220962446 – Lab11

Question 1. 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 1.

views.py

from django.shortcuts import render, redirect

from .models import Author, Publisher, Book

from .forms import BookForm

from django.db.models import Q

def manage\_books(request):

query = request.GET.get('query', '')

books = None

if query:

books = Book.objects.filter(

Q(title\_\_icontains=query) |

Q(authors\_\_first\_name\_\_icontains=query) |

Q(authors\_\_last\_name\_\_icontains=query)

).distinct()

return render(request, 'base.html', {'books': books, 'query': query, 'content': 'manage\_books'})

def add\_book(request):

if request.method == "POST":

title = request.POST.get('title')

publication\_date = request.POST.get('publication\_date')

authors = request.POST.getlist('authors')

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

book = Book(title=title, publication\_date=publication\_date)

publisher = Publisher.objects.get(id=publisher\_id)

book.publisher = publisher

book.save()

for author\_id in authors:

author = Author.objects.get(id=author\_id)

book.authors.add(author)

return redirect('manage\_books')

authors = Author.objects.all()

publishers = Publisher.objects.all()

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

'content': 'add\_book',

'authors': authors,

'publishers': publishers

})

def add\_author(request):

if request.method == "POST":

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

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

email = request.POST['email']

Author.objects.create(first\_name=first\_name, last\_name=last\_name, email=email)

return redirect('manage\_books')

return render(request, 'base.html', {'content': 'add\_author'})

def add\_publisher(request):

if request.method == "POST":

name = request.POST['name']

street\_address = request.POST['street\_address']

city = request.POST['city']

state\_province = request.POST['state\_province']

country = request.POST['country']

website = request.POST['website']

Publisher.objects.create(name=name, street\_address=street\_address,

city=city, state\_province=state\_province,

country=country, website=website)

return redirect('manage\_books')

return render(request, 'base.html', {'content': 'add\_publisher'})

forms.py

from django import forms

from .models import Book

class BookForm(forms.ModelForm):

class Meta:

model = Book

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

urls.py

from django.urls import path

from . import views

urlpatterns = [

path('books/', views.manage\_books, name='manage\_books'), # Main management page for searching books

path('add-book/', views.add\_book, name='add\_book'), # URL to add a new book

path('add-author/', views.add\_author, name='add\_author'), # URL to add a new author

path('add-publisher/', views.add\_publisher, name='add\_publisher'), # URL to add a new 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()

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

admin.py

from django.contrib import admin

from .models import Author, Publisher, Book

admin.site.register(Author)

admin.site.register(Publisher)

admin.site.register(Book)

templates/base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Book Management</title>

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

</head>

<body class="bg-light">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container">

<a class="navbar-brand" href="{% url 'manage\_books' %}">Book System</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<div class="navbar-nav">

<a class="nav-link" href="{% url 'manage\_books' %}">Manage Books</a>

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

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

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

</div>

</div>

</div>

</nav>

<main class="container mt-4">

<div class="card shadow">

<div class="card-body">

{% if content == 'manage\_books' %}

<div class="mb-4">

<h2 class="mb-4">Manage Books</h2>

<form method="GET" class="row g-3">

<div class="col-md-8">

<input type="text" name="query" class="form-control"

placeholder="Search by title or author" required>

</div>

<div class="col-md-4">

<button type="submit" class="btn btn-primary w-100">Search</button>

</div>

</form>

</div>

{% if books %}

<div class="mt-4">

<h3 class="mb-3">Search Results</h3>

<div class="list-group">

{% for book in books %}

<div class="list-group-item">

<h5 class="mb-2">{{ book.title }}</h5>

<div class="d-flex justify-content-between text-muted small">

<span>Published: {{ book.publication\_date }}</span>

<span>Publisher: {{ book.publisher.name }}</span>

</div>

<div class="mt-2">

<strong>Authors:</strong>

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

<span class="badge bg-secondary">

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

</span>

{% endfor %}

</div>

</div>

{% endfor %}

</div>

</div>

{% elif query %}

<div class="alert alert-warning mt-4">No results found for "{{ query }}"</div>

{% endif %}

{% elif content == 'add\_book' %}

<h2 class="mb-4">Add New Book</h2>

<form method="POST" class="row g-3">

{% csrf\_token %}

<div class="col-md-6">

<label class="form-label">Title</label>

<input type="text" name="title" class="form-control" required>

</div>

<div class="col-md-6">

<label class="form-label">Publication Date</label>

<input type="date" name="publication\_date" class="form-control" required>

</div>

<div class="col-md-6">

<label class="form-label">Authors</label>

<select multiple class="form-select" name="authors" required>

{% for author in authors %}

<option value="{{ author.id }}">{{ author.first\_name }} {{ author.last\_name }}</option>

