

WP LAB 10– Databases – Part II

Name: Pranamya G Kulal
Class: CSE A
Reg no: 220905018
Roll no: 8

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.

i) settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'books'  
]
```

ii) mylibrary/urls.py

```
from django.contrib import admin  
from django.urls import path, include
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('books/', include('books.urls')),  
]
```

iii) books/urls.py

```
from django.urls import path  
from . import views
```

```
urlpatterns = [  
    path('add_author/', views.add_author, name='add_author'),  
    path('authors/', views.list_authors, name='list_authors'),  
    path('add_publisher/', views.add_publisher, name='add_publisher'),  
    path('publishers/', views.list_publishers, name='list_publishers'),  
    path('add_book/', views.add_book, name='add_book'),  
    path('books/', views.list_books, name='list_books'),  
]
```

iv) views.py

```
from django.shortcuts import render, redirect  
from .models import Author, Publisher, Book  
from .forms import AuthorForm, PublisherForm, BookForm
```

```

# View to add a new author
def add_author(request):
    if request.method == "POST":
        form = AuthorForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('list_authors')
    else:
        form = AuthorForm()
    return render(request, 'books/add_author.html', {'form': form})

# View to list all authors
def list_authors(request):
    authors = Author.objects.all()
    return render(request, 'books/list_authors.html', {'authors': authors})

# View to add a new publisher
def add_publisher(request):
    if request.method == "POST":
        form = PublisherForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('list_publishers')
    else:
        form = PublisherForm()
    return render(request, 'books/add_publisher.html', {'form': form})

# View to list all publishers
def list_publishers(request):
    publishers = Publisher.objects.all()
    return render(request, 'books/list_publishers.html', {'publishers': publishers})

# View to add a new book
def add_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, 'books/add_book.html', {'form': form})

# View to list all books
def list_books(request):
    books = Book.objects.all()
    return render(request, 'books/list_books.html', {'books': books})

v)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=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
```

```
vi) 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__'
```

```
vii) 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)

viii) list_publishers.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>List of Publishers</title>
</head>
<body>
  <h1>Publishers</h1>
  <ul>
    {% for publisher in publishers %}
      <li>{{ publisher.name }} - {{ publisher.city }}, {{ publisher.country }}</li>
    {% endfor %}
  </ul>
</body>
</html>
```

ix) list_books.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>List of Books</title>
</head>
<body>
  <h1>List of Books</h1>
  <ul>
    {% for book in books %}
      <li>{{ book.title }} (Published: {{ book.publication_date }})</li>
    {% empty %}
      <li>No books available.</li>
    {% endfor %}
  </ul>
</body>
</html>
```

x) list_authors.html

```
<h2>Authors</h2>
<ul>
  {% for author in authors %}
    <li>{{ author.first_name }} {{ author.last_name }} - {{ author.email }}</li>
  {% endfor %}
</ul>
<a href="{% url 'add_author' %}">Add Author</a>
```

xi) add_publishers.html

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Add Publisher</title>
</head>
<body>
  <h1>Add a New Publisher</h1>

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

xii) add_book.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Add Book</title>
</head>
<body>
  <h2>Add a New Book</h2>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Save</button>
  </form>
  <a href="{% url 'list_books' %}">Back to Books List</a>
</body>
</html>
```

xiii) add_author.html

```
<form method="post">
  {% csrf_token %}
  {{ form.as_p }}
  <button type="submit">Add Author</button>
</form>
<a href="{% url 'list_authors' %}">View All Authors</a>
```

xiv) Screenshots

List of Books

- Django for Beginners (Published: March 10, 2024)
- Advanced Python (Published: July 15, 2023)

Add a New Publisher

Name:

Street address:

City:

State province:

Country:

Website:

Add Publisher

Authors

- J.K. Rowling - jkrowling@example.com
- George Orwell - georgeorwell@example.com
- John Doe - johndoe@example.com
- Jane Smith - janesmith@example.com

[Add Author](#)

Publishers

- TechBooks Publishing - San Francisco, USA
- EduWorld Publishers - New York, USA

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.

i) settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'products',  
]
```

ii) myproject/urls.py

```
from django.contrib import admin  
from django.urls import path, include
```

