModel(
            name='Book',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('title', models.CharField(max_length=200)),
                ('pdate', models.DateField()),
                ('authors', models.ManyToManyField(to='prob3.Au')),
                ('publisher',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='prob3.publisher')),
            ],
        ),
    ]
```

prog3p1.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Publisher Entry</title>
</head>
<body>
    <h1>Publisher Registration:</h1>
    <form action="publisherEntry" method="POST">
        {% csrf_token %}
        <table>
            {{form.as_table}}
        </table>
        <input type="submit" value=register>
    </form>
    <a href="{% url 'home' %}">Go back to home</a>
</body>
</html>
```

urls.py:

```
from django.contrib import admin
from django.urls import path,include
urlpatterns = [
    path('admin/', admin.site.urls),
    #path('',include('prob1.urls'))
    #path('',include('prob2.runserverurls'))
    path('',include('prob3.urls'))
    #path('',include('prob4.urls'))
]
```

prog3.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Main Page</title>
</head>
<body>
    <a href="{% url 'publisherEntry' %}">Register a publisher</a><br>
    <a href="{% url 'authorEntry' %}">Register a author</a><br>
    <a href="{% url 'bookEntry' %}">Register a book</a><br>
    <a href="{% url 'searchBook' %}">Search for a book</a><br>
    <a href="{% url 'searchAuthor' %}">Search for a author</a><br>
    <a href="{% url 'searchPublisher' %}">Search for a publisher</a>
</body>
</html>
```

prog3p1.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Publisher Entry</title>
</head>
<body>
    <h1>Publisher Registration:</h1>
    <form action="publisherEntry" method="POST">
        {% csrf_token %}
        <table>
            {{form.as_table}}
        </table>
        <input type="submit" value=register>
    </form>
    <a href="{% url 'home' %}">Go back to home</a>
</body>
</html>
```

prog3p2.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Author Entry</title>
</head>
<body>
    <h1>Author Registration:</h1>
    <form action="authorEntry" method="POST">
        {% csrf_token %}
        <table>
            {{form.as_table}}
        </table>
        <input type="submit" value=register>
    </form>
    <a href="{% url 'home' %}">Go back to home</a>
</body>
```

```
</html>
```

prog3p3html:

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Book Entry</title>
</head>
<body>
   <h1>Book Registration:</h1>
   <form action="bookEntry" method="POST">
      {% csrf_token %}
      <table>
         {{form.as_table}}
      </table>
      <input type="submit" value=register>
   </form>
   <a href="{% url 'home' %}">Go back to home</a>
</body>
</html>
```

prog3p4.html:

```
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Book Search</title>
</head>
<body>
    <h1>Search for book</h1>
    <form action="searchBook" method="POST">
        {% csrf_token %}
        {{form}}
        <br><input type="submit" value = "Search">
    </form>
    <table>
        <thead>
            <td>Title</td>
            <td>Published Date</td>
            <td>Name of the Publisher</td>
            <td>Name of the authors</td>
        </thead>
        <tr>
            <td>{{book.title}}</td>
            <td>{{book.pdate}}</td>
            <td>{{book.publisher.name}}</td>

            <td>

               {% for author in book.authors.all %}

               {{author.fname}} {{author.lname}} <br>

               {% endfor %}

            </td>

         </tr>

      </table>

      <a href="{% url 'home' %}">Go back to home</a>

   </body>

   </html>
```

prog3p5.html:

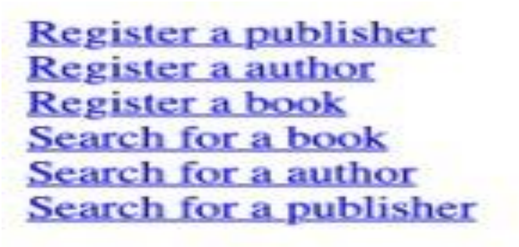
```
<html lang="en">

<head>

   <meta charset="UTF-8">

   <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

   <title>Author Search</title>

</head>

<body>

   <h1>Search for Author</h1>

   <form action="searchAuthor" method="POST">

      {% csrf_token %}
```

```
        {{form}}
        <br><input type="submit" value = "Search">
    </form>
    <table>
        <thead>
            <td>First Name</td>
            <td>Last Name</td>
            <td>email</td>
        </thead>
        <tr>
            <td>{{author.fname}}</td>
            <td>{{author.lname}}</td>
            <td>{{author.em}}</td>
        </tr>
    </table>
    <a href="{% url 'home' %}">Go back to home</a>
</body>
</html>
```