{% endfor %}

</select>

</div>

<div class="col-md-6">

<label class="form-label">Publisher</label>

<select class="form-select" name="publisher" required>

{% for publisher in publishers %}

<option value="{{ publisher.id }}">{{ publisher.name }}</option>

{% endfor %}

</select>

</div>

<div class="col-12">

<button type="submit" class="btn btn-success">Save Book</button>

</div>

</form>

{% elif content == 'add\_author' %}

<h2 class="mb-4">Add New Author</h2>

<form method="POST" class="row g-3">

{% csrf\_token %}

<div class="col-md-4">

<label class="form-label">First Name</label>

<input type="text" name="first\_name" class="form-control" required>

</div>

<div class="col-md-4">

<label class="form-label">Last Name</label>

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

</div>

<div class="col-md-4">

<label class="form-label">Email</label>

<input type="email" name="email" class="form-control" required>

</div>

<div class="col-12">

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

</div>

</form>

{% elif content == 'add\_publisher' %}

<h2 class="mb-4">Add New Publisher</h2>

<form method="POST" class="row g-3">

{% csrf\_token %}

<!-- Form fields remain same but styled -->

<div class="col-md-6">

<label class="form-label">Name</label>

<input type="text" name="name" class="form-control" required>

</div>

<div class="col-md-6">

<label class="form-label">Website</label>

<input type="url" name="website" class="form-control">

</div>

<div class="col-12">

<label class="form-label">Street Address</label>

<input type="text" name="street\_address" class="form-control" required>

</div>

<div class="col-md-4">

<label class="form-label">City</label>

<input type="text" name="city" class="form-control" required>

</div>

<div class="col-md-4">

<label class="form-label">State/Province</label>

<input type="text" name="state\_province" class="form-control">

</div>

<div class="col-md-4">

<label class="form-label">Country</label>

<input type="text" name="country" class="form-control">

</div>

<div class="col-12">

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

</div>

</form>

{% endif %}

</div>

</div>

</main>

<footer class="mt-4 text-center text-muted small">

<p>&copy; 2023 Book Management System</p>

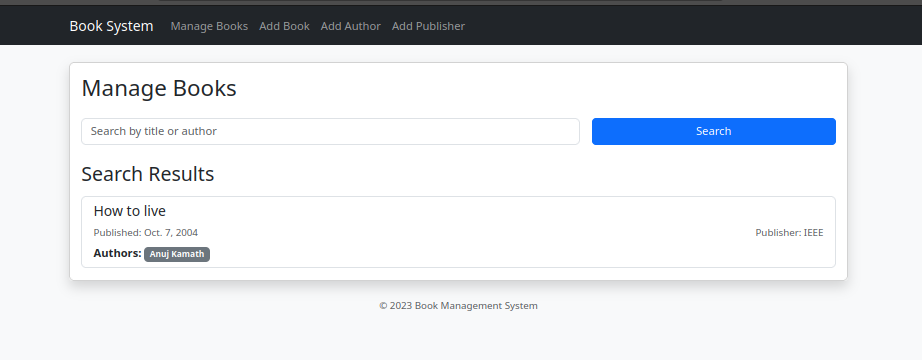
</footer>

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

</body>

</html>

Output 1)



Question 2. 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 2.

views.py

# webapp/views.py

from django.shortcuts import render, redirect

from .forms import ProductForm

from .models import Product

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, 'create\_product.html', {'form': form})

def product\_list(request):

products = Product.objects.all()

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

forms.py

# webapp/forms.py

from django import forms

from .models import Product

class ProductForm(forms.ModelForm):

class Meta:

model = Product

fields = ('title', 'price', 'description')

urls.py

# webapp/urls.py

from django.urls import path

from .views import create\_product, product\_list

urlpatterns = [

path('', product\_list, name='product\_list'),

path('create/', create\_product, name='create\_product'),

]

models.py

# webapp/models.py

from django.db import models

class Product(models.Model):

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

price = models.DecimalField(max\_digits=8, decimal\_places=2)

description = models.TextField()

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

return self.title

admin.py

# products/admin.py

from django.contrib import admin

from .models import Product

admin.site.register(Product)

templates/base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Product App</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

