Q1)

model.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=100)

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=200)

publication\_date = models.DateField()

authors = models.ManyToManyField(Author)

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

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

return self.title

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):

class Meta:

model = Book

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

views.py

from django.shortcuts import render, redirect

from .models import Author, Publisher, Book

from .forms import AuthorForm, PublisherForm, BookForm

def create\_author(request):

if request.method == 'POST':

form = AuthorForm(request.POST)

if form.is\_valid():

form.save()

return redirect('create\_author')

else:

form = AuthorForm()

return render(request, 'library/author\_form.html', {'form': form})

def create\_publisher(request):

if request.method == 'POST':

form = PublisherForm(request.POST)

if form.is\_valid():

form.save()

return redirect('create\_publisher')

else:

form = PublisherForm()

return render(request, 'library/publisher\_form.html', {'form': form})

def create\_book(request):

if request.method == 'POST':

form = BookForm(request.POST)

if form.is\_valid():

form.save()

return redirect('list\_books')

else:

form = BookForm()

return render(request, 'library/book\_form.html', {'form': form})

def list\_books(request):

books = Book.objects.all()

return render(request, 'library/book\_list.html', {'books': books})

author\_form.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Create Author</title>

</head>

<body>

<h1>Create Author</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Save Author</button>

</form>

<a href="{% url 'create\_publisher' %}">Create Publisher</a> |

<a href="{% url 'create\_book' %}">Create Book</a> |

<a href="{% url 'list\_books' %}">List Books</a>

</body>

</html>

publisher\_form.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Create Publisher</title>

</head>

<body>

<h1>Create Publisher</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Save Publisher</button>

</form>

<a href="{% url 'create\_author' %}">Create Author</a> |

<a href="{% url 'create\_book' %}">Create Book</a> |

<a href="{% url 'list\_books' %}">List Books</a>

</body>

</html>

book\_form.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Create Book</title>

</head>

<body>

<h1>Create Book</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Save Book</button>

</form>

<a href="{% url 'create\_author' %}">Create Author</a> |

<a href="{% url 'create\_publisher' %}">Create Publisher</a> |

<a href="{% url 'list\_books' %}">List Books</a>

</body>

</html>

book\_list.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Book List</title>

</head>

<body>

<h1>Book List</h1>

<ul>

{% for book in books %}

<li>{{ book.title }} (Published on: {{ book.publication\_date }})</li>

{% empty %}

<li>No books available.</li>

{% endfor %}

</ul>

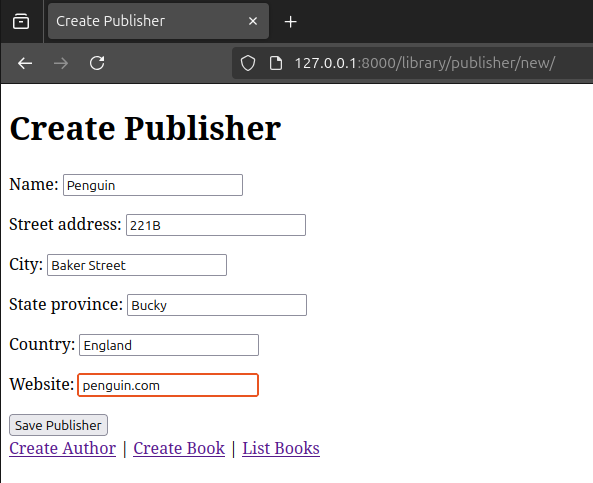
<a href="{% url 'create\_author' %}">Create Author</a> |

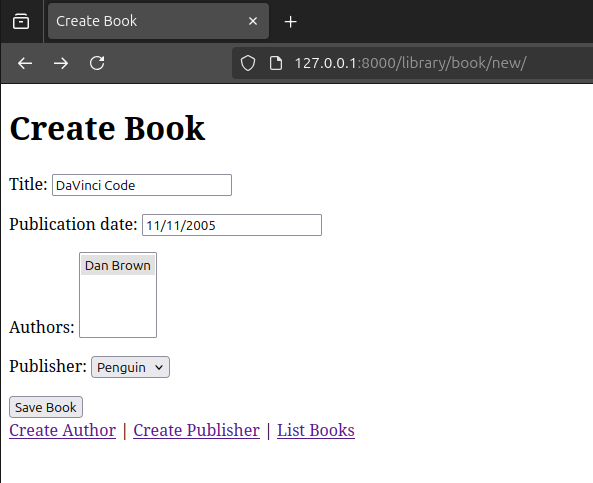
<a href="{% url 'create\_publisher' %}">Create Publisher</a> |

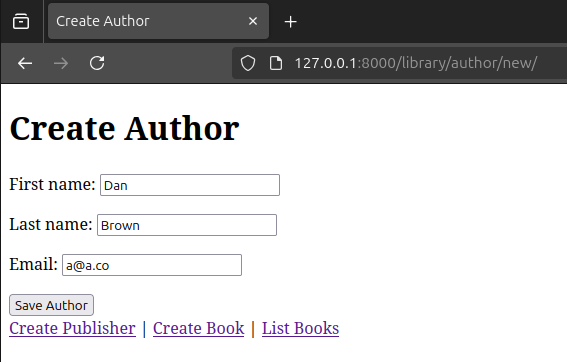
<a href="{% url 'create\_book' %}">Create Book</a>

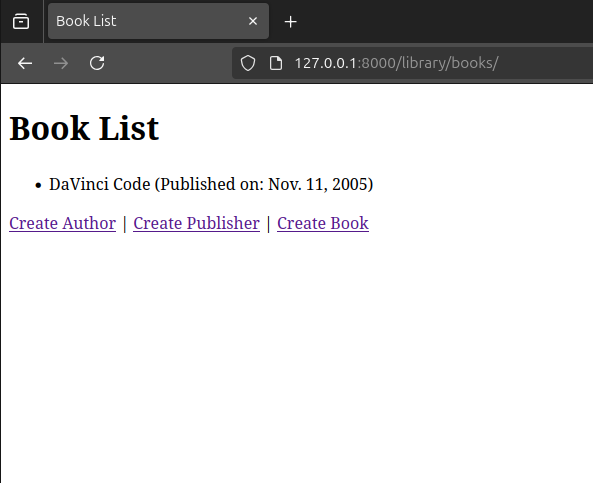
</body>

</html>









Q2

model.py

# store/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

forms.py

# store/forms.py

from django import forms

from .models import Product

class ProductForm(forms.ModelForm):

class Meta:

model = Product

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

views.py

# store/views.py

from django.shortcuts import render, redirect

from .models import Product

from .forms import ProductForm

def create\_product(request):

if request.method == 'POST':

form = ProductForm(request.POST)

if form.is\_valid():

form.save()

return redirect('product\_list')

else:

form = ProductForm()

return render(request, 'store/product\_form.html', {'form': form})

def product\_list(request):

products = Product.objects.all()

return render(request, 'store/product\_list.html', {'products': products})

product\_form.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Create Product</title>

</head>

<body>

<h1>Create Product</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

<a href="{% url 'product\_list' %}">View Products</a>

</body>

</html>

product\_list.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Product List</title>

