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.

==========

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=255)
    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=255)
    publication_date = models.DateField()
    authors         = models.ManyToManyField(Author)
    publisher       = models.ForeignKey(Publisher, on_delete=models.CASCADE)

    def __str__(self):
        return self.title
```

==========

views.py:

```
from django.shortcuts import render, redirect, get_object_or_404
from .forms import BookForm
from .models import Book

def create_book(request):
    if request.method == 'POST':
        book_form = BookForm(request.POST)
        if book_form.is_valid():
```

```
            book = book_form.save()
            return redirect('book_detail', pk=book.pk)
    else:
        book_form = BookForm()
    return render(request, 'library/create_book.html', {'book_form': book_form})

def book_detail(request, pk):
    book = get_object_or_404(Book, pk=pk)
    return render(request, 'library/book_detail.html', {'book': book})

def book_list(request):
    books = Book.objects.all()
    return render(request, 'library/book_list.html', {'books': books})
```

==========

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):
    class Meta:
        model = Book
        fields = ['title', 'publication_date', 'authors', 'publisher']
```

==========

book_detail.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>{{ book.title }}</title>
</head>
<body>
    <h1>{{ book.title }}</h1>
    <p><strong>Publication Date:</strong> {{ book.publication_date }}</p>
    <p><strong>Publisher:</strong> {{ book.publisher.name }}</p>
    <p><strong>Authors:</strong></p>
    <ul>
        {% for author in book.authors.all %}
            <li>{{ author.first_name }} {{ author.last_name }}</li>
        {% endfor %}
```

```
    </ul>
    <a href="{% url 'book_list' %}">Back to Book List</a>
</body>
</html>
```

==========

book_list.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Book List</title>
</head>
<body>
    <h1>Books</h1>
    <ul>
        {% for book in books %}
            <li>
                <a href="{% url 'book_detail' book.pk %}">{{ book.title }}</a>
            </li>
        {% endfor %}
    </ul>
    <a href="{% url 'create_book' %}">Add a new Book</a>
</body>
</html>
```

==========

create_book.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Create Book</title>
</head>
<body>
    <h1>Create Book</h1>
    <form method="post">
        {% csrf_token %}
        {{ book_form.as_p }}
        <button type="submit">Save Book</button>
    </form>
    <a href="{% url 'book_list' %}">Back to Book List</a>
</body>
</html>
```

==========

Output:

Create Book

Title: Harry

- Enter a valid date.

Publication date: 2003-12-04

Authors: John Doe

Publisher: Example Publisher

Save Book

Back to Book List

Books

- Harry

Add a new Book

Harry

Publication Date: Dec. 4, 2003

Publisher: Example Publisher

Authors:

- John Doe

Back to Book List

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.

==========

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
```

==========

forms.py:

```
from django import forms
from .models import Product

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

==========

views.py:

```
from django.shortcuts import render, redirect
from .forms import ProductForm
from .models import Product

def product_create(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/product_form.html', {'form': form})

def index(request):
    products = Product.objects.all()
    return render(request, 'products/index.html', {'products': products})
```

==========

index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Product List</title>
</head>
<body>
    <h1>Product List</h1>
    <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 'product_create' %}">Add a New Product</a>
</body>
</html>
```