</head>

<body>

<div class="container mt-5">

{% block content %}{% endblock %}

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

</body>

</html>

templates/create\_product.html

{% extends 'base.html' %}

{% block content %}

<h1 class="mb-3">Create Product</h1>

<form method="post">

{% csrf\_token %}

<div class="mb-3">

{{ form.title.label\_tag }} {{ form.title }}

</div>

<div class="mb-3">

{{ form.price.label\_tag }} {{ form.price }}

</div>

<div class="mb-3">

{{ form.description.label\_tag }} {{ form.description }}

</div>

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

</form>

{% endblock %}

templates/product\_list.html

{% extends 'base.html' %}

{% block content %}

<h1 class="mb-3">Product List</h1>

<ul class="list-group">

{% for product in products %}

<li class="list-group-item">

<span>{{ product.title }} ({{ product.price }})</span>

<br>

{{ product.description }}

</li>

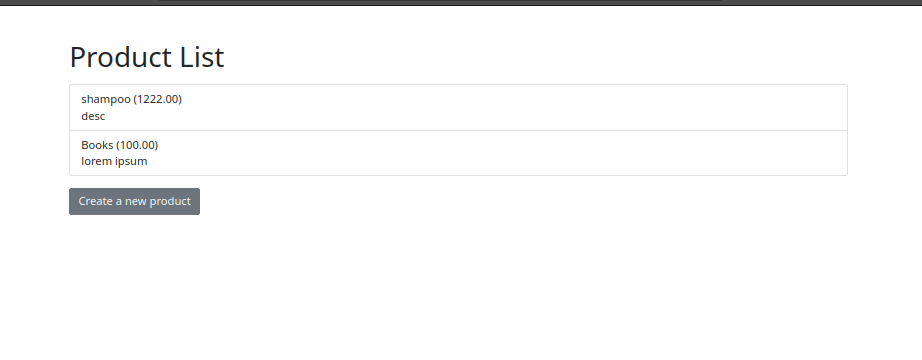
{% endfor %}

</ul>

<p><a href="{% url 'create\_product' %}" class="btn btn-secondary mt-3">Create a new product</a></p>

{% endblock %}

Output 2.



**Question 3. 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 3.

views.py

# webapp/views.py

from django.shortcuts import render, redirect

from .models import Human

from django.http import JsonResponse

def human\_list(request):

humans = Human.objects.all()

first\_names = [h.first\_name for h in humans]

return render(request, 'human\_list.html', {'first\_names': first\_names})

def get\_human\_details(request, first\_name):

try:

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

return JsonResponse({

'first\_name': human.first\_name,

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

'phone': human.phone,

'address': human.address,

'city': human.city,

})

except Human.DoesNotExist:

return JsonResponse({})

def update\_human(request, first\_name):

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

if request.method == 'POST':

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 redirect('human\_list')

def delete\_human(request, first\_name):

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

human.delete()

return redirect('human\_list')

forms.py

# 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')

urls.py

from django.urls import path

from .views import human\_list, get\_human\_details, update\_human, delete\_human

urlpatterns = [

path('', human\_list, name='human\_list'),

path('get\_details/<str:first\_name>/', get\_human\_details, name='get\_human\_details'),

path('update/<str:first\_name>/', update\_human, name='update\_human'),

path('delete/<str:first\_name>/', delete\_human, name='delete\_human'),

]

models.py

# 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.CharField(max\_length=200)

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

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

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

templates/human\_list.html

<!-- webapp/templates/human\_list.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Human Details</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

</head>

<body>

<div class="container mt-5">

<h1>Human Details</h1>

<select id="humanSelect" class="form-select">

{% for name in first\_names %}

<option value="{{ name }}">{{ name }}</option>

{% endfor %}

</select>

<form id="humanForm">

<div class="mb-3">

<label for="first\_name" class="form-label">First Name</label>

<input type="text" class="form-control" id="first\_name" disabled>

</div>

<div class="mb-3">

<label for="last\_name" class="form-label">Last Name</label>

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

</div>

<div class="mb-3">

<label for="phone" class="form-label">Phone</label>

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

</div>

<div class="mb-3">

<label for="address" class="form-label">Address</label>

<input type="text" class="form-control" id="address" name="address">

</div>

<div class="mb-3">

<label for="city" class="form-label">City</label>

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

</div>

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

<button type="button" class="btn btn-danger" id="deleteBtn">Delete</button>

</form>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>

<script>

document.addEventListener('DOMContentLoaded', function() {

const humanSelect = document.getElementById('humanSelect');

const humanForm = document.getElementById('humanForm');

humanSelect.addEventListener('change', function() {

const first\_name = this.value;

fetch('/get\_details/' + first\_name)

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

.then(data => {

document.getElementById('first\_name').value = data.first\_name;

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

document.getElementById('phone').value = data.phone || '';

document.getElementById('address').value = data.address || '';

document.getElementById('city').value = data.city || '';

});

});

humanForm.addEventListener('submit', function(event) {

event.preventDefault();

const first\_name = document.getElementById('first\_name').value;

const last\_name = document.getElementById('last\_name').value;

const phone = document.getElementById('phone').value;

const address = document.getElementById('address').value;

const city = document.getElementById('city').value;

$.ajax({

type: 'POST',

url: '/update/' + first\_name + '/',

data: {

'last\_name': last\_name,

'phone': phone,

'address': address,

'city': city,

'csrfmiddlewaretoken': '{{ csrf\_token }}',

},

success: function() {

alert('Details updated successfully!');

},

error: function(xhr) {

alert('Error updating details: ' + xhr.statusText);

}

});

});

document.getElementById('deleteBtn').addEventListener('click', function() {

const first\_name = document.getElementById('first\_name').value;

if (confirm('Are you sure you want to delete this record?')) {

$.ajax({

type: 'GET',

url: '/delete/' + first\_name + '/',

success: function() {

alert('Record deleted successfully!');

// Refresh dropdown

location.reload();

},

error: function(xhr) {

alert('Error deleting record: ' + xhr.statusText);

}

});

}

});

});

</script>

</body>

</html>

Output 3.

