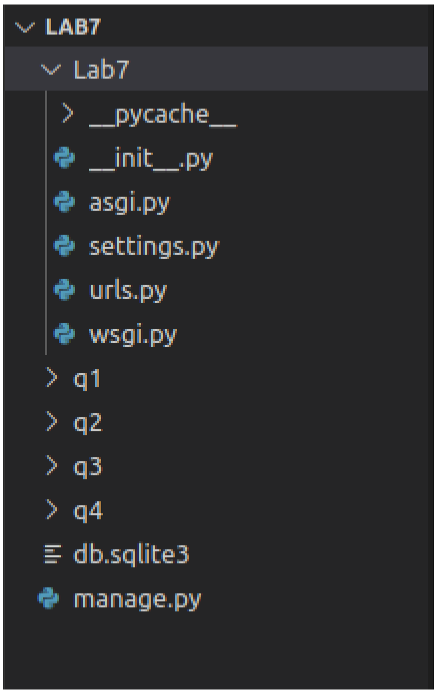
SANSKAAR PATNI
180905134 CSE C-23
LAB 7- DATABASES

Created a project Lab7 and 1 app for each project q1,q2,q3,q4 respectively:

Folder Structure



```
LAB7
  Lab7
    __pycache__
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  q1
  q2
  q3
  q4
  db.sqlite3
  manage.py
```

Changes in
Lab7/Lab7/settings.py

```
INSTALLED_APPS = [
   'q1',
   'q2',
   'q3',
   'q4',
   'django.contrib.admin',
   'django.contrib.auth',
```

Lab7/Lab7/urls.py

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

urlpatterns = [
   path('admin/', admin.site.urls),
   url(r'^q1/', include('q1.urls')),
   url(r'^q2/', include('q2.urls')),
   url(r'^q3/', include('q3.urls')),
   url(r'^q4/', include('q4.urls')),
]
```

1. 1st question

Lab7/q1/

models.py

```
from django.db import models
from django import forms
from django.db.models import constraints


class Category(models.Model):
    name = models.CharField(max_length=30)
    # noOfVisits is sum of all page views of the category
    noOfVisits = models.IntegerField(default=0)
    noOfLikes = models.IntegerField()

    def __str__(self):
        return self.name


class Page(models.Model):
    category = models.ForeignKey(Category, on_delete=models.CASCADE)
    title = models.CharField(max_length=30)
    # views:can set a default as well as it gets incremented if we click
on the link
    views = models.IntegerField(default=0)
    url = models.CharField(
        max_length=50, default="https://www.google.com")

    class Meta:
        ordering = ['views']

    def __str__(self):
        return self.title


class PageForm(forms.ModelForm):
    class Meta:
        model = Page
        fields = '__all__'


class CategoryForm(forms.ModelForm):
    class Meta:
        model = Category
        exclude = ('noOfVisits',)
```

Created urls.py

```
from django.conf.urls import url
from django.urls import path
from . import views
urlpatterns = [
    path('list', views.pageList, name='pageList'),
    path('forms', views.forms, name='forms'),
    path('categorySubmit', views.categorySubmit, name='categorySubmit'),
    path('pageSubmit', views.pageSubmit, name='pageSubmit'),
    path('<int:pk1>/updateViews/', views.updateViews,
name='updateViews'),

]
```

views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from q1.models import Category, Page, CategoryForm, PageForm


def pageList(request):
    if request.method == 'POST':
        category = request.POST['category']
        pages = Page.objects.filter(category__name__startswith=category)
        return render(request, 'q1/index.html', {'pages': pages})
    pages = Page.objects.all()[:10]
    return render(request, 'q1/index.html', {'pages': pages})


def forms(request):
    return render(request, 'q1/forms.html', {'categoryForm':
CategoryForm(), 'pageForm': PageForm()})
```

```python
def updateViews(request, pk1):
    page = Page.objects.get(pk=pk1)
    page.views += 1
    page.save()
    category = Category.objects.get(name=page.category)
    category.noOfVisits += 1
    category.save()
    return HttpResponseRedirect(page.url)


def pageSubmit(request):
    if request.method == 'POST':
        form = PageForm(request.POST)
        if form.is_valid():
            page = form.save(commit=False)
            page.save()
            category = Category.objects.get(name=page.category)
            category.noOfVisits += page.views
            category.save()
            print(category)
            return HttpResponseRedirect('/q1/list')
        else:
            return render(request, 'q1/forms.html', {'categoryForm': CategoryForm(), 'pageForm': PageForm()})


def categorySubmit(request):
    if request.method == 'POST':
        form = CategoryForm(request.POST)
        if form.is_valid():
            category = form.save(commit=False)
            category.save()
            category.noOfVisits = 0
            category.save()
            return HttpResponseRedirect('/q1/list')
        else:
            return render(request, 'q1/forms.html', {'categoryForm': CategoryForm(), 'pageForm': PageForm()})
```

Created Folder templates/q2/
Files Created:
index.html

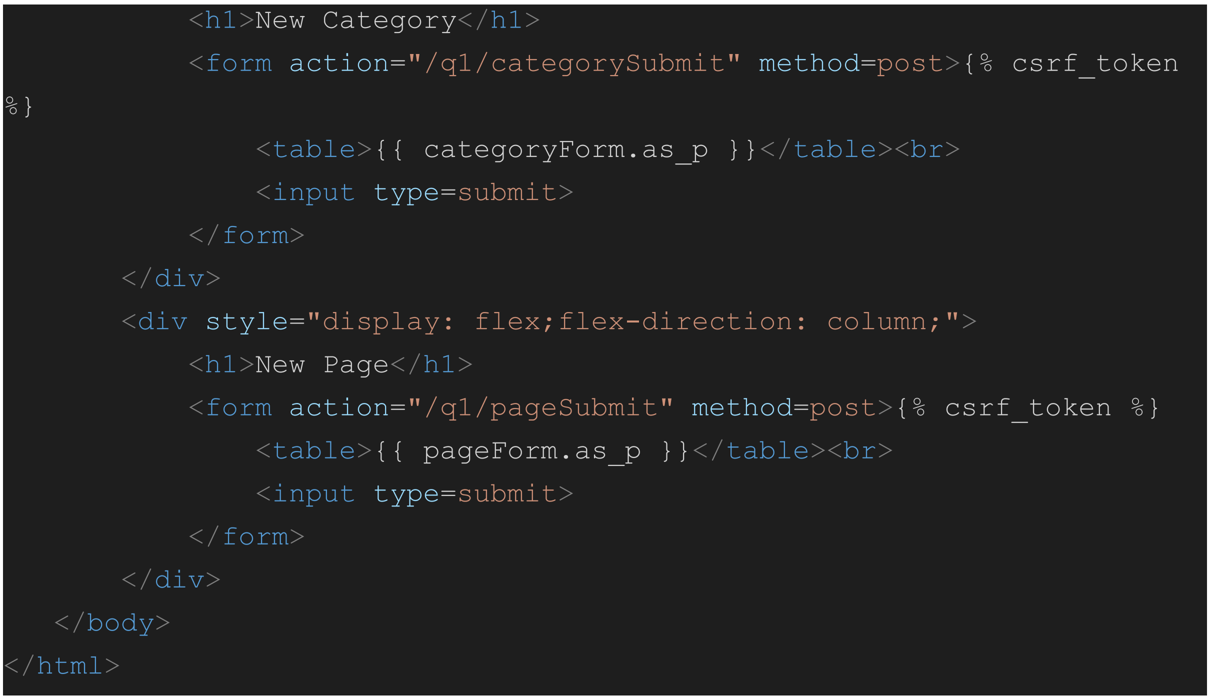
```html
<!DOCTYPE html>
```

```
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
   <head>
   <meta charset="utf-8" />
   <title>Pages List</title>
   </head>
   <body style="text-align: center;background-color:
#F7CAC9;margin-top: 20px;">
       <h1>Search pages by category</h1>
       <form action="/q1/list" method=post style="margin-bottom: 10px;"
>{% csrf_token %}
           <input type="text" placeholder="Category name"
name="category">
           <input type=submit>
       </form>
           {% for page in pages %}
           <div style="border: 1px solid #fff;width: 500px;margin:
auto;line-height: 1;">
               <h3>Title: {{page.title }}</h3>
               <h5>Page Views:{{page.views}}</h5>
               View Page: <a target="_blank" href="{% url 'updateViews'
page.id %}">{{page.title}}</a>
               <p>Category: {{ page.category.name }}
               <br> Category Visits:{{page.category.noOfVisits}} Likes:
{{page.category.noOfLikes}}</p>

           </div>
           {% endfor %}
           <div style="margin-top: 20px;">
               <a href="forms">New Page or New Category</a>
           </div>
   </body>
</html>
```

