**Web Programming Lab 10**

**Name: Sanketh Naik**

**Section and RollNo: (D2 57):**

1. **Source Code (lab10q1 folder)**

**Now type the command**

***django-admin startproject lab10q1***

***python manage.py startapp lab10-1app***

*In that open the following files:*

*urls.py:*

from django.urls import path

from . import views

urlpatterns=[

path('',views.index,name="index"),

path('publish/',views.publisher,name="publisher"),

path('book/',views.book,name="book")

]

*views.py update the code as follows:*

from django.shortcuts import render

from .models import \*

from .bforms import \*

# Create your views here.

def index(request):

if request.method=="POST":

af = auForm(request.POST)

if af.is\_valid():

af.save()

else:

af = auForm()

ad = ModAuth.objects.all()

print(ad)

return render(request,"author.html",{

"af":af,

"ads":ad,

})

def publisher(request):

if request.method=="POST":

pf = FormPub(request.POST)

if pf.is\_valid():

pf.save()

else:

pf = FormPub()

pd = ModPubl.objects.all()

return render(request,"publisher.html",{

"pf":pf,

"pd":pd,

})

def book(request):

if request.method=="POST":

bf = FormBk(request.POST)

if bf.is\_valid():

bf.save()

else:

bf = FormBk()

bookD = BookModel.objects.all().prefetch\_related("acate","pcate")

for book in bookD:

book.author\_fn = ", ".join([f"{author.fname} {author.lname}" for author in book.acate.all()])

return render(request,"book.html",{

"bf":bf,

"bookD":bookD,

})

***Open lab10q1 folder in that open urls.py:***

from django.contrib import admin

from django.urls import path,include

urlpatterns = [

path('admin/', admin.site.urls),

path('lab10-1app/', include('lab10-1app.urls')),

]

***Open lab10q1 folder in that open settings.py:***

here in top `import os` module

update the templates list DIRS params as follows:

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [os.path.join(BASE\_DIR,lab10-1appp/templates')],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

In lab10-1appp folder open bforms.py file:

from django import forms

from .models import \*

class auForm(forms.ModelForm):

class Meta:

model = ModAuth

fields = ['fname','lname','emailAddr']

widgets = {

'fname': forms.TextInput(),

'lname': forms.TextInput()

emailAddr': forms.EmailInput()

}

class FormPub(forms.ModelForm):

class Meta:

model = ModPubl

fields = ['name','street','city','state','country','website']

widgets = {

'name' : forms.TextInput(),

'street' : forms.TextInput(),

'city' : forms.TextInput(),

'state' : forms.TextInput(),

'country' : forms.TextInput(),

'website' : forms.TextInput()

}

class FormBk(forms.ModelForm):

class Meta:

model = BookModel

fields = ['title','publishDate','acate','pcate']

widgets = {

'title': forms.TextInput(attrs={),

'publishDate': forms.DateInput(),

'acate': forms.SelectMultiple(),

'pcate': forms.Select()

}

In models.py file:

from django.db import models

# Create your models here.

class ModPubl(models.Model):

name = models.CharField(max\_length=64)

street = models.CharField(max\_length=64)

city = models.CharField(max\_length=64)

state = models.CharField(max\_length=64)

country = models.CharField(max\_length=64)

website = models.CharField(max\_length=64)

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

return f"{self.name}, {self.country}"

class ModAuth(models.Model):

fname = models.CharField(max\_length=64)

lname = models.CharField(max\_length=64)

emailAddr = models.EmailField(max\_length=64)

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

return f"{self.fname}{self.lname}"

class BookModel(models.Model):

title = models.CharField(max\_length=64)

publishDate = models.DateField()

acate = models.ManyToManyField(ModAuth, blank=True, related\_name="author")

pcate = models.ForeignKey(ModPubl, blank=True, related\_name="publisher", on\_delete=models.CASCADE)

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

return self.title

In admin.py file:

from django.contrib import admin

from .models import \*

# Register your models here.

admin.site.register(ModAuth)

admin.site.register(ModPubl)

admin.site.register(BookModel)

*create a templates folder and add the html,css files inside it:*

*author.html file:*

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Author Page</title>

</head>

<body>

<div >

<h1>Author Form</h1>

<form method="post">

{%csrf\_token%}

<div class="mb-3">

{{af.as\_p}}

</div>

<div class="mb-3">

<button type="submit">Submit</button>

</div>

</form>

<div class="mb-3">

<a href="{% url 'publisherPage' %}">Publisher Table</a>

<a href="{% url 'bookPage' %}">Book Table</a>

</div>

</div>

<div>

<h1>Author Details</h1>

<table>

<thead>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Email Address</th>

</tr>

</thead>

<tbody>

{% for ad in ads%}

<tr>

<td>{{ad.fname}}</td>

<td>{{ad.lname}}</td>

<td>{{ad.emailAddr}}</td>

</tr>

{% empty %}

<p>No records found</p>

{% endfor %}

</tbody>

</table>

</div>

</body>

</html>

book.html file:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<title>Lab10\_1: Book Page</title>

</head>

<body>

<div class="container hover bg-dark text-info">

<h1 class="text-center">Book Form</h1>

<form method="post">

{%csrf\_token%}

<div class="mb-3">

{{bf.as\_p}}

</div>

<div class="mb-3">

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

</div>

</form>

<div class="mb-3">

<a href="{% url 'publisherPage' %}" class="mx-5 my-5 my-5 mb-5">Publisher Table</a>

<a href="{% url 'authorPage' %}" class="mx-5 my-5 my-5 mb-5">Author Table</a>

</div>

</div>

<div class="container bg-primary text-white table-responsive">

<h1 class="text-center">Book Details</h1>

<table class="table table-bordered table-hover">

<thead>

<tr>

<th>Book Title</th>

<th>Publishing Date</th>

<th>AuthCat </th>

<th>PublishCat</th>

</tr>

</thead>

<tbody>

{% for bookDetail in bookD%}

<tr>

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

<td>{{bookDetail.publishDate}}</td>

<td>{{bookDetail.author\_fn}}</td>

<td>{{bookDetail.pcate}}</td>

</tr>

{% empty %}

<p>No records found</p>

{% endfor %}

</tbody>

</table>

</div>

</body>

</html>

publisher.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<title>Lab10\_1: Publisher Page</title>

</head>

<body>

<div class="container hover bg-dark text-info">

<h1 class="text-center">Publisher Form</h1>

<form method="post">

{%csrf\_token%}

<div class="mb-3">

{{pf.as\_p}}

</div>

<div class="mb-3">

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

</div>

</form>

<div class="mb-3">

<a href="{% url 'authorPage' %}" class="mx-5 my-5 my-5 mb-5">Author Table</a>

<a href="{% url 'bookPage' %}" class="mx-5 my-5 my-5 mb-5">Book Table</a>

</div>

</div>

<div class="container bg-primary text-white table-responsive">

<h1 class="text-center">Publisher Details</h1>

<table class="table table-bordered table-hover">

<thead>

<tr>

<th>Name</th>

<th>Street Address</th>

<th>City</th>

<th>State/Province</th>

<th>Country</th>

<th>Web Site</th>

</tr>

</thead>

<tbody>

{% for publisherDetail in pd%}

<tr>

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

<td>{{publisherDetail.street}}</td>

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

<td>{{publisherDetail.state}}</td>

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

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

</tr>

{% empty %}

<p>No records found</p>

{% endfor %}

</tbody>

</table>

</div>

</body> html>

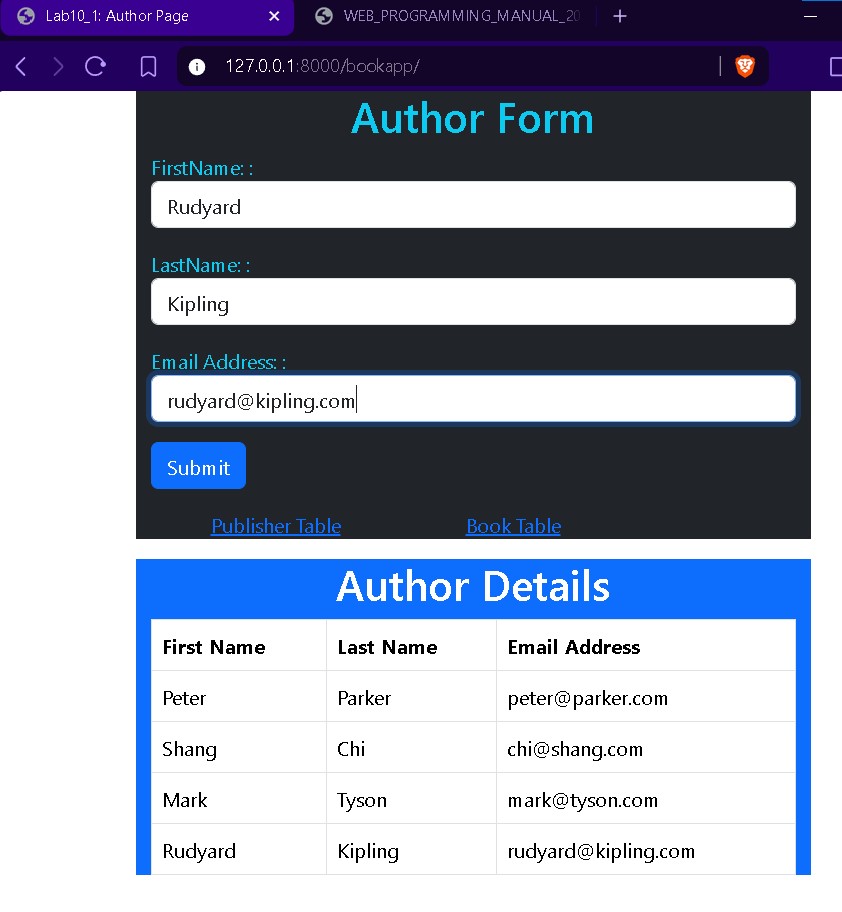
Now in terminal (under lab10q1 path) type:

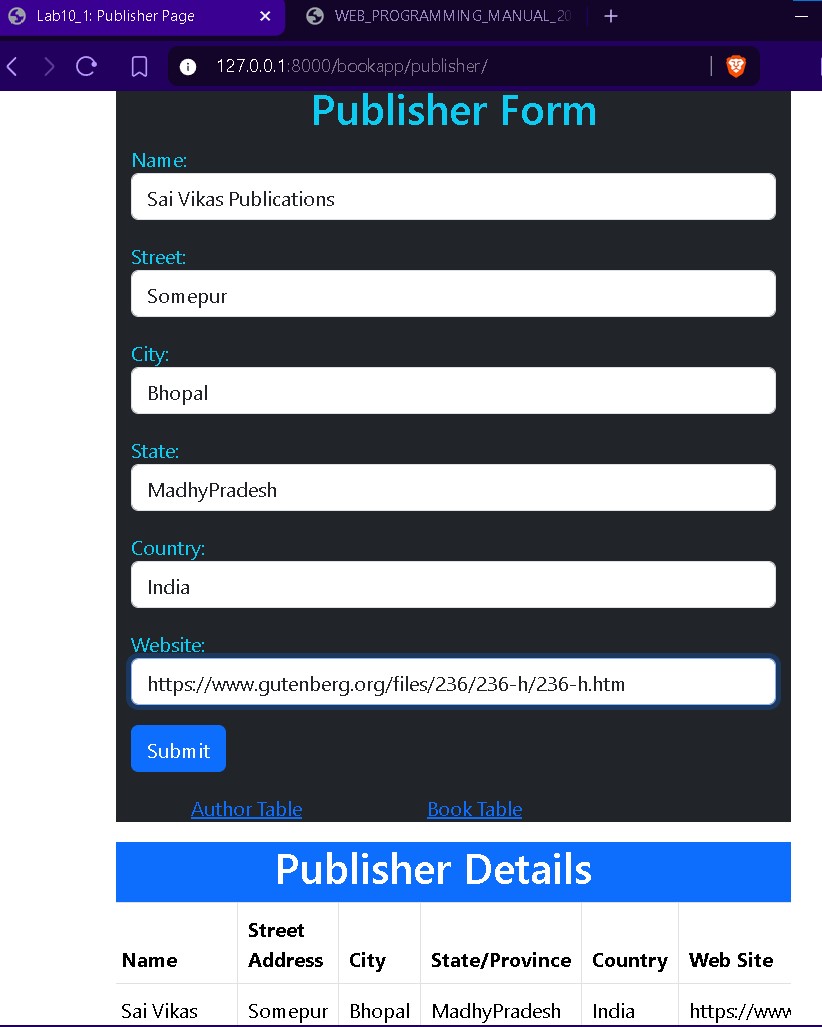
***python manage.py migrate***

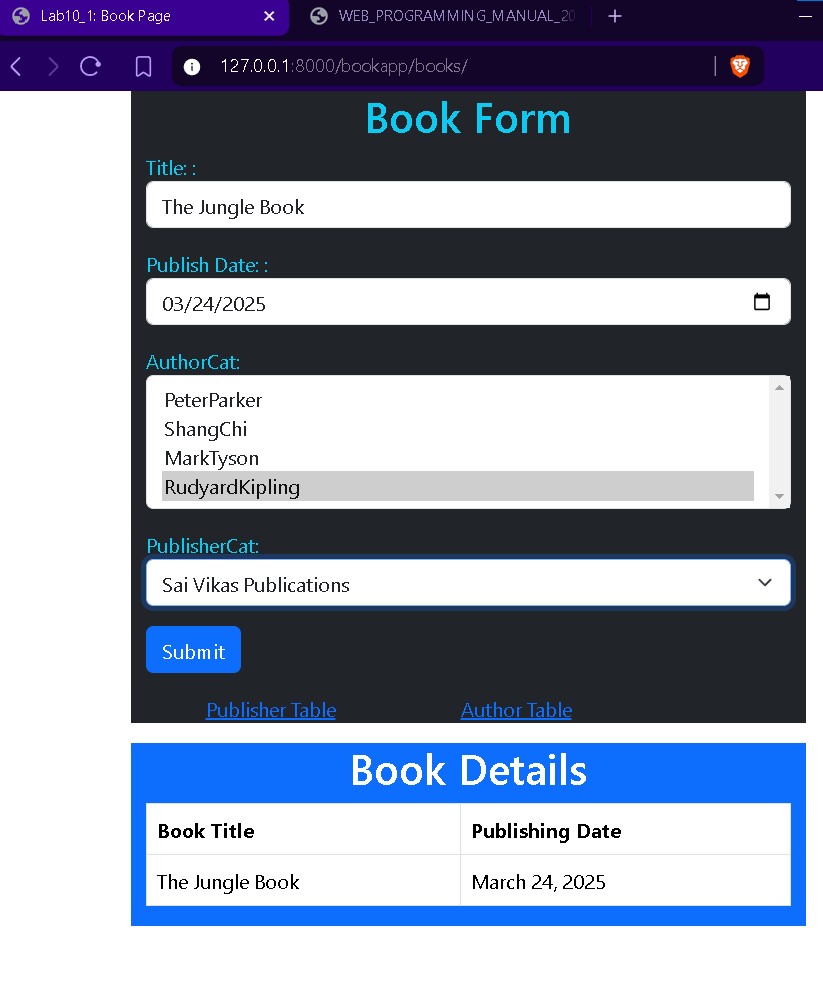
***python manage.py makemigrations***

***python manage.py createsuperuser***

***python manage.py runserver 8000***

Output:





1. **Source Code (lab10\_2q folder)**

**django-admin startproject lab10\_2q.**

**Now type the command** *python manage.py startapp lab10\_2qapp #by doing this lab10\_2qapp folder is created. In that open the following files:*

*urls.py (need to create this file):*

from django.urls import path

from . import views

urlpatterns=[

path('',views.index,name="productPage"),

]

*views.py update the code as follows:*

from django.shortcuts import render

from .models import ModPro

from .pforms import PForm

from django.contrib import messages

# Create your views here.

def index(request):

errMsg=None

if request.method=="POST":

pform = PForm(request.POST)

if pform.is\_valid():

ptitle = pform.cleaned\_data.get('title')

if ModPro.objects.filter(title=ptitle).exists():

errMsg = "The product detail already exists!"

messages.error(request,errMsg)

else:

pform.save()

else:

pform = PForm()

productDetails = ModPro.objects.all()

return render(request,"product.html",{

"pform":pform,

"productDetails":productDetails,

"errMsg":errMsg

})

***Open lab10\_2q folder in that open urls.py:***

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('lab10\_2qapp/', include('lab10\_2qapp.urls')),

]

***Open lab10\_2q folder in that open settings.py:***

here in top `import os` module

update the templates list DIRS params as follows:

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [os.path.join(BASE\_DIR, lab10\_2qapp /templates')],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

In lab10\_2qapp folder open pforms.py file:

from django import forms

