**ARUNIMA SINGH THAKUR**

**SECTION C**

**ROLL NO. 31**

**180905218**

**IT LAB 7**

**27TH MAY 2021**

week8/settings.py

from pathlib import Path

import os

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

SECRET\_KEY = 'django-insecure-)mskgr)a%7!h)t&4q)&+07\*kmm9)xs0182\_xg72z6%)jc@0!1c'

DEBUG = True

ALLOWED\_HOSTS = ['127.0.0.1']

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 = 'week8.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 = 'week8.wsgi.application'

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'week8',

'USER' : 'Arunima',

'PASSWORD' : '1234',

'HOST' : 'localhost'

}

}

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\_L10N = True

USE\_TZ = True

STATIC\_URL = '/static/'

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

1) prob1/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()

prob1/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()

prob1/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})

prob1/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")

]

prob1/migrations/0001\_initial.py

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()),

],

),

]

week8/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.urls'))*

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

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

]

templates/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>

templates/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>

templates/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>

templates/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>

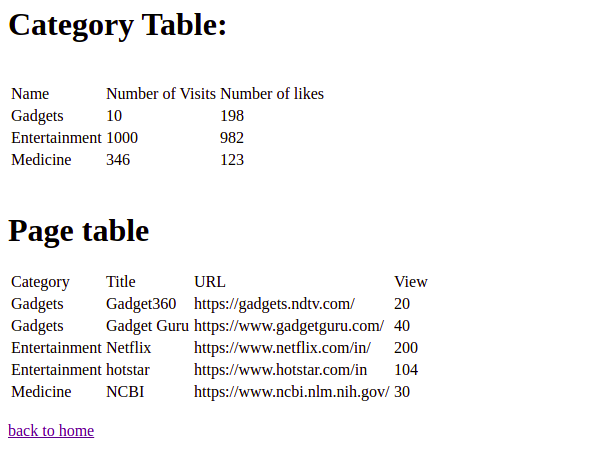
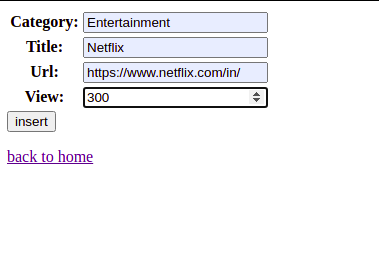
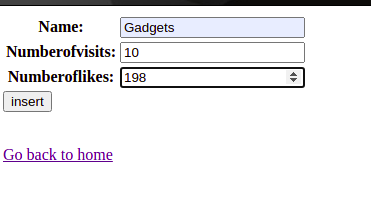
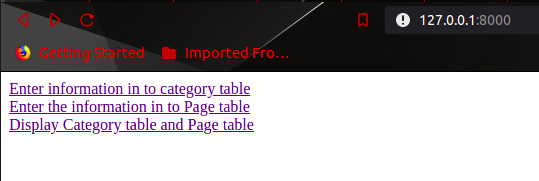
<br>

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

</body>

</html>

Outputs



2) prob2/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)

prob2/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)

prob2/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})

prob2/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")

]

prob2/migrations/0001\_initial.py

*# Generated by Django 3.2 on 2021-05-24 16:22*

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()),

],

),

]

week8/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.urls'))

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

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

]

templates/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>

templates/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>

templates/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 %}

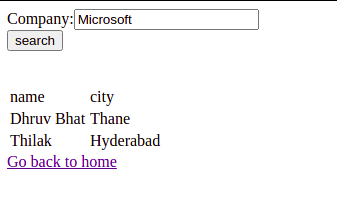
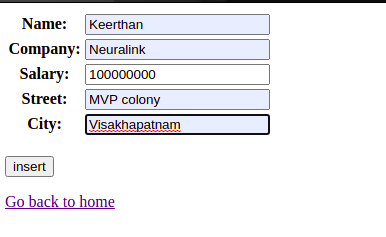
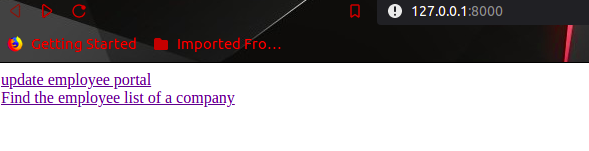
</table>

