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:

webapp/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=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

webapp/forms.py:

from django import forms

from .models import Author, Publisher, Book

class AuthorForm(forms.ModelForm):

class Meta:

model = Author

fields = '\_\_all\_\_'

class PublisherForm(forms.ModelForm):

class Meta:

model = Publisher

fields = '\_\_all\_\_'

class BookForm(forms.ModelForm):

class Meta:

model = Book

fields = '\_\_all\_\_'

widgets = {

'publication\_date': forms.DateInput(attrs={'type': 'date'}),

}

webapp/views.py:

from django.shortcuts import render, redirect

from .forms import AuthorForm, PublisherForm, BookForm

from .models import Author, Publisher, Book

def main(request):

authors = Author.objects.all()

publishers = Publisher.objects.all()

books = Book.objects.all()

if request.method == 'POST':

author\_form = AuthorForm(request.POST, prefix='author')

publisher\_form = PublisherForm(request.POST, prefix='publisher')

book\_form = BookForm(request.POST, prefix='book')

if author\_form.is\_valid():

author\_form.save()

return redirect('main')

if publisher\_form.is\_valid():

publisher\_form.save()

return redirect('main')

if book\_form.is\_valid():

book\_form.save()

return redirect('main')

else:

author\_form = AuthorForm(prefix='author')

publisher\_form = PublisherForm(prefix='publisher')

book\_form = BookForm(prefix='book')

context = {

'authors': authors,

'publishers': publishers,

'books': books,

'author\_form': author\_form,

'publisher\_form': publisher\_form,

'book\_form': book\_form,

}

return render(request, 'main.html', context)

webapp/templates/main.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Author Publisher Book</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-4">

<h2>Authors</h2>

<table class="table table-bordered">

<thead>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

{% for author in authors %}

<tr>

<td>{{ author.first\_name }}</td>

<td>{{ author.last\_name }}</td>

<td>{{ author.email }}</td>

</tr>

{% endfor %}

</tbody>

</table>

<h2>Publishers</h2>

<table class="table table-bordered">

<thead>

<tr>

<th>Name</th>

<th>Street Address</th>

<th>City</th>

<th>State/Province</th>

<th>Country</th>

<th>Website</th>

</tr>

</thead>

<tbody>

{% for publisher in publishers %}

<tr>

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

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

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

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

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

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

</tr>

{% endfor %}

</tbody>

</table>

<h2>Books</h2>

<table class="table table-bordered">

<thead>

<tr>

<th>Title</th>

<th>Publication Date</th>

<th>Authors</th>

<th>Publisher</th>

</tr>

</thead>

<tbody>

{% for book in books %}

<tr>

<td>{{ book.title }}</td>

<td>{{ book.publication\_date }}</td>

<td>

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

{{ author.first\_name }} {{ author.last\_name }}{% if not forloop.last %}, {% endif %}