from .models import ModPro

class PForm(forms.ModelForm):

class Meta:

model=ModPro

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

widgets={

'title':forms.TextInput(a),

'price':forms.NumberInput(),

'description':forms.Textarea(),

}

In models.py file:

from django.db import models

# Create your models here.

class ModPro(models.Model):

title = models.CharField(verbose\_name="Title: ",max\_length=128)

price = models.IntegerField(verbose\_name="Price: ")

description = models.TextField(verbose\_name="Description: ",max\_length=102)

In admin.py file:

from django.contrib import admin

from .models import ModPro

# Register your models here.

class ProductList(admin.ModelAdmin):

list\_display=("title","price","description")

admin.site.register(ModPro,ProductList)

*create a templates folder and add the html,css files inside it:*

*product.html file:*

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

</head>

<body>

<div>

<h1>Product Form</h1>

<form method="post">

{% csrf\_token %}

<div class="mb-3">

{{pform.as\_p}}

</div>

<div>

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

</div>

</form>

<h1 class="text-center mt-5 mb-5">Product Details</h1>

<table class="table table-striped table-bordered">

<tbody>

<ul class="list-group">

{% if errMsg %}

<p class="text-center">{{errMsg}}</p>

{% else %}

{% for productDetail in productDetails %}

<tr>

<td><h1>\*</h1></td>

<td>

<li class="list-group-item">{{productDetail.title}}</li>

</td>

<td>

<li class="list-group-item">{{productDetail.price}}</li>

</td>

<td>

<li class="list-group-item">{{productDetail.description}}</li>

</td>

{%empty%}

<p class="text-center">No records found</p>

</tr>

{% endfor %}

{% endif %}

</ul>

</tbody>

</table>

</div>

</body>

</html>

Now in terminal (under lab10\_2q path) type:

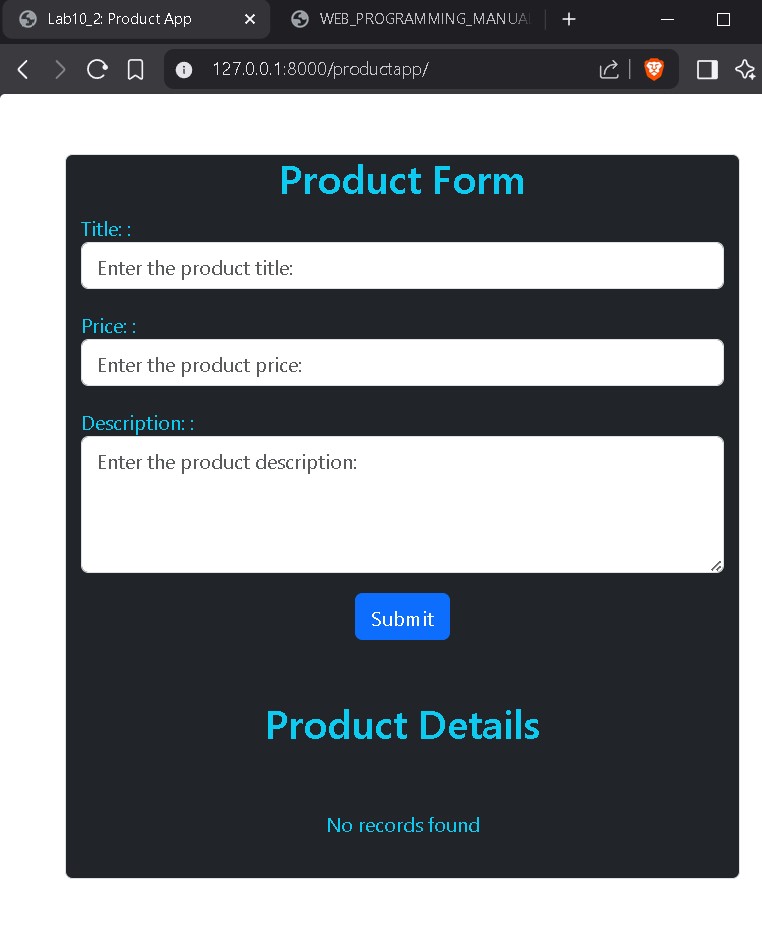
***python manage.py migrate***

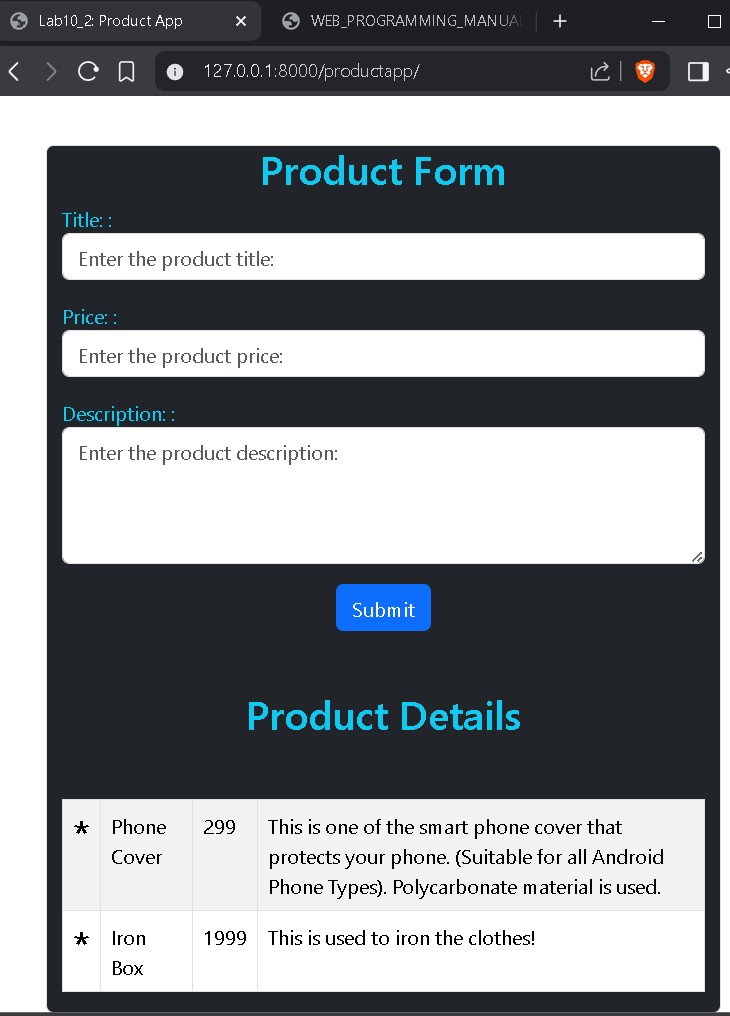
***python manage.py makemigrations***

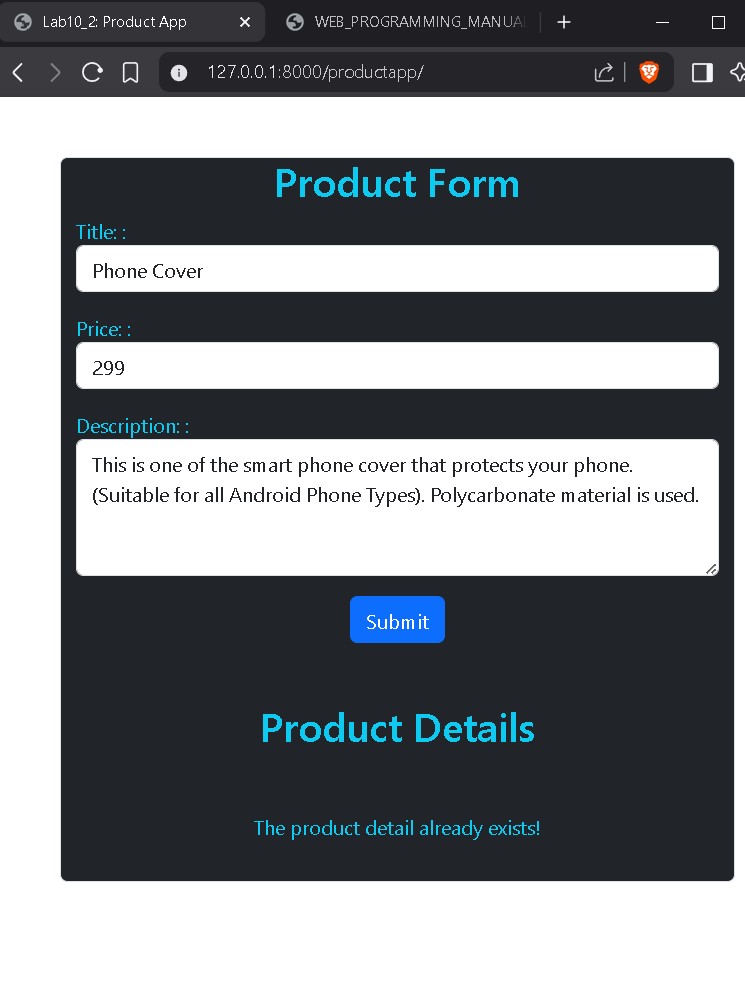
***python manage.py createsuperuser***

***python manage.py runserver 8000***

Output:







1. **Source Code (lab10\_3 folder) (**created viacommand: **django-admin startproject lab10\_3 ).**

**Now type the command** *python manage.py startapp lab10\_3qapp #by doing this lab10\_3qapp folder is created. In that open the following files:*

*urls.py (need to create this file):*

from django.urls import path

from . import views

urlpatterns=[

path('',views.index,name="humanPage"),

path('crudop/',views.crudOp,name="crudPage"),

]

*views.py update the code as follows:*

from django.shortcuts import render, redirect

from .models import MOdHum

from .hforms import FormHum

from django.contrib import messages

# Create your views here.

def index(request):

errMsg=None

if request.method == "POST":

hform = FormHum(request.POST)

if hform.is\_valid():

hphno = hform.cleaned\_data.get('phno')

if MOdHum.objects.filter(phno=hphno).exists():

errMsg="Detail already exists!"

messages.error(request,errMsg)

else:

hform.save()

else:

hform = FormHum()

return render(request,"index.html",{

"hForm":hform,

"errMsg":errMsg,

})

def crudOp(request):

hdetails = MOdHum.objects.all()

selected\_record = None

selected\_id = None

if request.method == "POST":

selected\_id = request.POST.get("selected\_id")

action = request.POST.get("action")

if selected\_id:

selected\_record = MOdHum.objects.get(id=selected\_id)

if action == "update":

selected\_record.fname = request.POST.get("fname")

selected\_record.lname = request.POST.get("lname")

selected\_record.phno = request.POST.get("phno")

selected\_record.address = request.POST.get("address")

selected\_record.city = request.POST.get("city")

selected\_record.save()

messages.success(request, "Record updated successfully!")

elif action == "delete":

selected\_record.delete()

