

AYANIKA PAUL

220905128

LAB 10

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.

CODE:

q1/urls.py

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

library/urls.py

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

library.views.py

```
from django.shortcuts import render, redirect
from .models import Author, Publisher, Book
from .forms import AuthorForm, PublisherForm, BookForm
def add_author(request):
    if request.method == 'POST':
        form = AuthorForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('index')
    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('index')
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('index')
else:
form = BookForm()
return render(request, 'library/add_book.html', {'form': form})
def index(request):
books = Book.objects.all()
return render(request, 'library/index.html', {'books': books})

```

library/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

```

add_book.html

```
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Add Book</button>
</form>
```

add_author.html

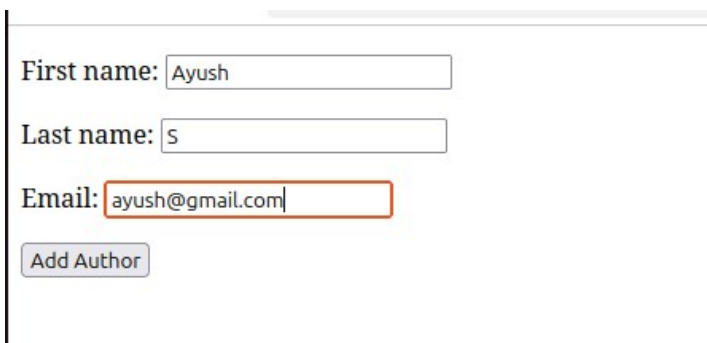
```
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Add Author</button>
</form>
```

index.html

```
<h1>Books</h1>
<ul>
{% for book in books %}
<li>{{ book.title }} ({{ book.publication_date }}) - Publisher: {{ book.publisher }} </li>
{% endfor %}
</ul>
<a href="{% url 'add_author' %}">Add Author</a> |
<a href="{% url 'add_publisher' %}">Add Publisher</a> |
<a href="{% url 'add_book' %}">Add Book</a>
```

add_publisher.html

```
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Add Publisher</button>
</form>
```



First name:

Last name:

Email:

Books

- Fundamentals of C (Feb. 24, 2024) - Publisher: Arihant Publications

[Add Author](#) | [Add Publisher](#) | [Add Book](#)

Name:

Street address:

City:

State province:

Country:

Website:

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change
LIBRARY		
Authors	+ Add	✎ Change
Books	+ Add	✎ Change
Publishers	+ Add	✎ Change

Recent actions

My actions

None available

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.

CODE:

q2/urls.py

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

products/urls.py

```

from django.urls import path
from .views import add_product, index
urlpatterns = [
    path("", index, name='index'),
    path('add/', add_product, name='add_product'),
]

```

products/views.py

```

from django.shortcuts import render, redirect
from .models import Product
from .forms import ProductForm
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})
def index(request):
    products = Product.objects.all()
    return render(request, 'products/index.html', {'products': products})

```

products/models.py

```

from django.db import models
class Product(models.Model):
    title = models.CharField(max_length=255)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    description = models.TextField()
    def __str__(self):
        return self.title
products/forms.py
from django import forms
from .models import Product
class ProductForm(forms.ModelForm):
    class Meta:
        model = Product
        fields = ['title', 'price', 'description']

```

Add Product

Title:

Price:

Description:

FOLD Phone

[View Products](#)

Select product to change

ADD PRODUCT +

Action: Go 0 of 3 selected☐ PRODUCT☐ Samsung☐ Ativa 7g☐ Iphone

3 products

Product List

- Iphone - \$100000.00 - very expensive
- Ativa 7g - \$105000.00 - only for rich people
- Samsung - \$1890000.00 - FOLD Phone

[Add Product](#)

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.

CODE:

q1/urls.py

```
from django.contrib import admin
```

```

from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('human.urls')),
]

```

human/urls.py

```

from django.urls import path
from . import views
urlpatterns = [
    path("", views.index, name='index'),
    path('get-details/<int:pk>/', views.get_details, name='get-details'),
    path('update/<int:pk>/', views.update_human, name='update-human'),
    path('delete/<int:pk>/', views.delete_human, name='delete-human'),
]

```

human/viewspy