{% endfor %}

</td>

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

</tr>

{% endfor %}

</tbody>

</table>

<div class="mt-4">

<h2>Add New Author</h2>

<form method="post">

{% csrf\_token %}

{{ author\_form.as\_p }}

<button type="submit" class="btn btn-primary">Add Author</button>

</form>

</div>

<div class="mt-4">

<h2>Add New Publisher</h2>

<form method="post">

{% csrf\_token %}

{{ publisher\_form.as\_p }}

<button type="submit" class="btn btn-primary">Add Publisher</button>

</form>

</div>

<div class="mt-4">

<h2>Add New Book</h2>

<form method="post">

{% csrf\_token %}

{{ book\_form.as\_p }}

<button type="submit" class="btn btn-primary">Add Book</button>

</form>

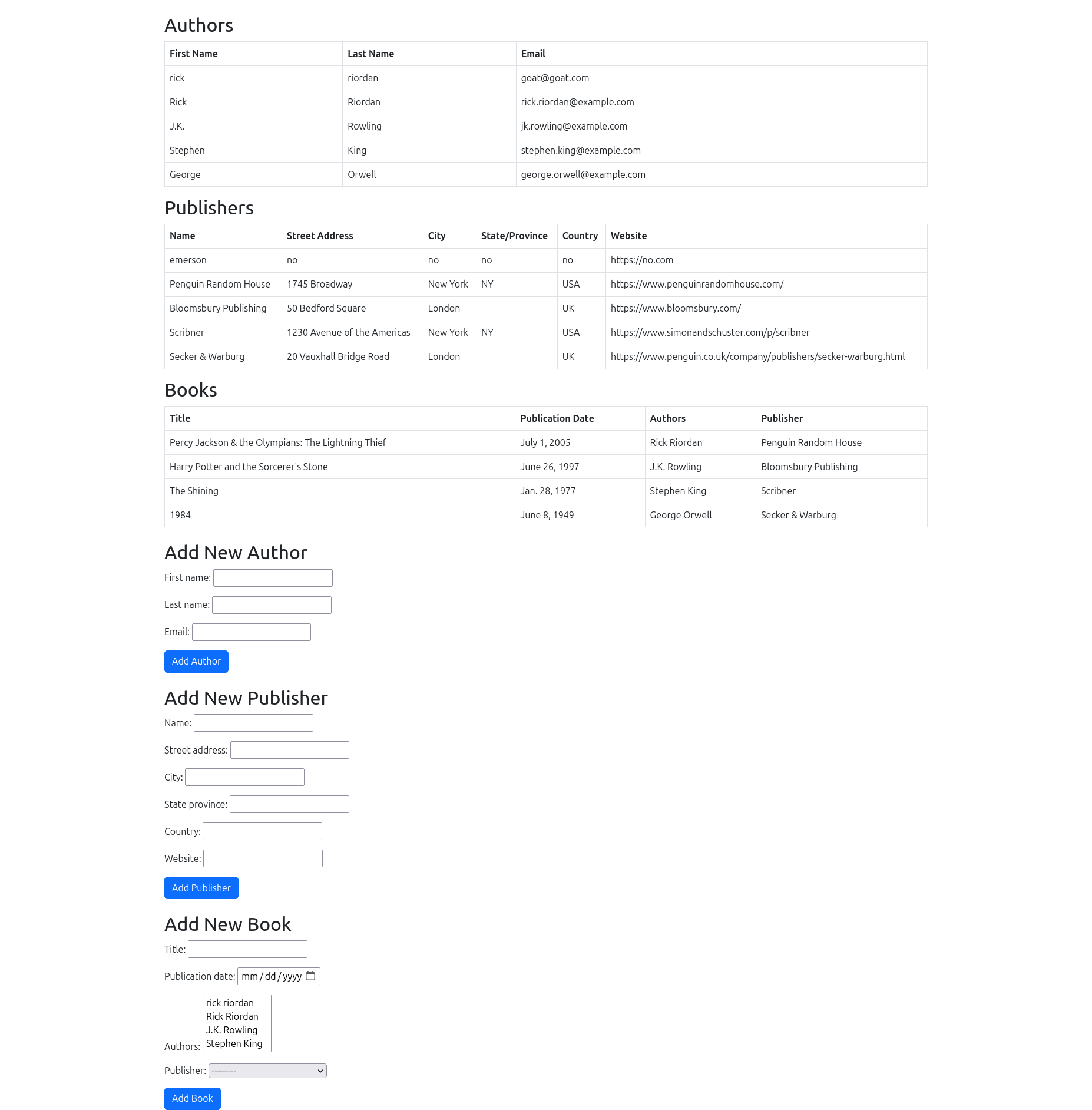
</div>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

Image:

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:

webapp/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

webapp/forms.py:

from django import forms

from .models import Product

class ProductForm(forms.ModelForm):

class Meta:

model = Product

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

webapp/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, '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, 'index.html', {'form': form})

webapp/templates/index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Product Info</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

</head>

<body>

<div class="container mt-5">

<h1>Product Information</h1>

{% if form %}

<form method="post" class="mt-4">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit" class="btn btn-primary">Save Product</button>