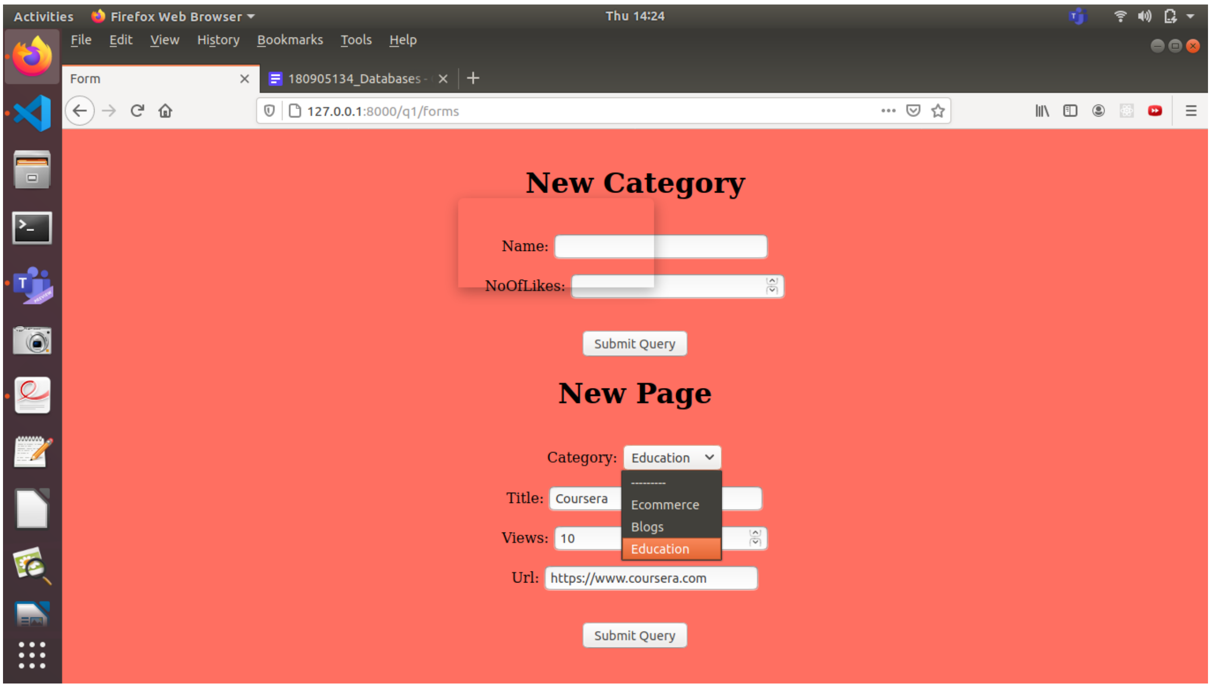
forms.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
   <head>
   <meta charset="utf-8" />
   <title>Form</title>
   </head>
   <body style="text-align: center;background-color:
#FF6F61;margin-top: 20px;">
       <div style="display: flex;flex-direction: column;">
```



```
            <h1>New Category</h1>
            <form action="/q1/categorySubmit" method=post>{% csrf_token
%}
                <table>{{ categoryForm.as_p }}</table><br>
                <input type=submit>
            </form>
        </div>
        <div style="display: flex;flex-direction: column;">
            <h1>New Page</h1>
            <form action="/q1/pageSubmit" method=post>{% csrf_token %}
                <table>{{ pageForm.as_p }}</table><br>
                <input type=submit>
            </form>
        </div>
    </body>
</html>
```

Screenshots:
Category visits+= all pages views of the category
Moreover pages and category gets updated when a page is visited



New Category

Name:

NoOfLikes:

Submit Query

New Page

Category: Education

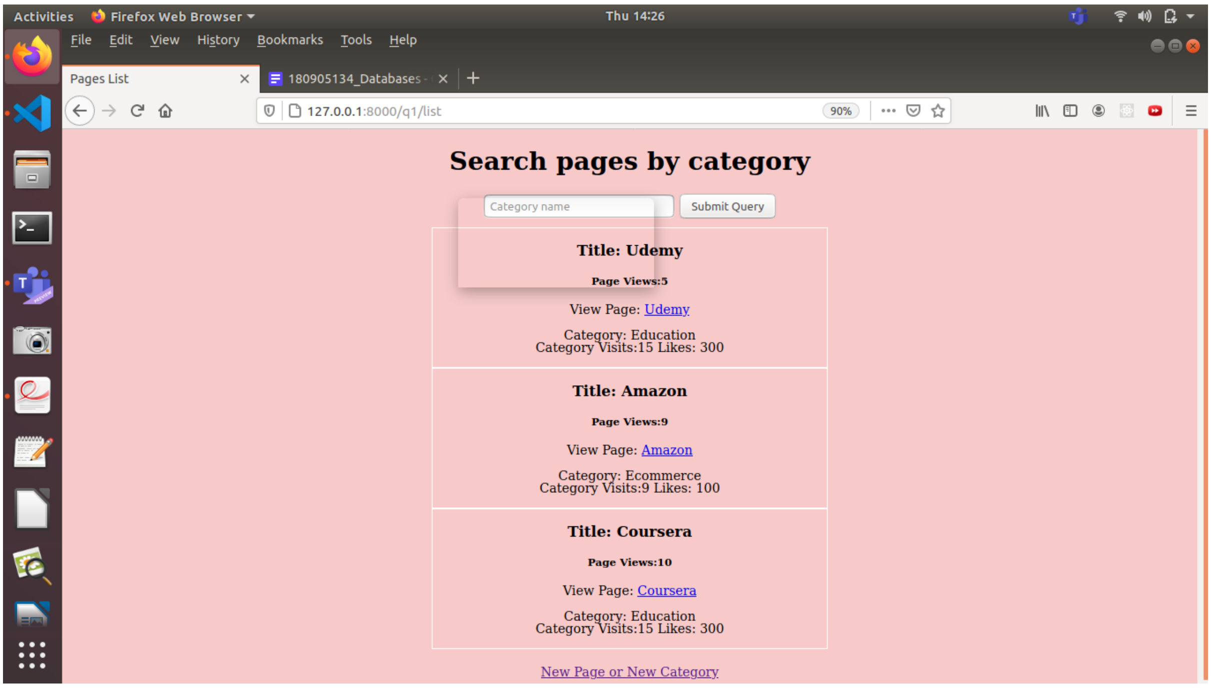
Ecommerce
Blogs
Education

Title: Coursera

Views: 10

Url: https://www.coursera.com

Submit Query



Search pages by category

Category name | Submit Query

Title: Udemy

Page Views:5

View Page: Udemy

Category: Education
Category Visits:15 Likes: 300

Title: Amazon

Page Views:9

View Page: Amazon

Category: Ecommerce
Category Visits:9 Likes: 100

Title: Coursera

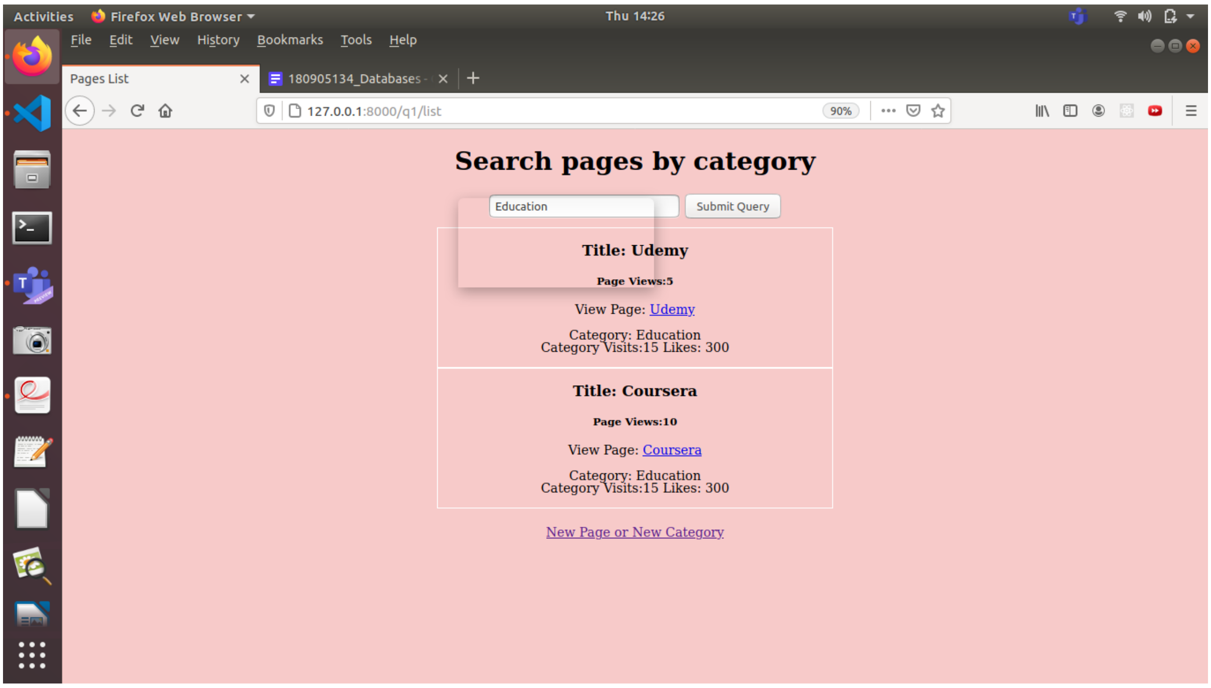
Page Views:10

View Page: Coursera

Category: Education
Category Visits:15 Likes: 300

New Page or New Category

Extra feature can search by a particular category also



Search pages by category

Education | Submit Query

Title: Udemy

Page Views:5

View Page: Udemy

Category: Education
Category Visits:15 Likes: 300

Title: Coursera

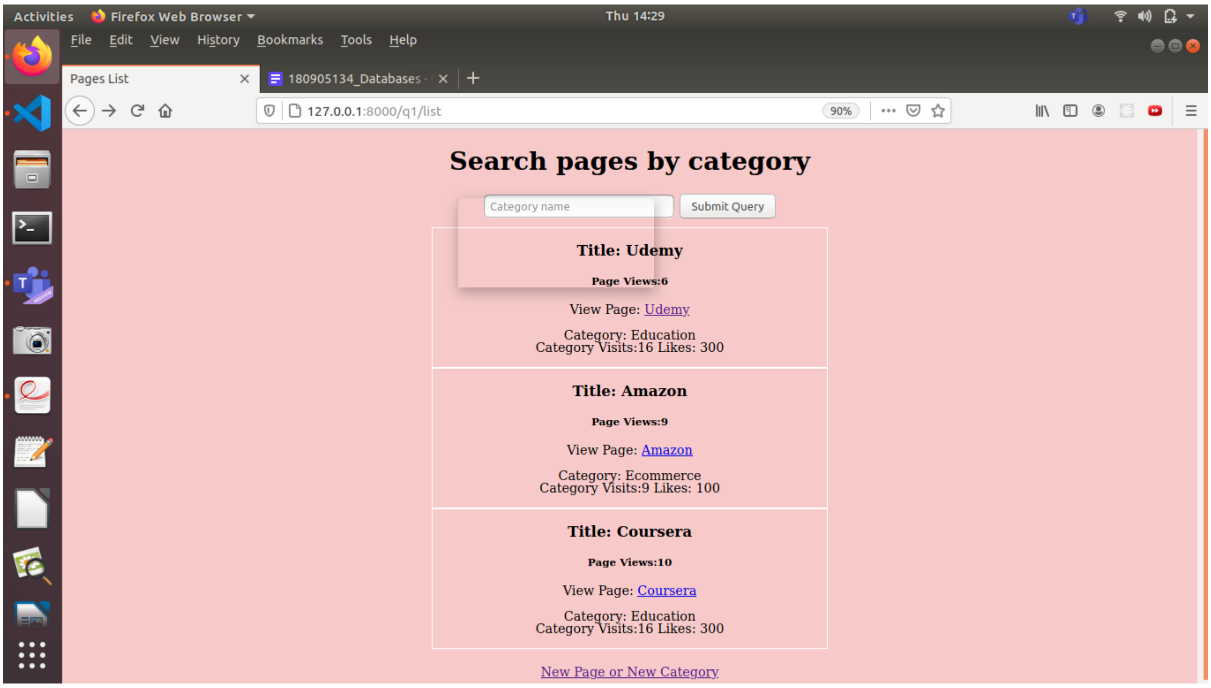
Page Views:10

View Page: Coursera

Category: Education
Category Visits:15 Likes: 300

New Page or New Category

After visiting Udemy



Search pages by category

Title: Udemy

Page Views:6

View Page: Udemy

Category: Education
Category Visits:16 Likes: 300

Title: Amazon

Page Views:9

View Page: Amazon

Category: Ecommerce
Category Visits:9 Likes: 100

Title: Coursera

Page Views:10

View Page: Coursera

Category: Education
Category Visits:16 Likes: 300

New Page or New Category

2. 2nd question

Lab7/q2/

models.py

Modeled Lives Table as PersonDetails

```
from django.db import models
from django import forms


class PersonDetails(models.Model):
   person_name = models.CharField(max_length=30)
   street = models.CharField(max_length=30)
   city = models.CharField(max_length=30)

   def __str__(self):
       return self.person_name


class Works(models.Model):
   person = models.ForeignKey(PersonDetails, on_delete=models.CASCADE)
   company_name = models.CharField(max_length=30)
   salary = models.FloatField()

   class Meta:
       ordering = ['person']
```

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


class PersonDetailsForm(forms.ModelForm):
    class Meta:
        model = PersonDetails
        fields = '__all__'


class WorksForm(forms.ModelForm):
    class Meta:
        model = Works
        fields = '__all__'
```

Created urls.py

```
from django.conf.urls import url
from django.urls import path
from . import views
urlpatterns = [
    path('list', views.personList, name='personList'),
    path('worksForm', views.worksForm, name='worksForm'),
    path('personForm', views.personForm, name='personForm'),
    path('worksSubmit', views.worksSubmit, name='worksSubmit'),
    path('personSubmit', views.personSubmit, name='personSubmit'),
]
```

views.py

```
objects.all()[:10]
    return render(request, 'q2/index.html', {'persons': persons})

def worksForm(request):
    return render(request, 'q2/worksForm.html', {'form': WorksForm()})

def personForm(request):
    return render(request, 'q2/personForm.html', {'form':
PersonDetailsForm()})

def personSubmit(request):
    if request.method == 'POST':
        form = PersonDetailsForm(request.POST)
        if form.is_valid():
            person = form.save(commit=False)
            person.save()
            return HttpResponseRedirect('/q2/list')
        else:
            return render(request, 'q2/personForm.html', {'form':
PersonDetailsForm()})

def worksSubmit(request):
    if request.method == 'POST':
        form = WorksForm(request.POST)
        if form.is_valid():
            works = form.save(commit=False)
            works.save()
            return HttpResponseRedirect('/q2/list')
        else:
            return render(request, 'q2/worksForm.html', {'form':
WorksForm()})
```

Created Folder templates/q2/
Files Created:
index.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>People List</title>
    </head>
    <body style="text-align: center;background-color:
#F7CAC9;margin-top: 20px;">
        <h1>Search people by company name</h1>
        <form action="/q2/list" method=post style="margin-bottom:
10px;">{% csrf_token %}
            <input type="text" placeholder="Company name"
name="company">
            <input type=submit>
        </form>
            {% for person in persons %}
            <div style="border: 1px solid #fff;width: 500px;margin:
auto;line-height: 1;">
                <h3>Name: {{person.person.person_name }}</h3>
                <p>Company: {{ person.company_name }}
Salary:{{person.salary}}</p>
                <p>Address: {{person.person.street}},
{{person.person.city}}</p>
```

```
           </div>
           {% endfor %}
           <div style="margin-top: 20px;">
               <a href="personForm" style="margin-right: 20px;">New
Person</a>
               <a href="worksForm">New Work Profile</a>
           </div>
   </body>
</html>
```

personForm.html

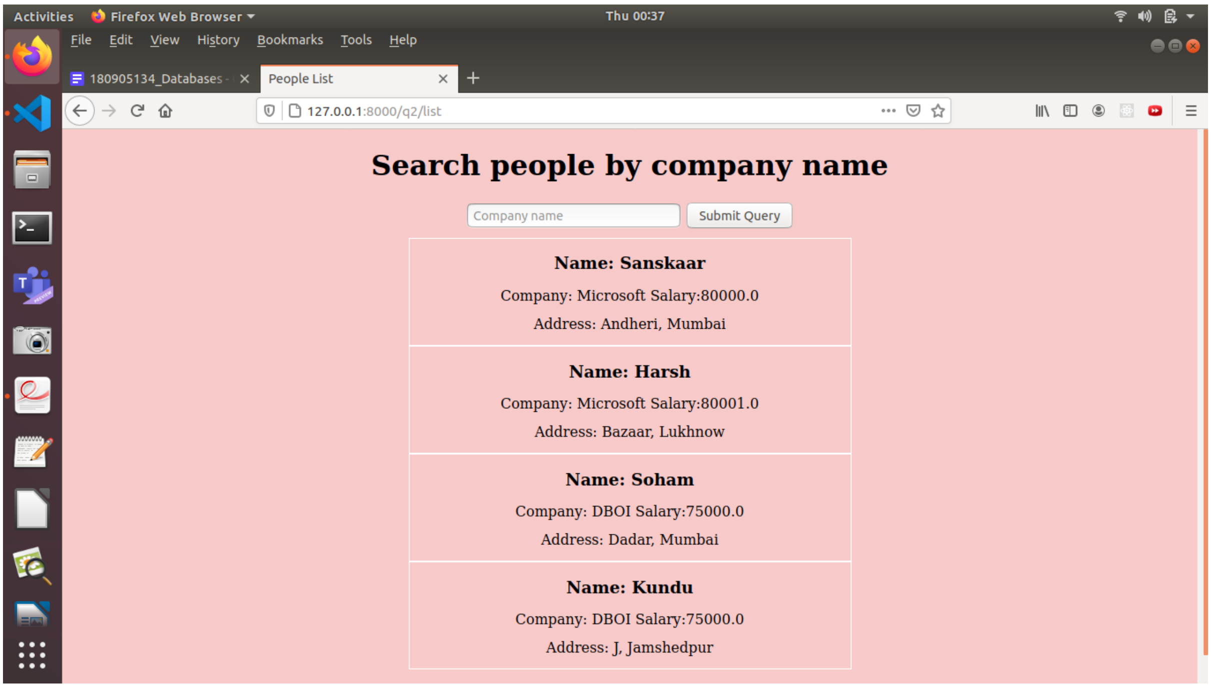
```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
   <head>
   <meta charset="utf-8" />
   <title>Person Form</title>
   </head>
   <body style="text-align: center;background-color:
#FF6F61;margin-top: 100px;">
       <div style="display: flex;flex-direction: column;">
           <h1>Enter Employee Name and Address Details</h1>
           <form action="/q2/personSubmit" method=post>{% csrf_token %}
               <table>{{ form.as_p }}</table><br>
               <input type=submit>
           </form>
       </div>
   </body>
</html>
```

worksForm.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
   <head>
   <meta charset="utf-8" />
   <title>Works Form</title>
   </head>
   <body style="text-align: center;background-color:
#FF6F61;margin-top: 100px;">
       <div style="display: flex;flex-direction: column;">
           <h1>Works Form</h1>
           <form action="/q2/worksSubmit" method=post>{% csrf_token %}
               <table>{{ form.as_p }}</table>
               <br>
```

```
                <input type=submit>
            </form>
        </div>
    </body>
</html>
```