</head>

<body>

<h1>Product List</h1>

<ul>

{% for product in products %}

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

{% empty %}

<li>No products available.</li>

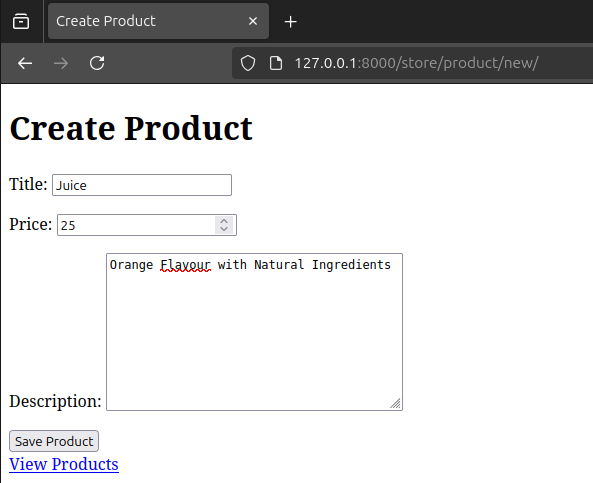
{% endfor %}

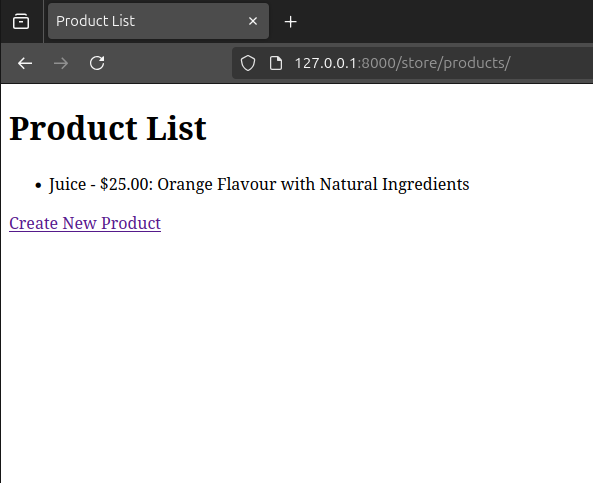
</ul>

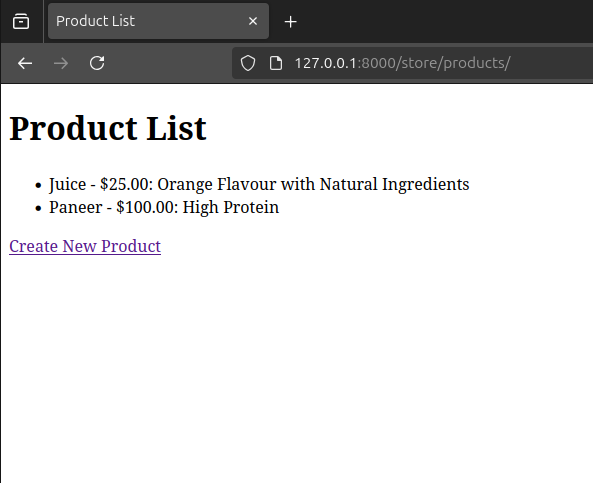
<a href="{% url 'create\_product' %}">Create New Product</a>

</body>

</html>







Q3)

models.py

# people/models.py

from django.db import models

class Human(models.Model):

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

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

phone = models.CharField(max\_length=15)

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

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

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

return self.first\_name

views.py

# people/views.py

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

from .models import Human

from .forms import HumanForm

def human\_management(request):

humans = Human.objects.all()

selected\_human = None

form = None

if request.method == 'POST':

if 'select\_human' in request.POST:

selected\_human\_id = request.POST.get('human')

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

form = HumanForm(instance=selected\_human)

elif 'update\_human' in request.POST:

selected\_human\_id = request.POST.get('human')

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

form = HumanForm(request.POST, instance=selected\_human)

if form.is\_valid():

form.save()

return redirect('human\_management')

elif 'delete\_human' in request.POST:

selected\_human\_id = request.POST.get('human')

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

selected\_human.delete()

return redirect('human\_management')

elif 'add\_human' in request.POST:

form = HumanForm(request.POST)

if form.is\_valid():

form.save()

return redirect('human\_management')

if selected\_human and not form:

form = HumanForm(instance=selected\_human)

return render(request, 'people/human\_management.html', {

'humans': humans,

'form': form,

'selected\_human': selected\_human,

})

forms.py

# people/forms.py

from django import forms

from .models import Human

class HumanForm(forms.ModelForm):

class Meta:

model = Human

fields = ['first\_name', 'last\_name', 'phone', 'address', 'city']

human\_management.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Human Management</title>

</head>

<body>

<h1>Human Management</h1>

<form method="post">

{% csrf\_token %}

<label for="human">Select Human:</label>

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

<option value="">--Select a Human--</option>

{% for human in humans %}

<option value="{{ human.id }}" {% if selected\_human and selected\_human.id == human.id %}selected{% endif %}>

{{ human.first\_name }}

</option>

{% endfor %}

</select>

<input type="submit" name="select\_human" value="Select">

</form>

{% if form %}

<h2>Edit Human Details</h2>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit" name="update\_human">Update</button>

<button type="submit" name="delete\_human">Delete</button>

</form>

{% endif %}

<h2>Add New Human</h2>

<form method="post" action="{% url 'human\_management' %}">

{% csrf\_token %}

<input type="text" name="first\_name" placeholder="First Name" required>

<input type="text" name="last\_name" placeholder="Last Name" required>

<input type="text" name="phone" placeholder="Phone" required>

<input type="text" name="address" placeholder="Address" required>

<input type="text" name="city" placeholder="City" required>

<button type="submit" name="add\_human">Add Human</button>

</form>

</body>

</html>