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

</body>

</html>

Outputs



3) prob3/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)

prob3/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)

prob3/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})

prob3/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"),

]

prob3/migrations/0001\_initial.py

*# Generated by Django 3.2 on 2021-05-25 11:15*

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')),

],

),

]

week8/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.urls'))*

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

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

]

templates/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>

templates/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>

templates/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>

templates/prog3p3.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 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>

templates/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>

templates/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>

templates/prog3p6.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 Search</title>

</head>

<body>

<h1>Search for Publisher</h1>

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

{% csrf\_token %}

{{form}}

<br><input type="submit" value = "Search">

</form>

<table>

<thead>

<td>Name</td>

<td>Street</td>

<td>City</td>

<td>State</td>

<td>Country</td>

<td>Website</td>

</thead>

<tr>

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

<td>{{publisher.street}}</td>

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

<td>{{publisher.state}}</td>

<td>{{publisher.country}}</td>

<td>{{publisher.site}}</td>

</tr>

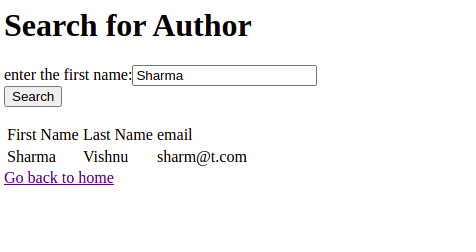
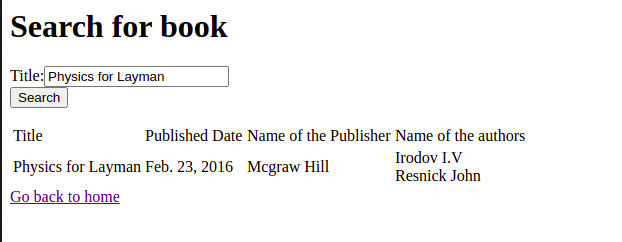
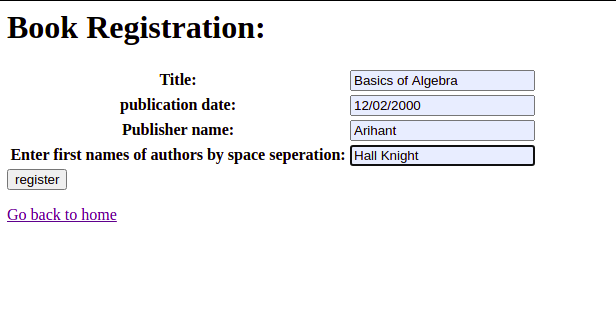
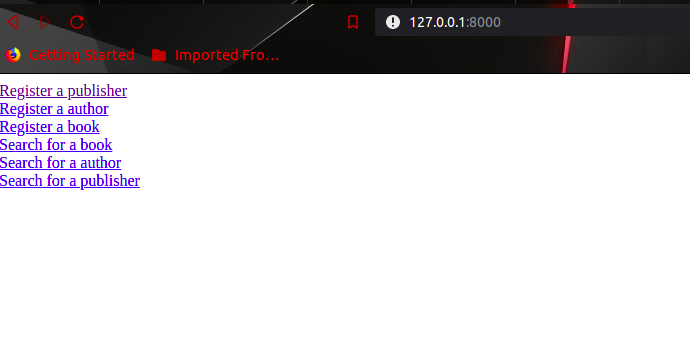
</table>

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

</body>

</html>

Outputs



4) prob4/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()

prob4/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")

prob4/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})

prob4/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")

]

prob4/migrations/0001\_initial.py

*# Generated by Django 3.2 on 2021-05-24 17:46*

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()),

],

),

]

week8/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.urls'))*

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

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

]

templates/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>

templates/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>

templates/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>

Outputs