Screenshots:



Search people by company name

Company name
Submit Query

Name: Sanskaar

Company: Microsoft Salary:80000.0

Address: Andheri, Mumbai

Name: Harsh

Company: Microsoft Salary:80001.0

Address: Bazaar, Lukhnow

Name: Soham

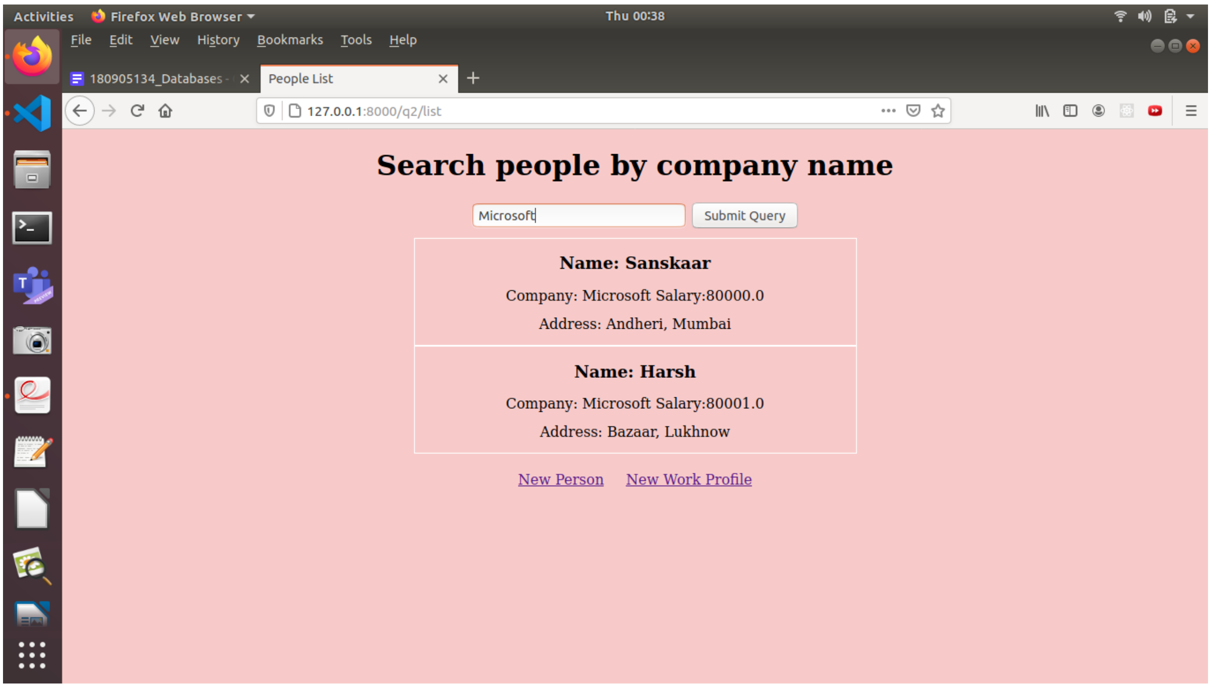
Company: DBOI Salary:75000.0

Address: Dadar, Mumbai

Name: Kundu

Company: DBOI Salary:75000.0

Address: J, Jamshedpur



Search people by company name

Microsoft
Submit Query

Name: Sanskaar

Company: Microsoft Salary:80000.0

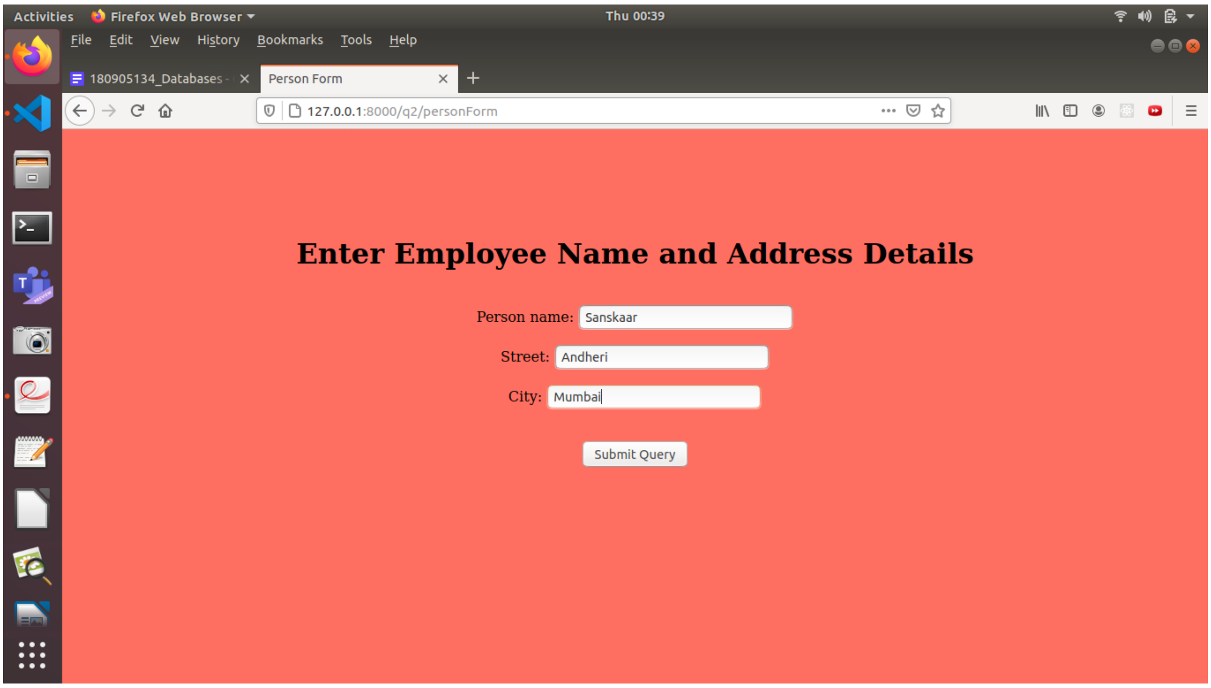
Address: Andheri, Mumbai

Name: Harsh

Company: Microsoft Salary:80001.0

Address: Bazaar, Lukhnow

New Person New Work Profile

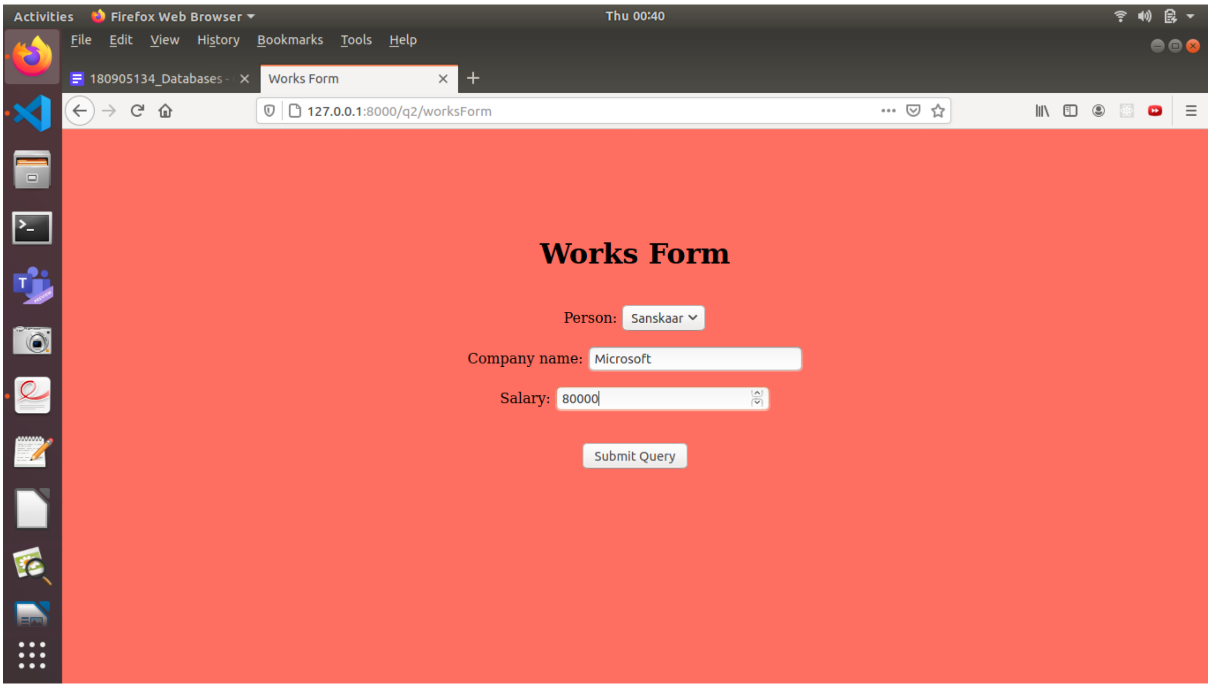

Activities
Firefox Web Browser
Thu 00:39
File Edit View History Bookmarks Tools Help
180905134_Databases
Person Form
127.0.0.1:8000/q2/personForm

Enter Employee Name and Address Details

Person name: Sanskaar

Street: Andheri

City: Mumbai


Activities
Firefox Web Browser
Thu 00:40
File Edit View History Bookmarks Tools Help
180905134_Databases
Works Form
127.0.0.1:8000/q2/worksForm

Works Form

Person: Sanskaar

Company name: Microsoft

Salary: 80000

3. 3rd question

Lab7/q3/

models.py

```
from django.db import models
from django import forms


class Publisher(models.Model):
    name = models.CharField(max_length=30)
    streetAddress = models.CharField(max_length=30)
    city = models.CharField(max_length=30)
    state = models.CharField(max_length=30)
    country = models.CharField(max_length=30)
    websiteURL = models.CharField(max_length=30)

    class Meta:
        ordering = ['name']

    def __str__(self):
        return self.name


class Author(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    email = models.EmailField()

    def __str__(self):
        return "%s %s" % (self.first_name, self.last_name)


class Book(models.Model):
    title = models.CharField(max_length=100)
    pub_date = models.DateField()
    publisher = models.ForeignKey(Publisher, on_delete=models.CASCADE)
    authors = models.ManyToManyField(Author)

    def __str__(self):
        return self.title

    class Meta:
        ordering = ['title']


class BookForm(forms.ModelForm):
    class Meta:
        model = Book
        exclude = ('pub_date',)


class AuthorForm(forms.ModelForm):
    class Meta:
        model = Author
        fields = '__all__'


class PublisherForm(forms.ModelForm):
    class Meta:
        model = Publisher
        fields = '__all__'

```

Created urls.py

```
from django.conf.urls import url
from django.urls import path
from . import views
urlpatterns = [
    path('books', views.bookList, name='bookList'),
    path('authors', views.authorList, name='authorList'),
    path('publishers', views.publisherList, name='publisherList'),
    path('bookForm', views.bookForm, name='bookForm'),
    path('publisherForm', views.publisherForm, name='publisherForm'),
    path('authorForm', views.authorForm, name='authorForm'),
    path('bookSubmit', views.bookSubmit, name='bookSubmit'),
    path('authorSubmit', views.authorSubmit, name='authorSubmit'),
    path('publisherSubmit', views.publisherSubmit,
name='publisherSubmit'),
]
```

views.py

```
from datetime import datetime
from django.http import HttpResponseRedirect
from django.shortcuts import render
from q3.models import Author, Book, Publisher, BookForm, AuthorForm,
PublisherForm


def bookList(request):
    books = Book.objects.all()[:10]
    return render(request, 'q3/books.html', {'books': books})


def authorList(request):
    authors = Author.objects.all()[:10]
    return render(request, 'q3/authors.html', {'authors': authors})


def publisherList(request):
    publishers = Publisher.objects.all()[:10]
    return render(request, 'q3/publishers.html', {'publishers':