</form>

{% endif %}

<h2 class="mt-4">Product List</h2>

<ul>

{% for product in products %}

<li>

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

{{ product.description }}

</li>

{% empty %}

<li>No products available.</li>

{% endfor %}

</ul>

<a href="{% url 'add\_product' %}" class="btn btn-success mt-3">Add Product</a>

</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

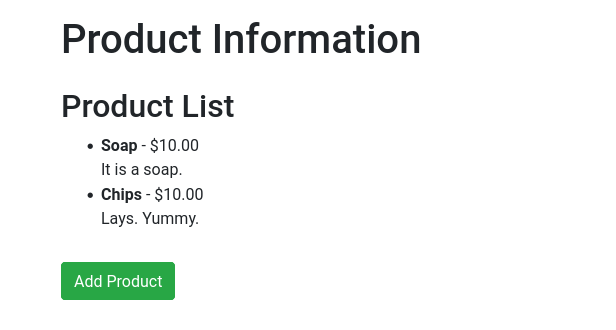
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

Image:



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:

webapp/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=20)

address = models.TextField()

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

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

return self.first\_name

webapp/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']

webapp/views.py:

from django.shortcuts import render, redirect

from .models import Human

from .forms import HumanForm

from django.http import JsonResponse

from django.views.decorators.csrf import csrf\_exempt

def index(request):

humans = Human.objects.all()

form = HumanForm()

return render(request, 'index.html', {'humans': humans, 'form': form})

@csrf\_exempt

def update\_human(request):

if request.method == 'POST':

form = HumanForm(request.POST)

if form.is\_valid():

form.save()

return redirect('index')

return JsonResponse({'status': 'error'})

@csrf\_exempt

def delete\_human(request):

if request.method == 'POST':

first\_name = request.POST.get('first\_name')

Human.objects.filter(first\_name=first\_name).delete()

return redirect('index')

return JsonResponse({'status': 'error'})

@csrf\_exempt

def get\_human\_data(request):

if request.method == 'POST':

first\_name = request.POST.get('first\_name')

try:

human = Human.objects.get(first\_name=first\_name)

data = {

'last\_name': human.last\_name,

'phone': human.phone,

'address': human.address,

'city': human.city,

}

return JsonResponse(data)

except Human.DoesNotExist:

return JsonResponse({})

return JsonResponse({'status': 'error'})

webapp/templates/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 Data</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

</head>

<body>

<div class="container mt-5">

<form method="post" action="{% url 'update\_human' %}">

{% csrf\_token %}

<div class="form-group">

<label for="id\_first\_name">First Name:</label>

<select class="form-control" name="first\_name" id="id\_first\_name">

{% for human in humans %}

<option value="{{ human.first\_name }}">{{ human.first\_name }}</option>

{% endfor %}

</select>

</div>

<div class="form-group">

<label for="id\_last\_name">Last Name:</label>

<input type="text" class="form-control" name="last\_name" id="id\_last\_name">

</div>

<div class="form-group">

<label for="id\_phone">Phone:</label>

<input type="text" class="form-control" name="phone" id="id\_phone">

</div>

<div class="form-group">

<label for="id\_address">Address:</label>

<textarea class="form-control" name="address" id="id\_address"></textarea>

</div>

<div class="form-group">

<label for="id\_city">City:</label>

<input type="text" class="form-control" name="city" id="id\_city">

</div>

<button type="submit" class="btn btn-primary">Update</button>

</form>

<form method="post" action="{% url 'delete\_human' %}">

{% csrf\_token %}

<input type="hidden" name="first\_name" id="delete\_first\_name">

<button type="submit" class="btn btn-danger mt-3">Delete</button>

</form>

</div>

<script>

document.getElementById('id\_first\_name').addEventListener('change', function() {

var selectedFirstName = this.value;

fetch('/get\_human\_data/', {

method: 'POST',

headers: {

'Content-Type': 'application/x-www-form-urlencoded',

'X-CSRFToken': '{{ csrf\_token }}',

},

body: 'first\_name=' + selectedFirstName,

})

.then(response => response.json())

.then(data => {

document.getElementById('id\_last\_name').value = data.last\_name;

document.getElementById('id\_phone').value = data.phone;

document.getElementById('id\_address').value = data.address;

document.getElementById('id\_city').value = data.city;

document.getElementById('delete\_first\_name').value = selectedFirstName;

});

});

</script>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

Image:

