WP Lab Week 11

Q1.There are three tables in the database an author table has a first name, a last name and an email address. A publisher table has a name, a street address, a city, a state/ province, a country, and a Web site. A book table has a title and a publication date. It also has one or more authors (a many-to-many relationship with authors) and a single publisher (a one-to-many relationship - aka foreign key - to publishers). Design a form which populates and retrieves the information from the above database using Django.

**add\_author.html**

<!DOCTYPE html>

<html>

<head>

<title>Add Author</title>

</head>

<body>

<h1>Add Author</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

<br>

<a href="{% url 'add\_publisher' %}">Add Publisher</a>

<br>

<a href="{% url 'add\_book' %}">Add Book</a>

<br>

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

</body>

</html>

**add\_book.html**

<!DOCTYPE html>

<html>

<head>

<title>Add Book</title>

</head>

<body>

<h1>Add Book</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

<br>

<a href="{% url 'add\_author' %}">Add Author</a>

<br>

<a href="{% url 'add\_publisher' %}">Add Publisher</a>

<br>

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

</body>

</html>

**add\_publisher.html**

<!DOCTYPE html>

<html>

<head>

<title>Add Publisher</title>

</head>

<body>

<h1>Add Publisher</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

<br>

<a href="{% url 'add\_author' %}">Add Author</a>

<br>

<a href="{% url 'add\_book' %}">Add Book</a>

<br>

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

</body>

</html>

**list\_books.html**

<!DOCTYPE html>

<html>

<head>

<title>List of Books</title>

</head>

<body>

<h1>List of Books</h1>

<ul>

{% for book in books %}

<li>{{ book.title }} by {% for author in book.authors.all %}{{ author }}{% endfor %} - {{ book.publisher.name }}</li>

{% endfor %}

</ul>

</body>

</html>

**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)

class Meta:

model = Book

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

**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(unique=True)

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

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

class Publisher(models.Model):

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

street\_address = models.CharField(max\_length=200)

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

**library/urls.py**

from django.urls import path

from . import views

urlpatterns = [

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

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

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

path('list\_books/', views.list\_books, name='list\_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, 'library/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')

else:

form = PublisherForm()

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

def list\_books(request):

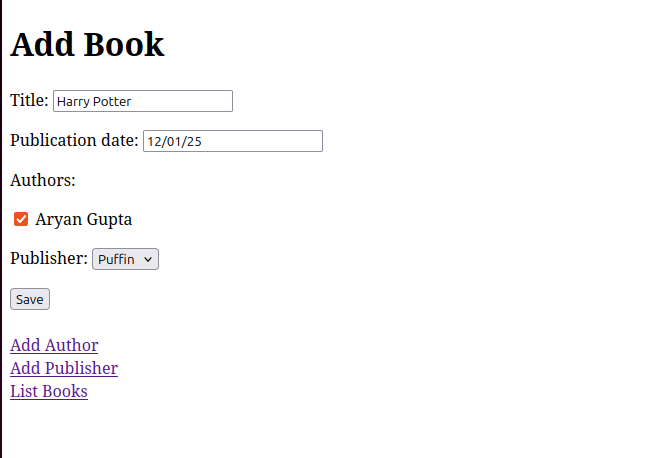
books = Book.objects.all()

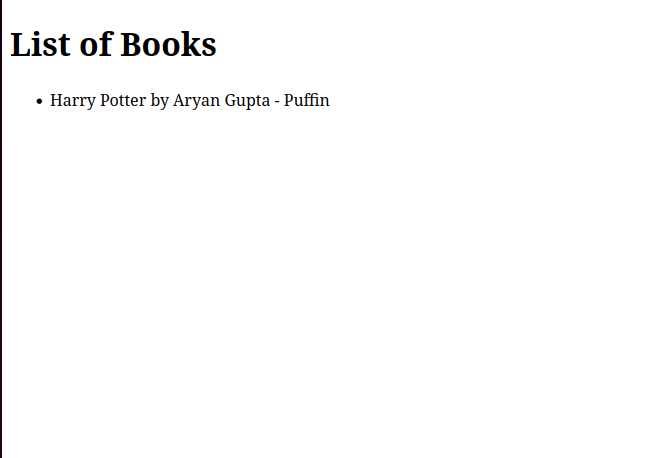
return render(request, 'library/list\_books.html', {'books': books})

**Output:**









Q2. Create a Django Page for entry of a Product information (title, price and description) and save it into the db. Create the index page where you would view the product entries in an unordered list.

**<!-- product/templates/product/add\_product.html -->**

<!DOCTYPE html>

<html>

<head>

<title>Add Product</title>

</head>

<body>

<h1>Add New Product</h1>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

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

</body>

</html>

**<!-- product/templates/product/index.html -->**

<!DOCTYPE html>

<html>

<head>

<title>Product List</title>

</head>

<body>

<h1>Products</h1>

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

<ul>

{% for product in products %}

<li>

{{ product.title }} - ${{ product.price }}

<p>{{ product.description }}</p>

</li>

{% empty %}

<li>No products yet.</li>

{% endfor %}

</ul>

</body>

</html>

**# product/admin.py**

from django.contrib import admin

from .models import Product

admin.site.register(Product)

**# product/forms.py**

from django import forms

from .models import Product

class ProductForm(forms.ModelForm):

class Meta:

model = Product

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

**# product/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()

created\_at = models.DateTimeField(auto\_now\_add=True)