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

iii) products/urls.py

```
from django.urls import path  
from . import views
```

```
urlpatterns = [  
    path("", views.index, name='index'),  
    path('add/', views.add_product, name='add_product'),  
]
```

iv) 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})
```

v) 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
```

vi) forms.py
from django import forms
from .models import Product

```
class ProductForm(forms.ModelForm):  
    class Meta:  
        model = Product  
        fields = ['title', 'price', 'description']
```

vii) admin.py

```
from django.contrib import admin  
from .models import Product
```

```
admin.site.register(Product)
```

viii) add_product.html
<!DOCTYPE html>
<html>
<head>
 <title>Add Product</title>
</head>
<body>
 <h2>Add Product</h2>
 <form method="post">
 {% csrf_token %}
 {{ form.as_p }}
 <button type="submit">Save</button>
 </form>
 Back to Products
</body>
</html>

ix) index.html
<!DOCTYPE html>
<html>
<head>
 <title>Products</title>
</head>
<body>
 <h2>Product List</h2>


```
{% for product in products %}
  <li><strong>{{ product.title }}</strong> - ${{ product.price }}</li>
  <p>{{ product.description }}</p>
{% endfor %}
</ul>
<a href="{% url 'add_product' %}">Add New Product</a>
</body>
</html>
```

x)Screenshots

Add Product

Title:

Price:

Description:

Marvel limited edition Groot toy

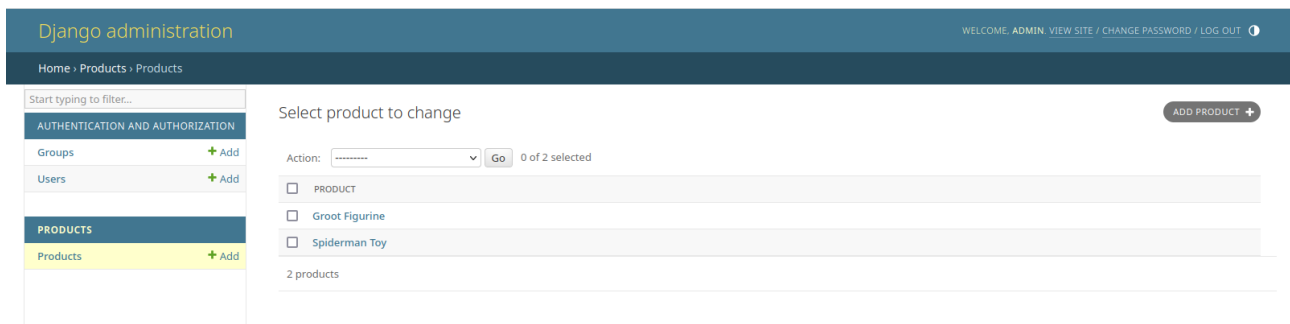
Save

[Back to Products](#)

Product List

- **Spiderman Toy** - \$800.00
Miniature spiderman toy limited edition
- **Groot Figurine** - \$650.00
Marvel limited edition Groot toy

[Add New 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.

i) settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'humans',
]
```

ii) myproject/urls.py

```
from django.contrib import admin
from django.urls import path, include
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('humans.urls')),
]
```

iii) humans/urls.py

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path("", views.human_list, name='human_list'),
    path('get_human/<str:first_name>', views.get_human, name='get_human'),
    path('update/', views.update_human, name='update_human'),
    path('delete/', views.delete_human, name='delete_human'),
]
```

iv) views.py

```
from django.shortcuts import render, get_object_or_404, redirect
from django.http import JsonResponse
from .models import Human
```

```

from .forms import HumanForm
from django.views.decorators.csrf import csrf_exempt

@csrf_exempt
def human_list(request):
    humans = Human.objects.all() # Fetch all first names
    return render(request, 'humans/human_form.html', {'humans': humans})

def get_human(request, first_name):
    human = get_object_or_404(Human, first_name=first_name)
    data = {
        'last_name': human.last_name,
        'phone': human.phone,
        'address': human.address,
        'city': human.city
    }
    return JsonResponse(data)

def update_human(request):
    if request.method == "POST":
        first_name = request.POST.get('first_name')
        human = get_object_or_404(Human, first_name=first_name)
        form = HumanForm(request.POST, instance=human)
        if form.is_valid():
            form.save()
            return JsonResponse({'success': True})
    return JsonResponse({'success': False})

def delete_human(request):
    if request.method == "POST":
        first_name = request.POST.get("first_name")
        try:
            Human.objects.filter(first_name=first_name).delete()
            return JsonResponse({"message": "Record deleted successfully!"})
        except:
            return JsonResponse({"message": "Error deleting record"}, status=500)
    return JsonResponse({"message": "Invalid request"}, status=400)