messages.success(request, "Record deleted successfully!")

return redirect('crudOp')

return render(request, "crud.html", {

"hDetails": hdetails,

"selected\_record": selected\_record,

"selected\_id": selected\_id

})

def update\_record(request, pk):

record = MOdHum.objects.get(id=pk)

if request.method == "POST":

form = FormHum(request.POST, instance=record)

if form.is\_valid():

form.save()

messages.success(request, "Record updated successfully!")

return redirect('crudOp')

else:

form = FormHum(instance=record)

return render(request, "update.html", {

'form': form,

'record': record

})

def delete\_record(request, pk):

record = MOdHum.objects.get(id=pk)

record.delete()

messages.success(request, "Record deleted successfully!")

return redirect('crudOp')

***Open lab10\_3 folder in that open urls.py:***

from django.contrib import admin

from django.urls import path,include

urlpatterns = [

path('admin/', admin.site.urls),

path('crudapp/', include('lab10\_3qapp.urls'))

]

***Open lab10\_3 folder in that open settings.py:***

here in top `import os` module

update the templates list DIRS params as follows:

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [os.path.join(BASE\_DIR,lab10\_3qapp/templates')],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

In lab10\_3qapp folder open hforms.py file:

from django import forms

from .models import MOdHum

class FormHum(forms.ModelForm):

class Meta:

model=MOdHum

fields="\_\_all\_\_"

widgets={

'fname':forms.TextInput(),

'lname':forms.TextInput(),

'phno':forms.NumberInput(),

'address':forms.TextInput(),

'city':forms.TextInput(),

}

In models.py file:

from django.db import models

# Create your models here.

class MOdHum(models.Model):

fname = models.CharField(max\_length=64,verbose\_name="FirstName: ")

lname = models.CharField(max\_length=64,verbose\_name="LastName: ")

phno = models.IntegerField(max\_length=10,verbose\_name="Phno: ")

address = models.CharField(max\_length=64,verbose\_name="Address: ")

city = models.CharField(max\_length=64,verbose\_name="City: ")