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

return self.title

**# product/urls.py**

from django.urls import path

from . import views

urlpatterns = [

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

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

]

**# product/views.py**

from django.shortcuts import render, redirect

from .forms import ProductForm

from .models import Product

def index(request):

products = Product.objects.all()

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

def add\_product(request):

if request.method == 'POST':

form = ProductForm(request.POST)

if form.is\_valid():

form.save()

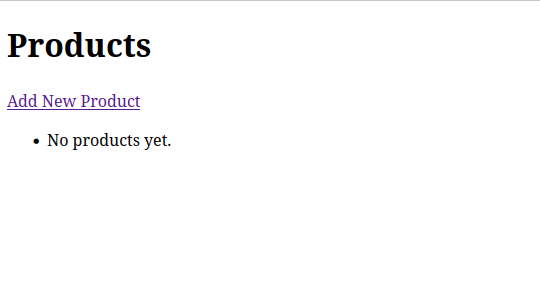
return redirect('index')

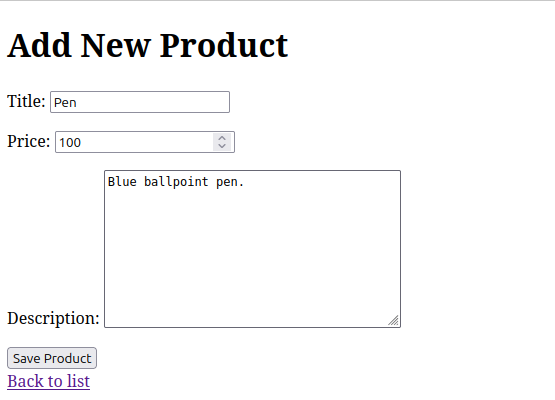
else:

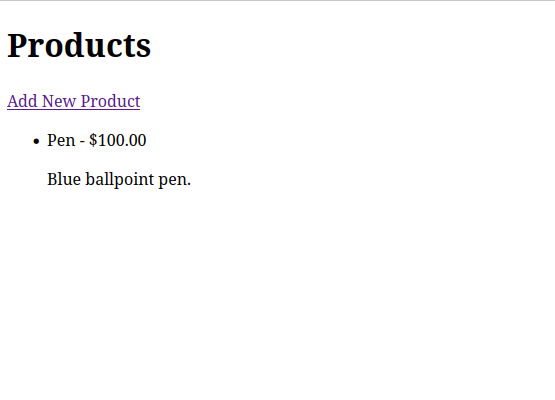
form = ProductForm()

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

**Output:**





Q3.Create a web page with DropDownList, Textboxes and Buttons. Assume the table ‘Human’ with First name, Last name, Phone, Address and City as fields. When the page is loaded, only first names will be displayed in the drop-down list. On selecting the name, other details will be displayed in the respective TextBoxes. On clicking the update button, the table will be updated with new entries made in the text box. On clicking the delete button, the selected record will be deleted from the table, and the DropDownList is refreshed.

**<!-- human\_manager/templates/human\_manager/index.html -->**

<!DOCTYPE html>

<html>

<head>

<title>Human Manager</title>

<style>

.form-container { margin-top: 20px; }

.field { margin-bottom: 10px; }

</style>

</head>

<body>

<h1>Human Manager</h1>

<form method="post">

{% csrf\_token %}

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

<option value="">Select a person</option>

{% for human in humans %}

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

{{ human.first\_name }}

</option>

{% endfor %}

</select>

</form>

<div class="form-container">

<form method="post">

{% csrf\_token %}

<input type="hidden" name="human\_id" value="{{ selected\_human.id }}">

<div class="field">

<label>First Name:</label><br>

<input type="text" name="first\_name" value="{{ form.first\_name.value|default:'' }}">

</div>

<div class="field">

<label>Last Name:</label><br>

<input type="text" name="last\_name" value="{{ form.last\_name.value|default:'' }}">

</div>

<div class="field">

<label>Phone:</label><br>

<input type="text" name="phone" value="{{ form.phone.value|default:'' }}">

</div>

<div class="field">

<label>Address:</label><br>

<textarea name="address">{{ form.address.value|default:'' }}</textarea>

</div>

<div class="field">

<label>City:</label><br>

<input type="text" name="city" value="{{ form.city.value|default:'' }}">

</div>

{% if selected\_human %}

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

<button type="submit" name="delete" onclick="return confirm('Are you sure?')">Delete</button>

{% endif %}

</form>

</div>

</body>

</html>

**# human\_manager/admin.py**

from django.contrib import admin

from .models import Human

admin.site.register(Human)

**# human\_manager/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\_manager/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.TextField()

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

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

return self.first\_name

**# human\_manager/urls.py**

from django.urls import path

from . import views

urlpatterns = [

path('', views.human\_manager, name='human\_manager'),

]

**# human\_manager/views.py**

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

from .models import Human

from .forms import HumanForm

def human\_manager(request):

humans = Human.objects.all()

selected\_human = None

form = HumanForm()

if request.method == 'POST':

if 'update' in request.POST:

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

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

form = HumanForm(request.POST, instance=human)

if form.is\_valid():

form.save()

return redirect('human\_manager')

elif 'delete' in request.POST:

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

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

human.delete()

return redirect('human\_manager')

elif 'human\_id' in request.POST:

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

if human\_id:

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

form = HumanForm(instance=selected\_human)

return render(request, 'human\_manager/index.html', {

'humans': humans,

'form': form,

'selected\_human': selected\_human

})

**Output:**