publishers})


def bookForm(request):
    return render(request, 'q3/bookForm.html', {'form': BookForm()})


def authorForm(request):
    return render(request, 'q3/authorForm.html', {'form': AuthorForm()})


def publisherForm(request):
    return render(request, 'q3/publisherForm.html', {'form':
PublisherForm()})


def bookSubmit(request):
    if request.method == 'POST':
        form = BookForm(request.POST)
        if form.is_valid():
            product = form.save(commit=False)
            product.pub_date = datetime.now()
            product.save()
            return HttpResponseRedirect('/q3/books')
        else:
            return render(request, 'q3/bookForm.html', {'form':
BookForm()})


def publisherSubmit(request):
    if request.method == 'POST':
        form = PublisherForm(request.POST)
        if form.is_valid():
            product = form.save(commit=False)
            product.save()
            return HttpResponseRedirect('/q3/publishers')
        else:
            return render(request, 'q3/publisherForm.html', {'form': PublisherForm()})


def authorSubmit(request):
    if request.method == 'POST':
        form = AuthorForm(request.POST)
        if form.is_valid():
            product = form.save(commit=False)
            product.save()
            return HttpResponseRedirect('/q3/authors')
        else:
            return render(request, 'q3/authorForm.html', {'form': AuthorForm()})
```

Created Folder templates/q3/
Files Created:
authors.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>Authors List</title>
    </head>
    <body style="text-align: center;background-color: #F7CAC9;margin-top: 50px;">
        <h1>List of Authors</h1>
            {% for author in authors %}
            <div style="border: 1px solid #fff;width: 500px;margin: auto;">
                <p>{{ author.first_name }} {{ author.last_name }}</p>
                <p>Email: {{author.email}}</p>
            </div>
            {% endfor %}
        <a href="authorForm">New author</a>
    </body>
</html>
```

books.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>Books List</title>
    </head>
    <body style="text-align: center;background-color:
#F7CAC9;margin-top: 50px;">
        <h1>List of books</h1>
            {% for book in books %}
            <p>{{ book.title }}</p>
            <p>Publishers: {{book.publisher}}</p>
            <p>Authors: {{book.authors.0}}</p>
            <p>Published Date: {{book.pub_date}}</p>
            {% endfor %}
        <a href="bookForm">New book</a>
    </body>
</html>
```

publishers.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>Publishers List</title>
    </head>
    <body style="text-align: center;background-color:
#F7CAC9;margin-top: 50px;">
        <h1>List of publishers</h1>
        <div style="border: 1px solid #fff;width: 500px;margin: auto;">
            {% for publisher in publishers %}
            <p>{{ publisher.name }}</p>
            Address:<p>{{publisher.streetAddress}},
{{publisher.city}}<br>
                        {{publisher.state}}, {{publisher.country}}
            </p>
            <p>Website URL: {{publisher.websiteURL}}</p>
            {% endfor %}
        </div>
        <a href="publisherForm">New publisher</a>
    </body>
</html>
```

bookForm.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>Book Form</title>
    </head>
    <body style="text-align: center;background-color:
#FF6F61;margin-top: 100px;">
        <div style="display: flex;flex-direction: column;">
            <h1>Enter Book Details</h1>
            <form action="/q3/bookSubmit" method=post>{% csrf_token %}
                <table>{{ form.as_p }}</table><br>
                <input type=submit>
            </form>
        </div>
    </body>
</html>
```