```

from django.shortcuts import render, get_object_or_404
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from .models import Human
def index(request):
    humans = Human.objects.all()
    return render(request, 'human/index.html', {'humans': humans})
def get_details(request, pk):
    try:
        human = get_object_or_404(Human, pk=pk)
        data = {
            'first_name': human.first_name,
            'last_name': human.last_name,
            'phone': human.phone,
            'address': human.address,
            'city': human.city,
        }
        return JsonResponse(data)
    except Exception as e:
        return JsonResponse({'error': str(e)}, status=400)
@csrf_exempt
def update_human(request, pk):
    if request.method == 'POST':
        try:
            human = get_object_or_404(Human, pk=pk)
            human.first_name = request.POST.get('first_name')
            human.last_name = request.POST.get('last_name')
            human.phone = request.POST.get('phone')
            human.address = request.POST.get('address')
            human.city = request.POST.get('city')
            human.save()
            return JsonResponse({'message': 'Human updated successfully!'})

```

```

except Exception as e:
return JsonResponse({'error': str(e)}, status=400)
return JsonResponse({'error': 'Invalid request method.'}, status=405)
@csrf_exempt
def delete_human(request, pk):
if request.method == 'POST':
try:
human = get_object_or_404(Human, pk=pk)
human.delete()
return JsonResponse({'message': 'Human deleted successfully!'})
except Exception as e:
return JsonResponse({'error': str(e)}, status=400)
return JsonResponse({'error': 'Invalid request method.'}, status=405)

```

human/models.py

```

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=15)
address = models.TextField()
city = models.CharField(max_length=50)
def __str__(self):
return self.first_name

```

human/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']
widgets = {
'first_name': forms.TextInput(attrs={'placeholder': 'First Name'}),
'last_name': forms.TextInput(attrs={'placeholder': 'Last Name'}),
'phone': forms.TextInput(attrs={'placeholder': 'Phone'}),
'address': forms.TextInput(attrs={'placeholder': 'Address'}),
'city': forms.TextInput(attrs={'placeholder': 'City'}),
}

```

index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

```



```

<title>Human Records</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
<h1>Manage Human Records</h1>
<label for="dropdown">Select First Name:</label>
<select id="dropdown">
<option value="" disabled selected>Select...</option>
{% for human in humans %}
<option value="{{ human.id }}">{{ human.first_name }}</option>
{% endfor %}
</select>
<form id="human-form">
{% csrf_token %}
<input type="text" id="first_name" name="first_name" placeholder="First Name">
<input type="text" id="last_name" name="last_name" placeholder="Last Name">
<input type="text" id="phone" name="phone" placeholder="Phone">
<input type="text" id="address" name="address" placeholder="Address">
<input type="text" id="city" name="city" placeholder="City">
<button type="button" id="update-button">Update</button>
<button type="button" id="delete-button">Delete</button>
</form>
<script>
$(document).ready(function() {
$('#dropdown').change(function() {
const id = $(this).val();
if (id) {
$.get(`/get-details/${id}/`, function(data) {
$('#first_name').val(data.first_name);
$('#last_name').val(data.last_name);
$('#phone').val(data.phone);
$('#address').val(data.address);
$('#city').val(data.city);
});
});
$('#update-button').click(function() {
const id = $('#dropdown').val();
const formData = $('#human-form').serialize();
if (id) {
$.post(`/update/${id}/`, formData, function(response) {
alert(response.message);
location.reload();
}).fail(function(error) {
alert('Error: ' + error.responseJSON.error);
});
});
$('#delete-button').click(function() {
const id = $('#dropdown').val();

```

```
if (id && confirm('Are you sure you want to delete this record?')) {  
$.post(`/delete/${id}/`, function(response) {  
alert(response.message);  
location.reload();  
}).fail(function(error) {  
alert('Error: ' + error.responseJSON.error);  
});  
}  
});  
});  
</script>  
</body>  
</html>
```

Manage Human Records

Select First Name:

<input type="text" value="Ayush"/>	<input type="text" value="Das"/>	<input type="text" value="8908764321"/>	<input type="text" value="Eshwar Nagar"/>
<input type="text" value="Manipal"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>	

🌐 127.0.0.1:8000

Human updated successfully!

OK

Manage Human Records

Select First Name:

Select... v

First Name

City

Select...

Ayush

Rahul

Anuja


Last Name

ate

Delete

Phone

Address

WELCOME, **AYANIKA**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#) 

Home > Human > Humans

Select human to change

ADD HUMAN +

Action:

----- v

Go

 0 of 3 selected

<input type="checkbox"/>	FIRST NAME	LAST NAME	PHONE	ADDRESS	CITY
<input type="checkbox"/>	Anuja	S	9802287651	XYZ Road	Manipal
<input type="checkbox"/>	Rahul	Kumar	8809765431	MG Road	Bangalore
<input type="checkbox"/>	Ayush	Das	8908764321	Eshwar Nagar	Manipal

3 humans