In admin.py file:

from django.contrib import admin

from .models import MOdHum

# Register your models here.

admin.site.register(MOdHum)

*create a templates folder and add the html,css files inside it:*

*index.html file:*

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

</head>

<body>

<div class="container bg-dark border rounded text-info">

<h1 class="text-center">Add a New Record</h1>

{% if errMsg %}

<p class="text-center">{{ errMsg }}</p>

{% endif %}

<form method="POST">

{% csrf\_token %}

{{ hForm.as\_p }}

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

</form>

<a href="{% url 'crudOp' %}">Go to CRUD Operations</a>

</div>

</body>

</html>

crud.html file:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<title>CRUD Operations</title>

</head>

<body>

<div class="container bg-dark text-info">

<h1 class="text-center">CRUD Operations</h1>

<!-- Form to Select Record for Update or Delete -->

<form method="POST">

{% csrf\_token %}

<label for="name" class="form-label">Select Name:</label>

<select name="selected\_id" id="name">

<option value="">--Select a Name--</option>

{% for detail in hDetails %}

<option value="{{ detail.id }}" {% if detail.id == selected\_id %}selected{% endif %}>{{ detail.fname }}</option>

{% endfor %}

</select>

<button type="submit" name="action" value="select" class="btn btn-warning">Select</button>

</form>

{% if selected\_record %}

<!-- Display the selected record's details -->

<h2 class="text-center">Selected Record: {{ selected\_record.fname }} {{ selected\_record.lname }}</h2>

<form method="POST">

{% csrf\_token %}

<input type="hidden" name="selected\_id" value="{{ selected\_record.id }}">

<label for="fname">First Name:</label>

<input type="text" name="fname" value="{{ selected\_record.fname }}">

<br>

<label for="lname">Last Name:</label>

<input type="text" name="lname" value="{{ selected\_record.lname }}">

<br>

<label for="phno">Phone Number:</label>

<input type="text" name="phno" value="{{ selected\_record.phno }}">

<br>

<label for="address">Address:</label>

<input type="text" name="address" value="{{ selected\_record.address }}">

<br>

<label for="city">City:</label>

<input type="text" name="city" value="{{ selected\_record.city }}">

<br>

<!-- Buttons for update and delete -->

<button type="submit" name="action" value="update">Update</button>

<button type="submit" name="action" value="delete">Delete</button>

</form>

{% endif %}

<hr>

<h2>Existing Records</h2>

<table >

<tr>

<th>First Name</th>

<th>Actions</th>

</tr>

{% for detail in hDetails %}

<tr>

<td>{{ detail.fname }}</td>

<td>

<form method="POST" action="{% url 'update\_record' detail.id %}">

{% csrf\_token %}

<button type="submit">Edit</button>

</form>

<form method="POST" action="{% url 'delete\_record' detail.id %}">

{% csrf\_token %}

<button type="submit">Delete</button>

</form>

</td>

</tr>

{% endfor %}

</table>

<a href="{% url 'index' %}">Go to Add Record</a>

</div>

</body>

</html>

update.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Update Record</title>

</head>

<body>

<h1>Update Record for {{ record.fname }} {{ record.lname }}</h1>

<form method="POST">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Update</button>

</form>

<a href="{% url 'crudOp' %}">Back to CRUD Operations</a>

</body>

</html>

Now in terminal (under lab10\_3 path) type:

***python manage.py migrate***

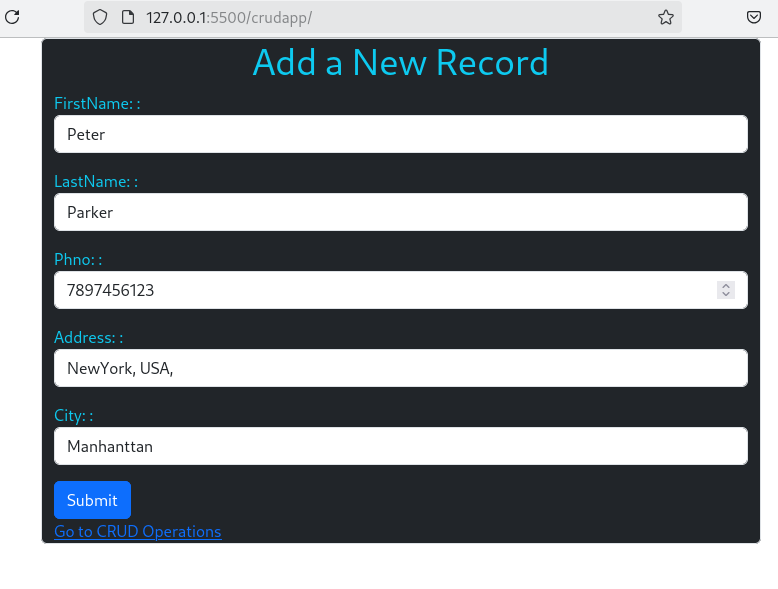
***python manage.py makemigrations***

***python manage.py createsuperuser***

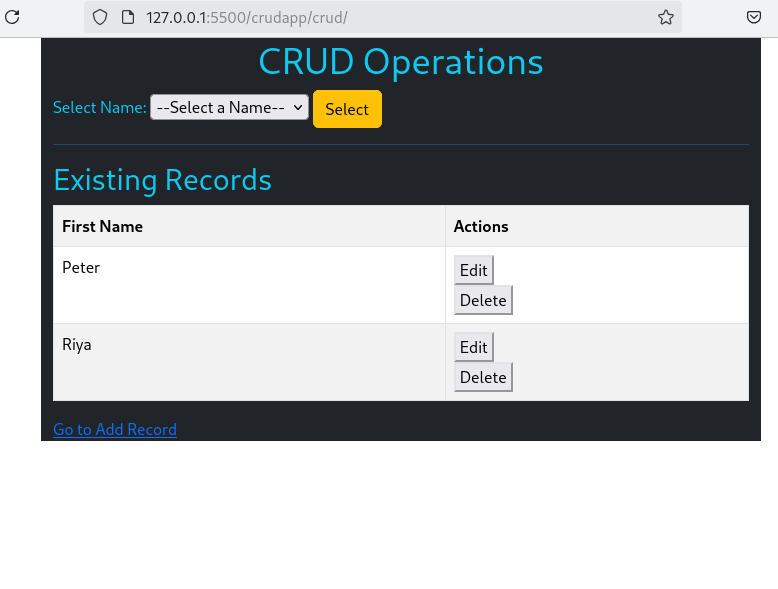
***python manage.py runserver 8000***

Output:

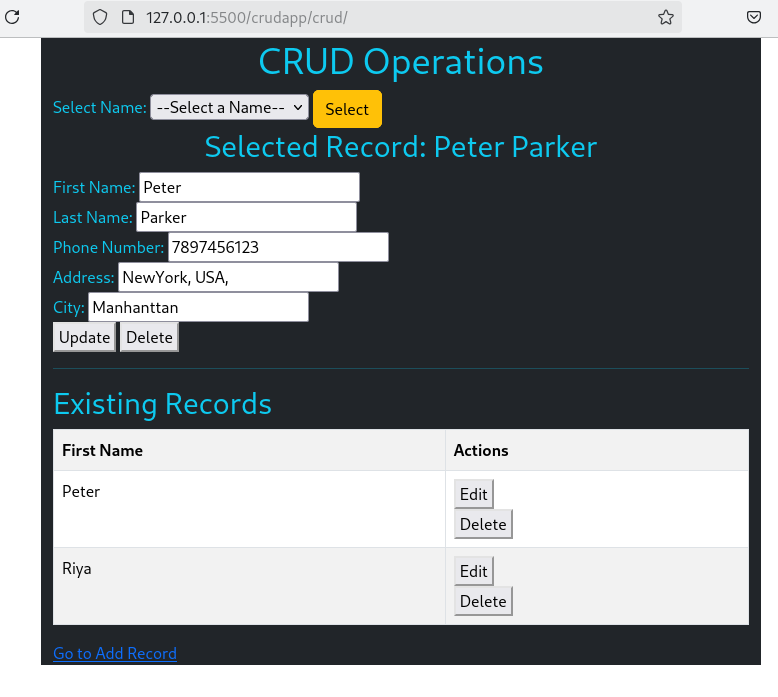
**User Interface:**

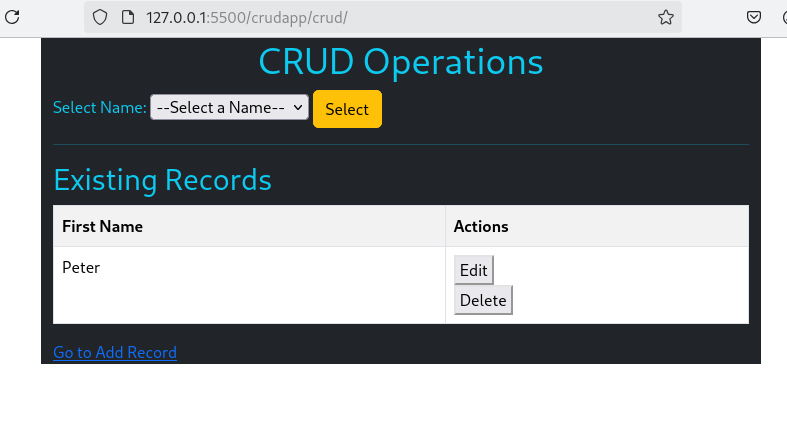


**CRUD Page:**



**Selection of Records:**

**Deleting Riya Record**



**Updating Peter Parker Record to Peter Hence:**