authorForm.html

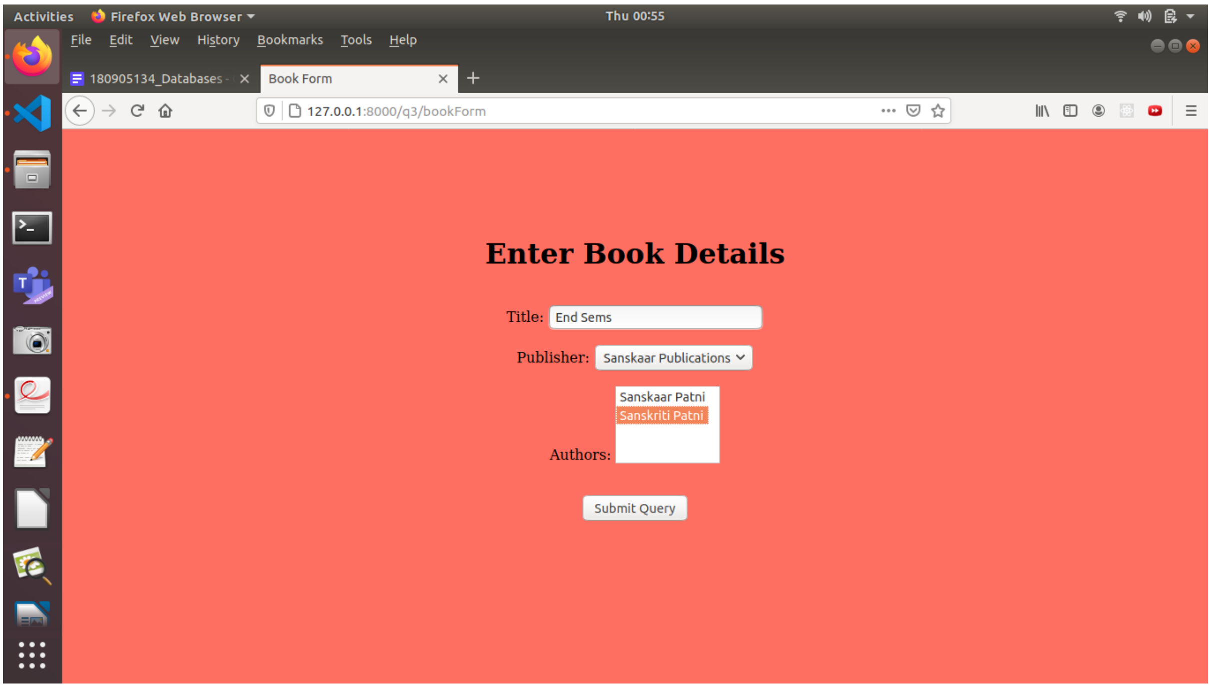
```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>Author Form</title>
    </head>
    <body style="text-align: center;background-color:
#FF6F61;margin-top: 100px;">
        <div style="display: flex;flex-direction: column;">
            <h1>Enter Author Details</h1>
            <form action="/q3/authorSubmit" method=post>{% csrf_token %}
                <table>{{ form.as_p }}</table><br>
                <input type=submit>
            </form>
        </div>
    </body>
</html>
```

publisherForm.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>Publisher Form</title>
    </head>
    <body style="text-align: center;background-color:
#FF6F61;margin-top: 100px;">
        <div style="display: flex;flex-direction: column;">
```

