1)

# books/forms.py

from django import forms

from .models import Author, Publisher, Book

class AuthorForm(forms.ModelForm):

class Meta:

model = Author

fields = ['first\_name', 'last\_name', 'email']

class PublisherForm(forms.ModelForm):

class Meta:

model = Publisher

fields = ['name', 'street\_address', 'city', 'state\_province', 'country', 'website']

class BookForm(forms.ModelForm):

authors = forms.ModelMultipleChoiceField(queryset=Author.objects.all(), widget=forms.CheckboxSelectMultiple)

publisher = forms.ModelChoiceField(queryset=Publisher.objects.all())

class Meta:

model = Book

fields = ['title', 'publication\_date', 'authors', 'publisher']

# books/models.py

from django.db import models

class Author(models.Model):

first\_name = models.CharField(max\_length=100)

last\_name = models.CharField(max\_length=100)

email = models.EmailField()

def \_\_str\_\_(self):

return f"{self.first\_name} {self.last\_name}"

class Publisher(models.Model):

name = models.CharField(max\_length=255)

street\_address = models.CharField(max\_length=255)

city = models.CharField(max\_length=100)

state\_province = models.CharField(max\_length=100)

country = models.CharField(max\_length=100)

website = models.URLField()

def \_\_str\_\_(self):

return self.name

class Book(models.Model):

title = models.CharField(max\_length=255)

publication\_date = models.DateField()

authors = models.ManyToManyField(Author)

publisher = models.ForeignKey(Publisher, on\_delete=models.CASCADE)

def \_\_str\_\_(self):

return self.title

# books/views.py

from django.shortcuts import render, redirect

from .forms import AuthorForm, PublisherForm, BookForm

from .models import Author, Publisher, Book

def add\_author(request):

if request.method == 'POST':

form = AuthorForm(request.POST)

if form.is\_valid():

form.save()

return redirect('add\_author')

else:

form = AuthorForm()

return render(request, 'books/add\_author.html', {'form': form})

def add\_publisher(request):

if request.method == 'POST':

form = PublisherForm(request.POST)

if form.is\_valid():

form.save()

return redirect('add\_publisher')

form = PublisherForm()

return render(request, 'books/add\_publisher.html', {'form': form})

def add\_book(request):

if request.method == 'POST':

form = BookForm(request.POST)

if form.is\_valid():

form.save()

return redirect('add\_book')

else:

form = BookForm()

return render(request, 'books/add\_book.html', {'form': form})

def view\_books(request):

books = Book.objects.all() # Retrieve all books

return render(request, 'books/view\_books.html', {'books': books})

<!-- books/templates/books/view\_books.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Books List</title>

</head>

<body>

<h2>Books List</h2>

<ul>

{% for book in books %}

<li>{{ book.title }} by

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

{{ author.first\_name }} {{ author.last\_name }}

{% endfor %}

(Published by {{ book.publisher.name }})

</li>

{% empty %}

<li>No books available.</li> <

{% endfor %}

</ul>

</body>

</html>

# lab11q1/urls.py

from django.contrib import admin

from django.urls import path, include

from django.views.generic import RedirectView

urlpatterns = [

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

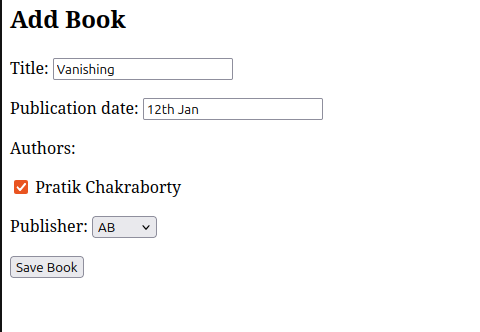
path('books/', include('books.urls')),

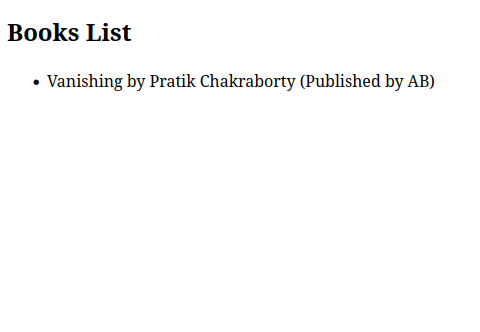
path('', RedirectView.as\_view(url='/books/', permanent=True)),

]









2)

<!-- products/templates/products/add\_product.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Add Product</title>

</head>

<body>

<h1>Add a New Product</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Add Product</button>

</form>

<a href="{% url 'index' %}">Back to product list</a>

</body>

</html>

<!-- products/templates/products/index.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Product List</title>

</head>

<body>

<h1>Product List</h1>

<ul>

{% for product in products %}

<li>{{ product.title }} - ${{ product.price }}<br>{{ product.description }}</li>

{% empty %}

<li>No products available.</li>

{% endfor %}

</ul>

<a href="{% url 'add\_product' %}">Add a new product</a>

</body>

</html>

# products/forms.py

from django import forms

from .models import Product

class ProductForm(forms.ModelForm):

class Meta:

model = Product

fields = ['title', 'price', 'description']