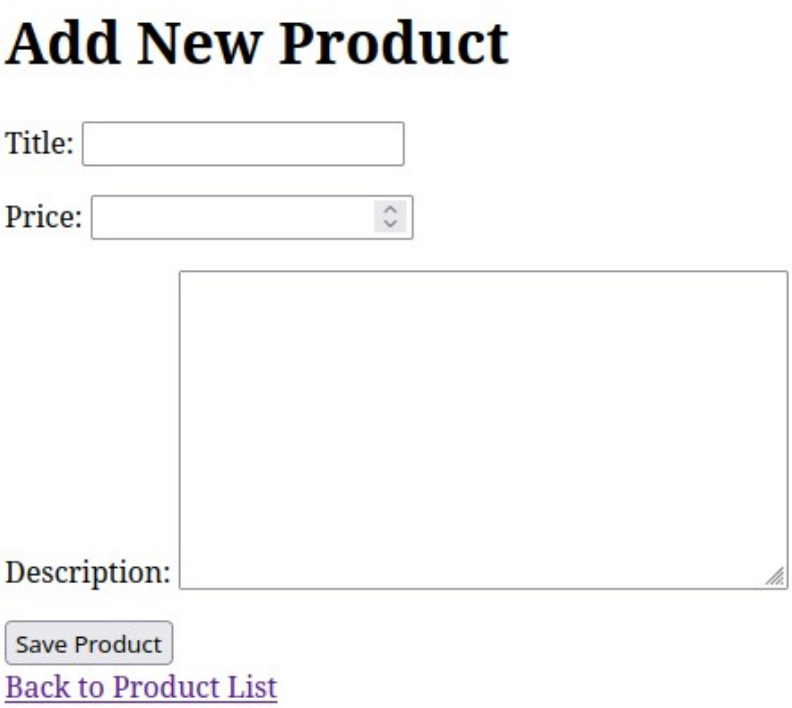
==========

product_form.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 Product List</a>
</body>
</html>
```

==========

Output:



==========

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.

==========

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=20)
    address = models.CharField(max_length=100)
    city = models.CharField(max_length=50)

    def __str__(self):
        return self.first_name
```

==========

views.py:

```
from django.shortcuts import render, get_object_or_404
from django.http import JsonResponse
from .models import Human
from django.views.decorators.csrf import csrf_exempt

def human_list(request):
    # Query all Human objects to populate the dropdown list
    humans = Human.objects.all()
    return render(request, 'human_list.html', {'humans': humans})

def get_human(request):
    # This view returns the details for a selected human via AJAX
    human_id = request.GET.get('id')
    if human_id:
        human = get_object_or_404(Human, pk=human_id)
        data = {
            'first_name': human.first_name,
            'last_name': human.last_name,
            'phone': human.phone,
            'address': human.address,
            'city': human.city,
        }
        return JsonResponse(data)
    return JsonResponse({'error': 'No id provided'}, status=400)

@csrf_exempt
def update_human(request):
    # Update the record with new data from the text boxes
    if request.method == 'POST':
        human_id = request.POST.get('id')
        human = get_object_or_404(Human, pk=human_id)
        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({'status': 'success'})
    return JsonResponse({'error': 'Invalid method'}, status=400)

@csrf_exempt
def delete_human(request):
    # Delete the selected human record
    if request.method == 'POST':
        human_id = request.POST.get('id')
        human = get_object_or_404(Human, pk=human_id)
        human.delete()
        return JsonResponse({'status': 'success'})
    return JsonResponse({'error': 'Invalid method'}, status=400)
```

==========

human_list.html:

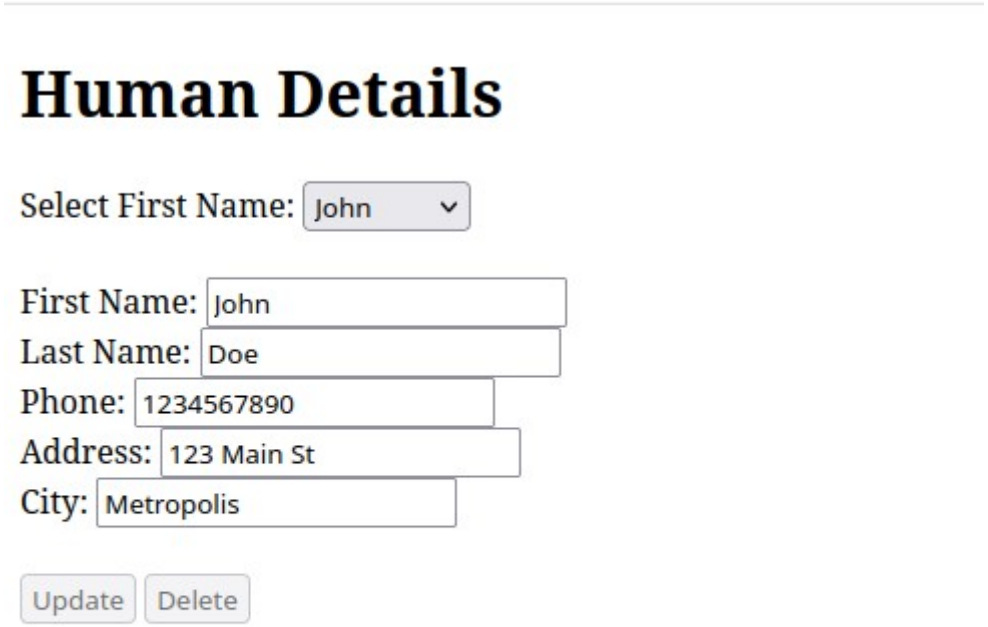
```
<!DOCTYPE html>
<html>
<head>
    <title>Human Details</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <h1>Human Details</h1>

    <!-- Dropdown with options from the database -->
    <label for="humanSelect">Select First Name:</label>
    <select id="humanSelect">
        <option value="">--Select--</option>
        {% for human in humans %}
            <option value="{{ human.id }}">{{ human.first_name }}</option>
        {% endfor %}
    </select>

    <br><br>

    <!-- Textboxes for details -->
    <label>First Name:</label>
    <input type="text" id="first_name"><br>

    <label>Last Name:</label>
    <input type="text" id="last_name"><br>

    <label>Phone:</label>
    <input type="text" id="phone"><br>

    <label>Address:</label>
    <input type="text" id="address"><br>
```

```
<label>City:</label>
<input type="text" id="city"><br>

<br>

<!-- Initially disabled buttons -->
<button id="updateBtn" disabled>Update</button>
<button id="deleteBtn" disabled>Delete</button>

<script>
   $(document).ready(function(){
      // Disable buttons on initial page load
      $('#updateBtn, #deleteBtn').prop('disabled', true);

      // When a record is selected
      $('#humanSelect').change(function(){
         var id = $(this).val();
         if(id) {
            // Enable buttons when a valid record is chosen
            $('#updateBtn, #deleteBtn').prop('disabled', false);

            // Fetch and display record details via AJAX
            $.ajax({
               url: "{% url 'get_human' %}",
               data: { id: id },
               dataType: 'json',
               success: 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);
               }
            });
         } else {
            // Clear fields and disable buttons if no record is selected
            $('#first_name, #last_name, #phone, #address, #city').val('');
            $('#updateBtn, #deleteBtn').prop('disabled', true);
         }
      });

      // Update button handler
      $('#updateBtn').click(function(){
         var id = $('#humanSelect').val();
         if(id) {
            $.ajax({
               url: "{% url 'update_human' %}",
               type: 'POST',
               data: {
                  id: id,
                  first_name: $('#first_name').val(),
                  last_name: $('#last_name').val(),
```

```
                phone: $('#phone').val(),
                address: $('#address').val(),
                city: $('#city').val(),
                csrfmiddlewaretoken: '{{ csrf_token }}'
            },
            dataType: 'json',
            success: function(response) {
                alert('Record updated successfully.');
                location.reload();
            }
        });
    } else {
        alert('Please select a record to update.');
    }
});

// Delete button handler
$('#deleteBtn').click(function(){
    var id = $('#humanSelect').val();
    if(id) {
        if(confirm('Are you sure you want to delete this record?')) {
            $.ajax({
                url: "{% url 'delete_human' %}",
                type: 'POST',
                data: {
                    id: id,
                    csrfmiddlewaretoken: '{{ csrf_token }}'
                },
                dataType: 'json',
                success: function(response) {
                    alert('Record deleted successfully.');
                    location.reload();
                }
            });
        }
    } else {
        alert('Please select a record to delete.');
    }
});
});
</script>
</body>
</html>
```

==========

Output:


Human Details
Select First Name: John
First Name: John
Last Name: Doe
Phone: 1234567890
Address: 123 Main St
City: Metropolis
Update
Delete