```
            <h1>Enter Publisher Details</h1>
            <form action="/q3/publisherSubmit" method=post>{% csrf_token %}
                <table>{{ form.as_p }}</table><br>
                <input type=submit>
            </form>
        </div>
    </body>
</html>
```

Screenshots:



127.0.0.1:8000/q3/bookForm

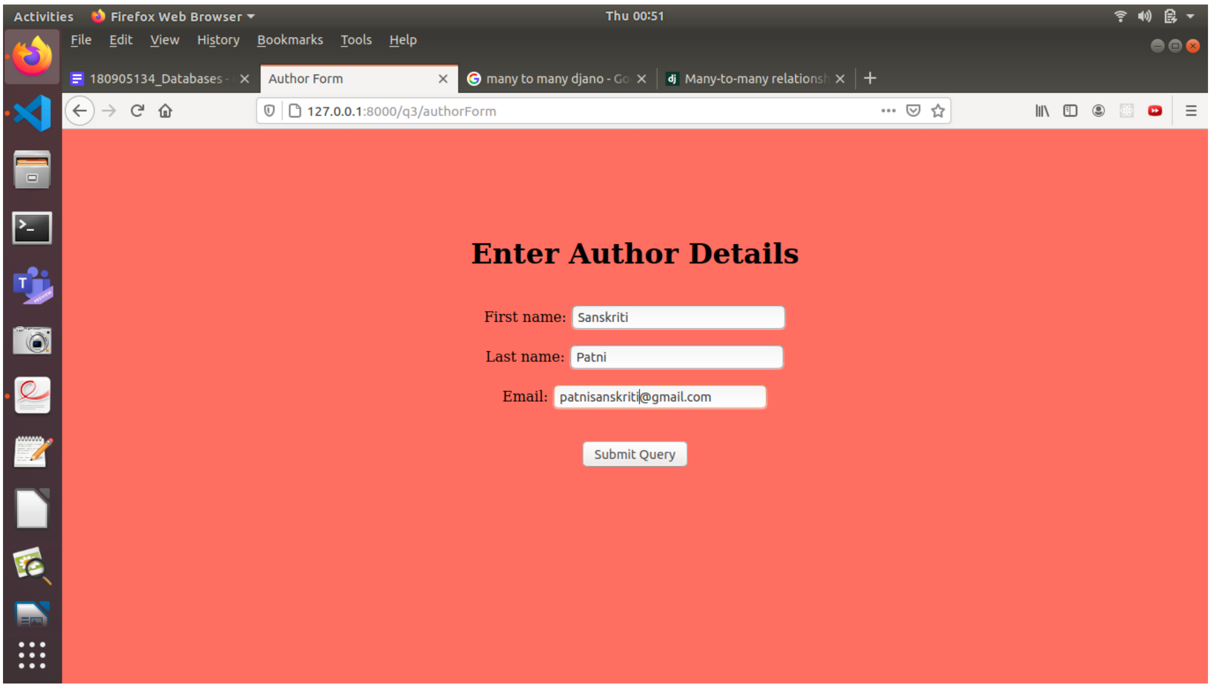
Enter Book Details

Title: End Sems

Publisher: Sanskaar Publications

Authors: Sanskaar Patni
Sanskriti Patni

Submit Query



127.0.0.1:8000/q3/authorForm

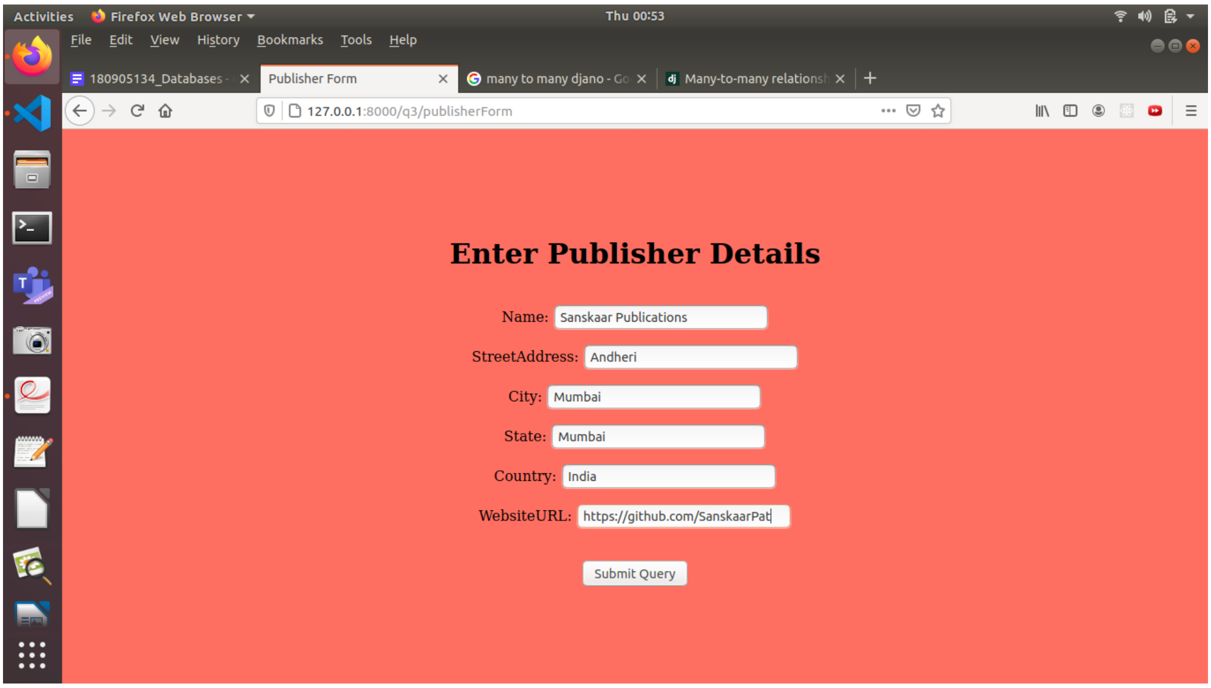
Enter Author Details

First name: Sanskriti

Last name: Patni

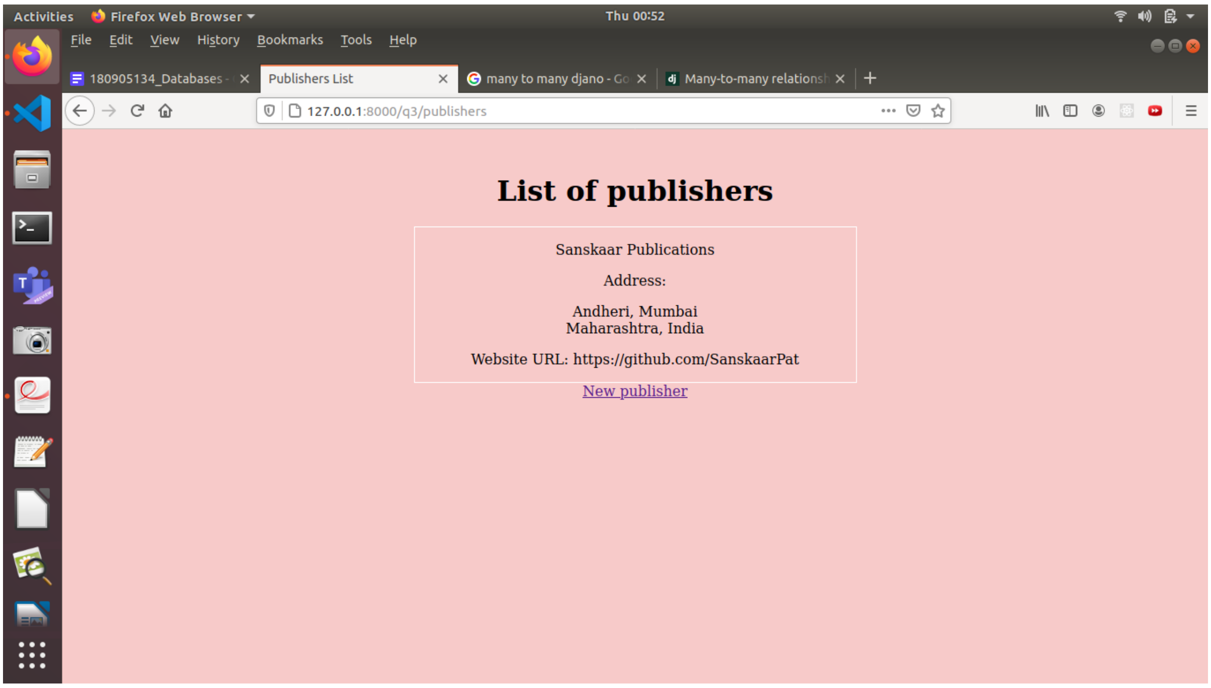
Email: patnisanskriti@gmail.com

Submit Query



Enter Publisher Details

Name:	Sanskaar Publications
StreetAddress:	Andheri
City:	Mumbai
State:	Mumbai
Country:	India
WebsiteURL:	https://github.com/SanskaarPat



List of publishers

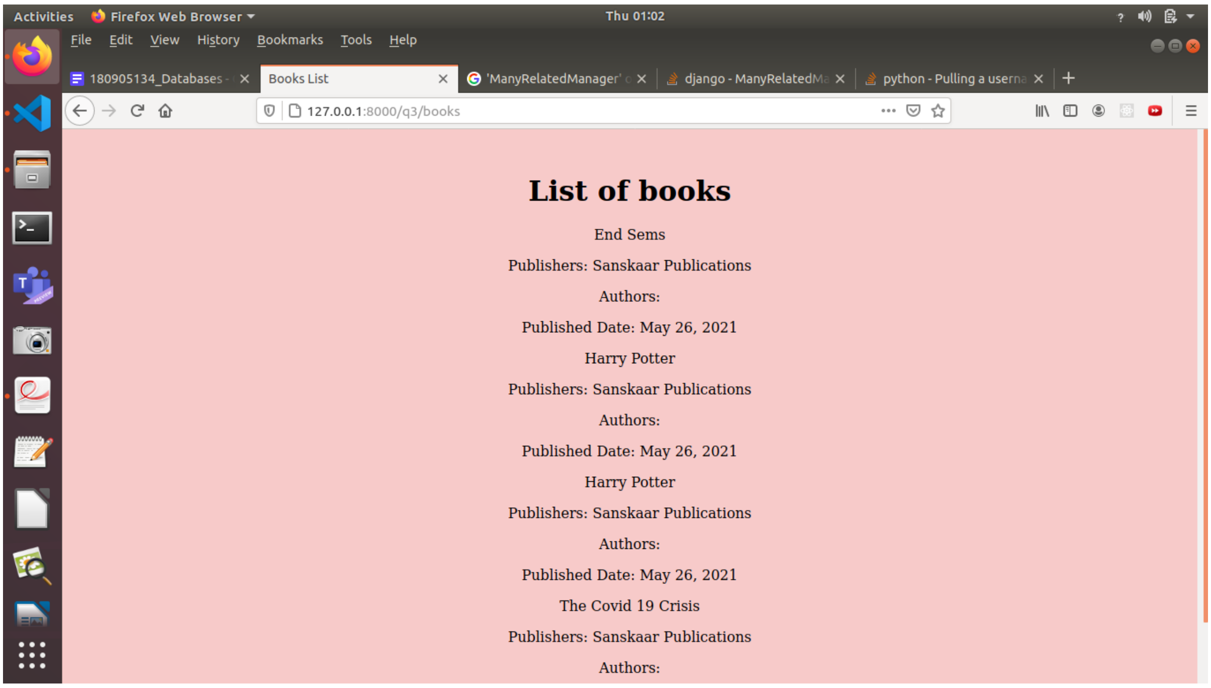
Sanskaar Publications

Address:

Andheri, Mumbai
Maharashtra, India

Website URL: https://github.com/SanskaarPat

New publisher



List of books

End Sems

Publishers: Sanskaar Publications

Authors:

Published Date: May 26, 2021

Harry Potter

Publishers: Sanskaar Publications

Authors:

Published Date: May 26, 2021

Harry Potter

Publishers: Sanskaar Publications

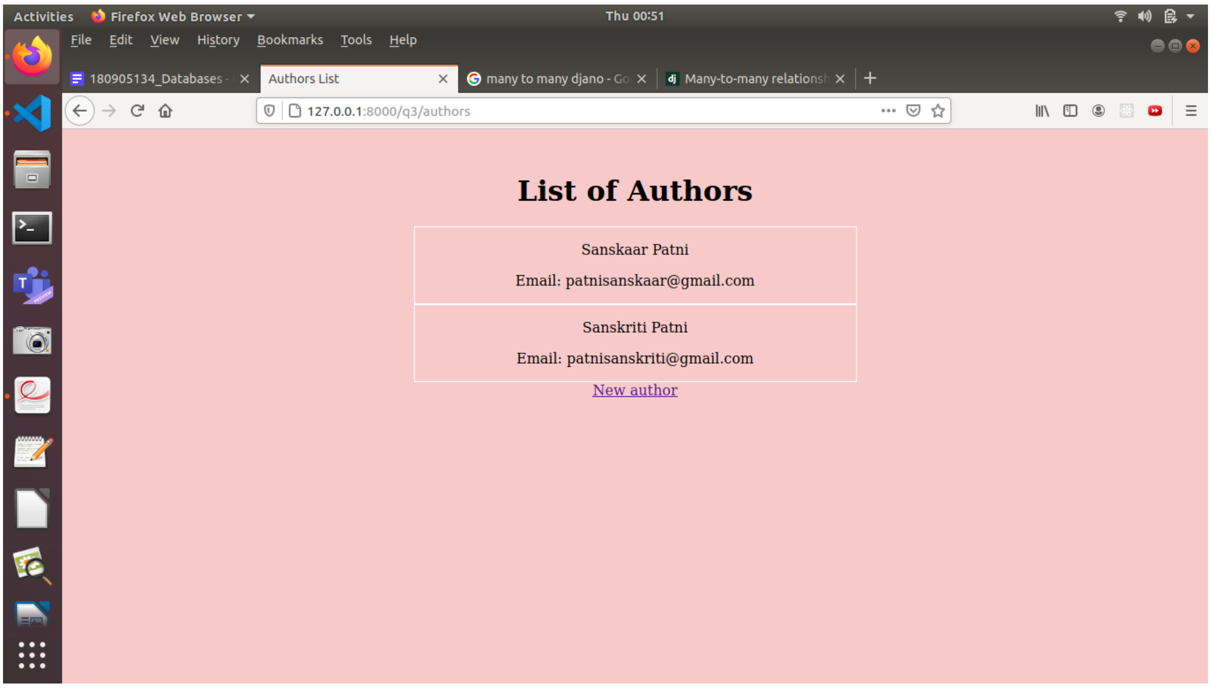
Authors:

Published Date: May 26, 2021

The Covid 19 Crisis

Publishers: Sanskaar Publications

Authors:



List of Authors

Sanskaar Patni

Email: patnisanskaar@gmail.com

Sanskriti Patni

Email: patnisanskriti@gmail.com

New author

4. 4th question

Lab7/q4/

models.py