```

v) models.py

```

from django.db import models

```

```

class Human(models.Model):
    first_name = models.CharField(max_length=100, unique=True)
    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

```

```
vi) 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']
```

```
vii) admin.py
from django.contrib import admin
from .models import Human
```

```
admin.site.register(Human)
```

```
viii) human_form.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>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="container mt-5">

    <h2 class="text-center">Manage Human Records</h2>

    <form id="humanForm">
        {% csrf_token %}
        <input type="hidden" name="csrfmiddlewaretoken" value="{{ csrf_token }}">
        <!-- Dropdown for selecting first name -->
        <div class="mb-3">
            <label for="humanDropdown" class="form-label">Select First Name:</label>
            <select id="humanDropdown" class="form-select">
                <option value="">-- Select --</option>
                {% for human in humans %}
                    <option value="{{ human.first_name }}">{{ human.first_name }}</option>
                {% endfor %}
            </select>
        </div>

        <!-- Textboxes for other fields -->
        <div class="mb-3">
            <label for="last_name" class="form-label">Last Name:</label>
            <input type="text" id="last_name" class="form-control">
        </div>

        <div class="mb-3">
            <label for="phone" class="form-label">Phone:</label>
```

```

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

    <div class="mb-3">
        <label for="address" class="form-label">Address:</label>
        <input type="text" id="address" class="form-control">
    </div>

    <div class="mb-3">
        <label for="city" class="form-label">City:</label>
        <input type="text" id="city" class="form-control">
    </div>

    <!-- Action buttons -->
    <button type="button" id="updateBtn" class="btn btn-primary">Update</button>
    <button type="button" id="deleteBtn" class="btn btn-danger">Delete</button>
</form>

<script>
    $(document).ready(function() {
        function getCSRFToken() {
            return $('input[name=csrfmiddlewaretoken]').val();
        }

        // Fetch details when selecting a name
        $('#humanDropdown').change(function() {
            var firstName = $(this).val();
            if (firstName) {
                $.getJSON("/get_human/" + firstName + "/", function(data) {
                    $('#last_name').val(data.last_name);
                    $('#phone').val(data.phone);
                    $('#address').val(data.address);
                    $('#city').val(data.city);
                });
            } else {
                $('#humanForm input').val(""); // Clear fields if no name selected
            }
        });

        // Update human details
        $('#updateBtn').click(function() {
            var firstName = $('#humanDropdown').val();
            var lastName = $('#lastName').val();
            var phone = $('#phone').val();
            var address = $('#address').val();
            var city = $('#city').val();

            if (!firstName) {
                alert("Please select a name first.");
                return;
            }
        });
    });

```

```

$.ajax({
  url: "/update/",
  type: "POST",
  data: {
    first_name: firstName,
    last_name: lastName,
    phone: phone,
    address: address,
    city: city,
    csrfmiddlewaretoken: getCSRFToken()
  },
  success: function(response) {
    alert(response.message);
    location.reload();
  },
  error: function() {
    alert("Error updating record.");
  }
});
});

```

```

// Delete human record

```

```

$("#deleteBtn").click(function() {
  var firstName = $("#humanDropdown").val();
  if (!firstName) {
    alert("Please select a name first.");
    return;
  }

```

```

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

```

```

    $.ajax({
      url: "/delete/",
      type: "POST",
      data: {
        first_name: firstName,
        csrfmiddlewaretoken: getCSRFToken()
      },
      success: function(response) {
        alert(response.message);
        location.reload();
      },
      error: function() {
        alert("Error deleting record.");
      }
    });
  }

```

```

  });

```

```

});

```

```

</script>

```

```

</body>

```

```

</html>

```

ix) Screenshots

Manage Humans

Select First Name:

John

Last Name:

Doe

Phone:

1234567890

Address:

123 Elm Street

City:

New York

Update

Delete

Manage Humans

Select First Name:

Last Name:

Phone:

Address:

City:

Update

Delete