# products/models.py

from django.db import models

class Product(models.Model):

title = models.CharField(max\_length=200)

price = models.DecimalField(max\_digits=10, decimal\_places=2)

description = models.TextField()

def \_\_str\_\_(self):

return self.title

from django.urls import path

from . import views

urlpatterns = [

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

path('add/', views.add\_product, name='add\_product'),

]

# products/views.py

from django.shortcuts import render, redirect

from .models import Product

from .forms import ProductForm

def index(request):

products = Product.objects.all()

return render(request, 'products/index.html', {'products': products})

def add\_product(request):

if request.method == 'POST':

form = ProductForm(request.POST)

if form.is\_valid():

form.save()

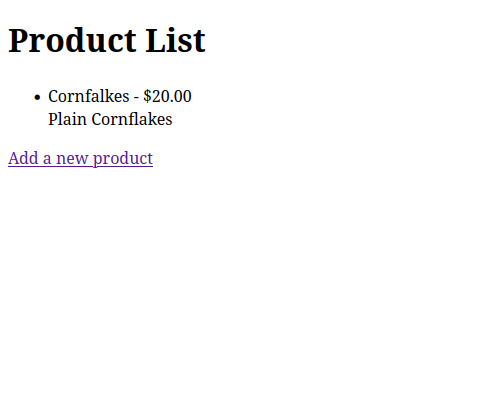
return redirect('index')

else:

form = ProductForm()

return render(request, 'products/add\_product.html', {'form': form})





3)

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

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

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

]

<!DOCTYPE html>

<html>

<head><title>Human Details</title></head>

<body>

<h2>Select a Human</h2>

<form method="get" action="/books/">

<select name="human\_id" onchange="this.form.submit()">

<option value="">-- Select First Name --</option>

{% for h in humans %}

<option value="{{ h.id }}" {% if selected and h.id == selected.id %}selected{% endif %}>

{{ h.first\_name }}

</option>

{% endfor %}

</select>

</form>

{% if selected %}

<h3>Edit / Delete Human</h3>

<form method="post" action="/books/update/{{ selected.id }}/">

{% csrf\_token %}

Last Name: <input type="text" name="last\_name" value="{{ selected.last\_name }}"><br>

Phone: <input type="text" name="phone" value="{{ selected.phone }}"><br>

Address: <input type="text" name="address" value="{{ selected.address }}"><br>

City: <input type="text" name="city" value="{{ selected.city }}"><br>

<button type="submit">Update</button>

</form>

<form method="post" action="/books/delete/{{ selected.id }}/">

{% csrf\_token %}

<button type="submit" onclick="return confirm('Delete this record?')">Delete</button>

</form>

{% endif %}

<hr>

<h3>Add New Human</h3>

<form method="post" action="/books/">

{% csrf\_token %}

First Name: <input type="text" name="first\_name"><br>

Last Name: <input type="text" name="last\_name"><br>

Phone: <input type="text" name="phone"><br>

Address: <input type="text" name="address"><br>

City: <input type="text" name="city"><br>

<button type="submit" name="create">Create</button>

</form>

</body>

</html>

from django.db import models

class Human(models.Model):

first\_name = models.CharField(max\_length=50)

last\_name = models.CharField(max\_length=50)

phone = models.CharField(max\_length=20)

address = models.CharField(max\_length=200)

city = models.CharField(max\_length=50)

def \_\_str\_\_(self):

return self.first\_name

from django.urls import path

from . import views

urlpatterns = [

path('books/', views.index, name='index'), # change '' to 'books/'

path('books/update/<int:human\_id>/', views.update\_human, name='update\_human'),

path('books/delete/<int:human\_id>/', views.delete\_human, name='delete\_human'),

]

from django.shortcuts import render, get\_object\_or\_404, redirect

from .models import Human

def index(request):

humans = Human.objects.all()

selected\_id = request.GET.get('human\_id')

selected = None

# Handle form submission to create new Human

if request.method == 'POST' and 'create' in request.POST:

first\_name = request.POST['first\_name']

last\_name = request.POST['last\_name']

phone = request.POST['phone']

address = request.POST['address']

city = request.POST['city']

Human.objects.create(

first\_name=first\_name,

last\_name=last\_name,

phone=phone,

address=address,

city=city

)

return redirect('/books/') # Refresh to show new entry

# Handle selection

if selected\_id:

selected = get\_object\_or\_404(Human, id=selected\_id)

return render(request, 'index.html', {

'humans': humans,

'selected': selected

})

def update\_human(request, human\_id):

human = get\_object\_or\_404(Human, id=human\_id)

if request.method == 'POST':

human.last\_name = request.POST['last\_name']

human.phone = request.POST['phone']

human.address = request.POST['address']

human.city = request.POST['city']

human.save()

return redirect('/')

def delete\_human(request, human\_id):

human = get\_object\_or\_404(Human, id=human\_id)

human.delete()

return redirect('/')