```
from django.db import models
from django import forms


class ProductModel(models.Model):
    title = models.CharField(max_length=20)
    description = models.TextField()
    timestamp = models.DateTimeField()
    price = models.FloatField()

    def __str__(self):
        return self.title

    class Meta:
        ordering = ('-timestamp',)


class ProductForm(forms.ModelForm):
    class Meta:
        model = ProductModel
        exclude = ('timestamp',)
```

Created urls.py

```
from django.conf.urls import url
from django.urls import path
from . import views
urlpatterns = [
    path('', views.productlist, name='productlist'),
    path('submit', views.submit, name='submit'),
    path('form', views.form, name='form'),
]
```

views.py

```
from datetime import datetime
from django.http import HttpResponseRedirect
from django.shortcuts import render
from q4.models import ProductModel, ProductForm


def productlist(request):
    products = ProductModel.objects.all()[:10]
    return render(request, 'q4/index.html', {'products': products})


def form(request):
    return render(request, 'q4/form.html', {'form': ProductForm()})
```

```
def submit(request):
    if request.method == 'POST':
        form = ProductForm(request.POST)
        if form.is_valid():
            product = form.save(commit=False)
            product.timestamp = datetime.now()
            product.save()
            return HttpResponseRedirect('/q4/')
        else:
            return render(request, 'q4/form.html', {'form':
ProductForm()})
```

Created Folder templates/q4/
Files Created:

form.html

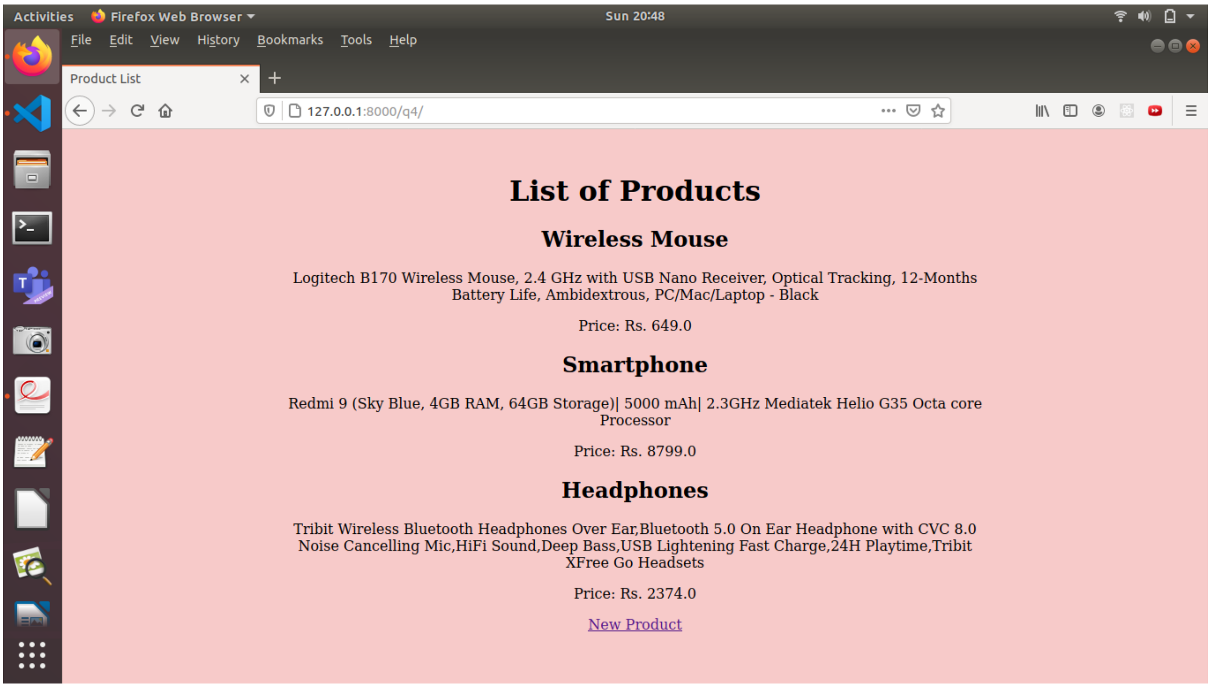
```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>Product Form</title>
    </head>
    <body style="text-align: center;background-color:
#FF6F61;margin-top: 100px;">
        <div style="display: flex;flex-direction: column;">
            <h1>Enter Product Details</h1>
            <form action="/q4/submit" method=post>{% csrf_token %}
                <table>{{ form.as_p }}</table><br>
                <input type=submit>
            </form>
        </div>
    </body>
</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta charset="utf-8" />
    <title>Product List</title>
    </head>
    <body style="text-align: center;background-color:
#F7CAC9;margin-top: 50px;">
```

```
        <h1>List of Products</h1>
            {% for product in products %}
            <h2>{{ product.title }}</h2>
            <p style="width: 800px;margin: auto;">{{ product.description
}}</p>
            <p>Price: Rs. {{ product.price }}</p>
            {% endfor %}
        <a href="form">New Product</a>
    </body>
</html>
```

Screenshots:



List of Products

Wireless Mouse

Logitech B170 Wireless Mouse, 2.4 GHz with USB Nano Receiver, Optical Tracking, 12-Months Battery Life, Ambidextrous, PC/Mac/Laptop - Black

Price: Rs. 649.0

Smartphone

Redmi 9 (Sky Blue, 4GB RAM, 64GB Storage)| 5000 mAh| 2.3GHz Mediatek Helio G35 Octa core Processor

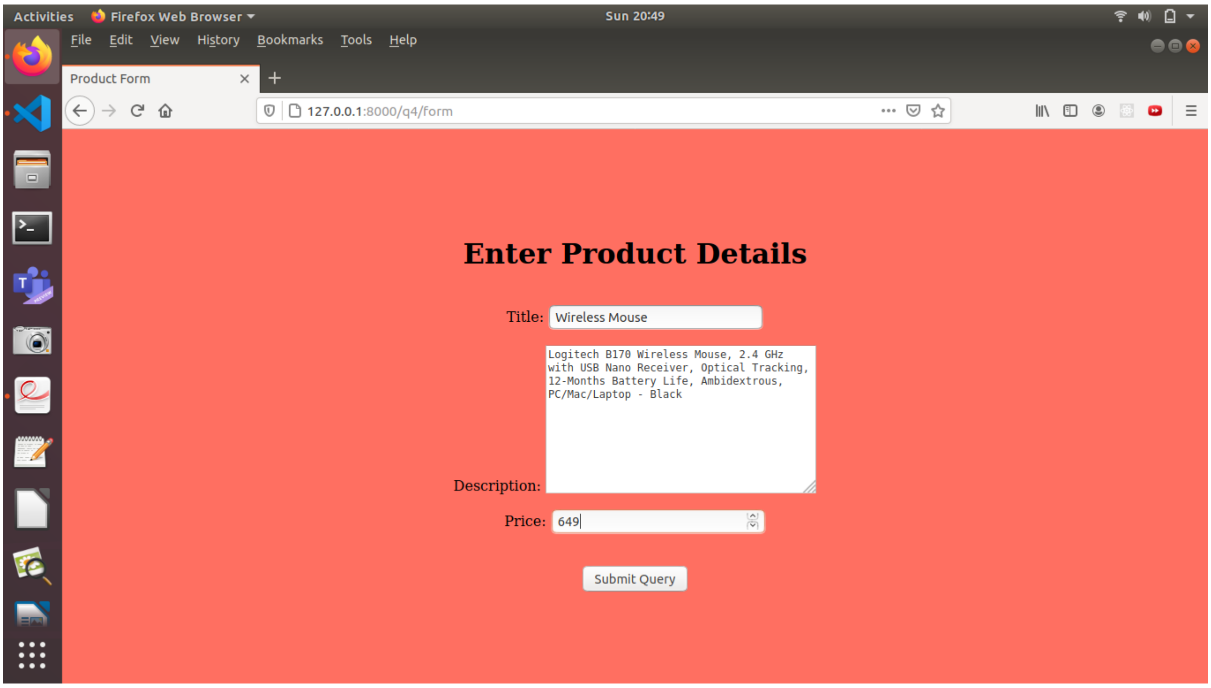
Price: Rs. 8799.0

Headphones

Tribit Wireless Bluetooth Headphones Over Ear,Bluetooth 5.0 On Ear Headphone with CVC 8.0 Noise Cancelling Mic,HiFi Sound,Deep Bass,USB Lightening Fast Charge,24H Playtime,Tribit XFree Go Headsets

Price: Rs. 2374.0

New Product



Enter Product Details

Title: Wireless Mouse

Description:

```
Logitech B170 Wireless Mouse, 2.4 GHz
with USB Nano Receiver, Optical Tracking,
12-Months Battery Life, Ambidextrous,
PC/Mac/Laptop - Black
```

Price: 649