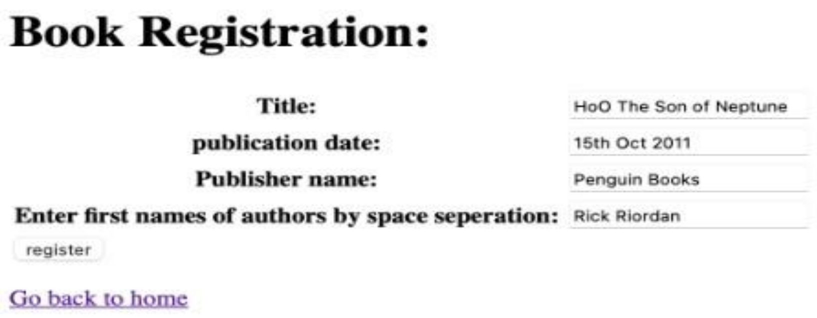
Prog3p6.html:

```
from django.contrib import admin
from django.urls import path,include
urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('prob1.urls'))

    #path('',include('prob2.runserverurls'))

    #path('',include('prob3.urls'))

    #path('',include('prob4.urls'))

]
```

Output:



Register a publisher
Register a author
Register a book
Search for a book
Search for a author
Search for a publisher



Book Registration:

Title: HoO The Son of Neptune

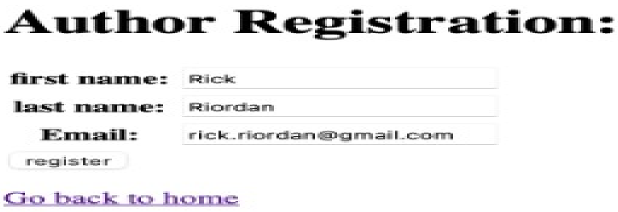
publication date: 15th Oct 2011

Publisher name: Penguin Books

Enter first names of authors by space seperation: Rick Riordan

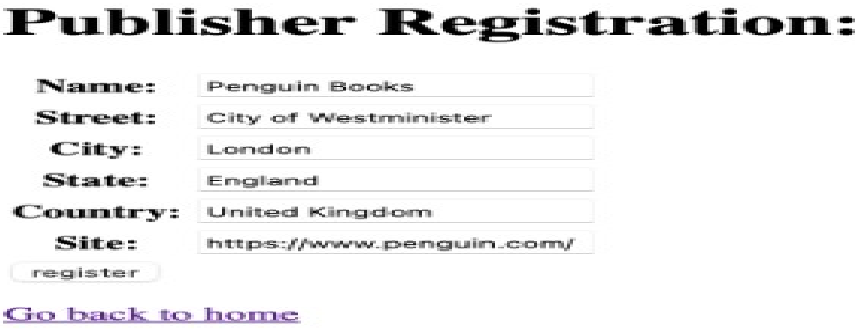
register

Go back to home



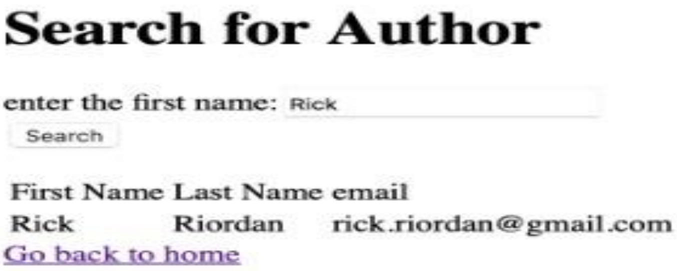
Author Registration:

first name: Rick
last name: Riordan
Email: rick.riordan@gmail.com



Publisher Registration:

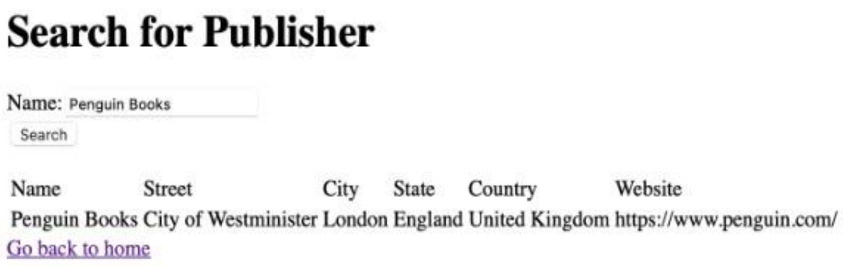
Name: Penguin Books
Street: City of Westminister
City: London
State: England
Country: United Kingdom
Site: https://www.penguin.com/



Search for Author

enter the first name: Rick

First Name	Last Name	email
Rick	Riordan	rick.riordan@gmail.com



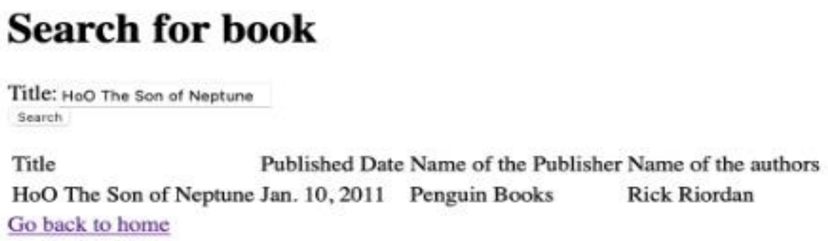
Search for Publisher

Name: Penguin Books

Search

Name	Street	City	State	Country	Website
Penguin Books	City of Westminister	London	England	United Kingdom	https://www.penguin.com/

Go back to home



Search for book

Title: HoO The Son of Neptune

Search

Title	Published Date	Name of the Publisher	Name of the authors
HoO The Son of Neptune	Jan. 10, 2011	Penguin Books	Rick Riordan

Go back to home

P4)

models.py:

```
from django.db import models

# Create your models here.

class Product(models.Model):

    title = models.CharField(max_length=100)

    price = models.IntegerField()

    desc = models.TextField()
```

forms.py:

```
from django import forms

class ProductForm(forms.Form):

    title = forms.CharField(max_length=100)

    price = forms.IntegerField()
    desc = forms.CharField(widget=forms.Textarea(),label="description")
```

views.py:

```
from django.shortcuts import render
from .forms import ProductForm
from .models import Product
# Create your views here.
def home(request):
    return render(request,'prog4.html')
def entry(request):
    form1 = ProductForm(request.POST)
    form = ProductForm()
    if form1.is_valid():
        title = form1.cleaned_data['title']
        price = form1.cleaned_data['price']
        desc = form1.cleaned_data['desc']
        Product.objects.create(title = title,price = price,desc = desc)
    return render(request,'prog4p1.html',{"form":form})
def index(request):
    products = Product.objects.all()
    return render(request,'prog4p2.html',{"products":products})
```

urls.py:

```
from django.urls import path

from . import views

urlpatterns = [

    path('',views.home,name="home"),

    path('entry',views.entry,name="entry"),

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

]
```

0001_initial.py:

```
# Generated by Django 3.2 on 2021-05-26 13:30

from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [

    ]

    operations = [

        migrations.CreateModel(

            name='Product',

            fields=[

                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),

                ('title', models.CharField(max_length=100)),

                ('price', models.IntegerField()),

                ('desc', models.TextField()),

            ],),]
```

urls.py:

```
from django.contrib import admin

from django.urls import path,include

urlpatterns = [

    path('admin/', admin.site.urls),

    #path('',include('prob1.urls'))

    #path('',include('prob2.runserverurls'))

    #path('',include('prob3.urls'))

    path('',include('prob4.urls')) ]
```

prog4.html:

```
<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Document</title>

</head>

<body>

    <a href="{% url 'entry' %}">Enter a new product</a><br>

    <a href="{% url 'index' %}">View Products</a>

</body>

</html>
```

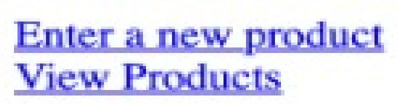
prog4p1.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form action="entry" method="POST">
        {% csrf_token %}
        <table>
            {{form.as_table}}
        </table>
        <input type="submit" value="add">
    </form>
    <a href="{% url 'home' %}">Go back to home</a>
</body>
</html>
```

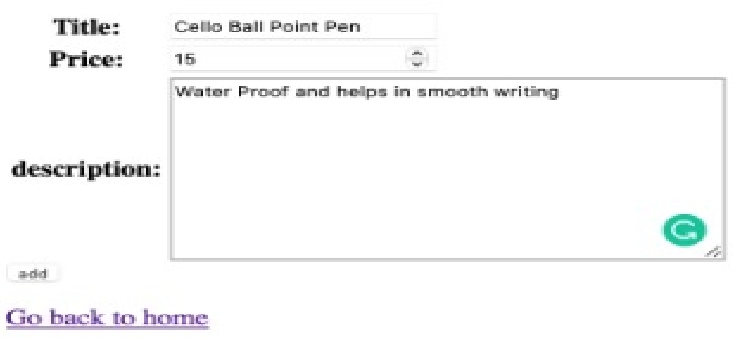
prog4p2.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Products:</h1><br>
    <ul>
        {% for product in products %}
        <li>{{product.title}} <br>₹{{product.price}} <br>{{product.desc}}</li>
        {% endfor %}
    </ul>
</body>
</html>
```

Output :

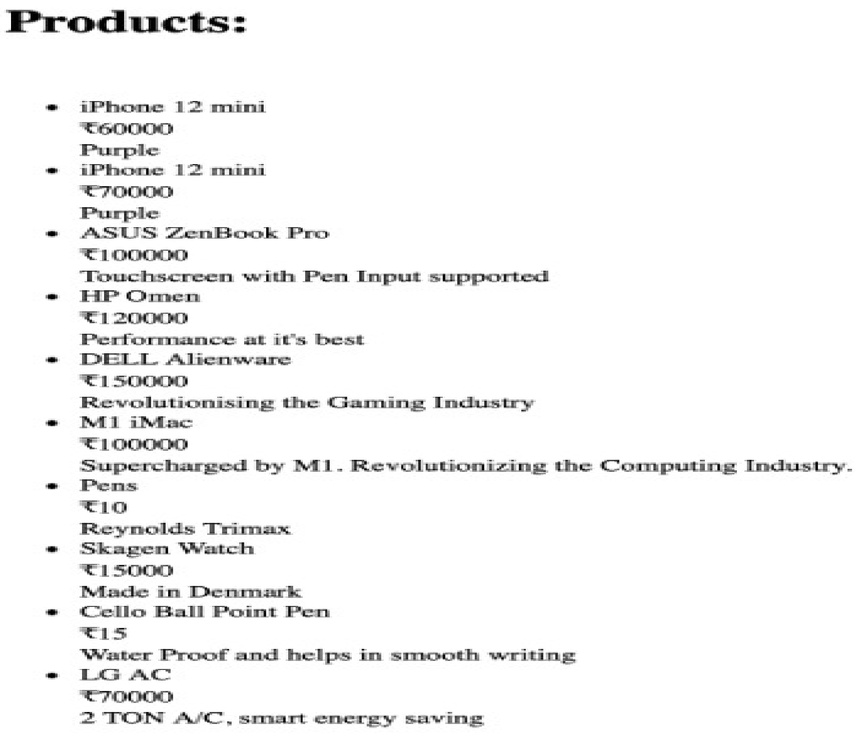


Enter a new product
View Products



Title: Cello Ball Point Pen
Price: 15
description: Water Proof and helps in smooth writing

Go back to home



Products:

- iPhone 12 mini
  ₹60000
  Purple
- iPhone 12 mini
  ₹70000
  Purple
- ASUS ZenBook Pro
  ₹100000
  Touchscreen with Pen Input supported
- HP Omen
  ₹120000
  Performance at it's best
- DELL Alienware
  ₹150000
  Revolutionising the Gaming Industry
- M1 iMac
  ₹100000
  Supercharged by M1. Revolutionizing the Computing Industry.
- Pens
  ₹10
  Reynolds Trimax
- Skagen Watch
  ₹15000
  Made in Denmark
- Cello Ball Point Pen
  ₹15
  Water Proof and helps in smooth writing
- LG AC
  ₹70000
  2 TON A/C, smart energy saving