



**INSTITUTE OF TECHNOLOGY,  
NIRMA UNIVERSITY**

## **2CSDE86 Application Development Framework**

### **PRACTICAL 5**

**Name :** Jaimik S. Chauhan

**Roll No. :** 20BCE503

**Semester :** 7<sup>th</sup>

**Teacher's Signature :**

## Aim

Continue with the social media application using the Django platform.

Create Forms File in Social Media Application

### 1. ...Continue from Practical 4

### 2. Django Forms in User App

/p3/social/forms.py

```
from django import forms
from .models import User

class CreateUserForm(forms.ModelForm):
    class Meta:
        model = User
        fields= '__all__'
```

/p3/social/views.py

```
from http.client import HTTPResponse
from django.shortcuts import render
from django.views.generic import ListView, DetailView
from django.views.generic.edit import CreateView, UpdateView, DeleteView,
FormView
from .models import User
from django.urls import reverse_lazy
from django.http import HttpResponse
from django.contrib.admin.widgets import AdminDateWidget
from social.forms import CreateUserForm
from django.shortcuts import render, redirect
from .models import User
from django.contrib import messages
from django import forms

# Create your views here.

# Base View
def home(request):
    # template_name = 'home.html'
    return render(request, 'templates/home.html')

# List Users View
class ListUser(ListView):
    model = User
    query_results = User.objects.all()
    template_name = 'ListUsers.html'

class UserDetail(DetailView):
    model = User
    template_name = 'UserDetail.html'

# Create User
class CreateUser(CreateView):
    # model= User
    form_class = CreateUserForm
    # fields = '__all__'
    template_name = 'CreateUser.html'
    success_url = reverse_lazy('ListUsers')

    # to display date picker widget in form
    def get_form(self, form_class=None):
        form = super(CreateUser, self).get_form(form_class)
        form.fields['birthdate'].widget = AdminDateWidget(attrs={'type':
'date'})
        return form

# Update User View
class UpdateUser(UpdateView):
    model = User
    # fields = '__all__'
    form_class = CreateUserForm
    template_name = 'UpdateUser.html'
    success_url = reverse_lazy('ListUsers')
```

```
# Remove User View
class RemoveUser(DeleteView):
    model = User
    form_class = CreateUserForm
    template_name = 'RemoveUser.html'
    success_url = reverse_lazy('ListUsers')
```

## Output :

Social Media App   Home (current)   Profile   Posts

### Registered User

[Create User](#)

Name	Email	Phone	BirthDate	Gender	Edit	Delete	Detail
Jaimik	jaimik@gmail.com	8401563076	April 6, 2002	Male	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Details</a>
Jaimik Chauhan	jaimikchauhan10@gmail.com	08401563076	Nov. 16, 2022	Male	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Details</a>
Anil Bhai	jaimik@gmail.com	8401563076	Oct. 16, 2022	Male	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Details</a>

Output 1. 1 Home Page For User

Social Media App Home (current) Profile Posts

## User Sign In Page

Name:

Email:

Password:

Phone:

Username:

Date:

Gender:

Is active:  1->active, 0->inactive

Output 1. 2 User Creation Page

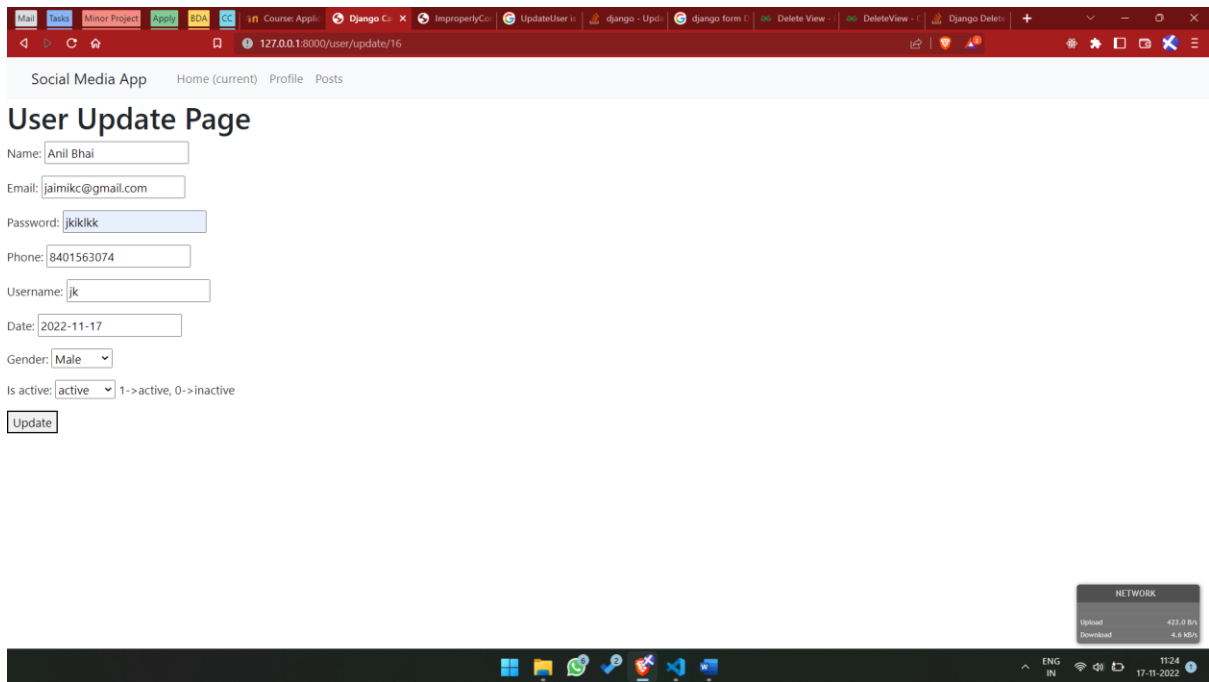
Social Media App Home (current) Profile Posts

## Registered User

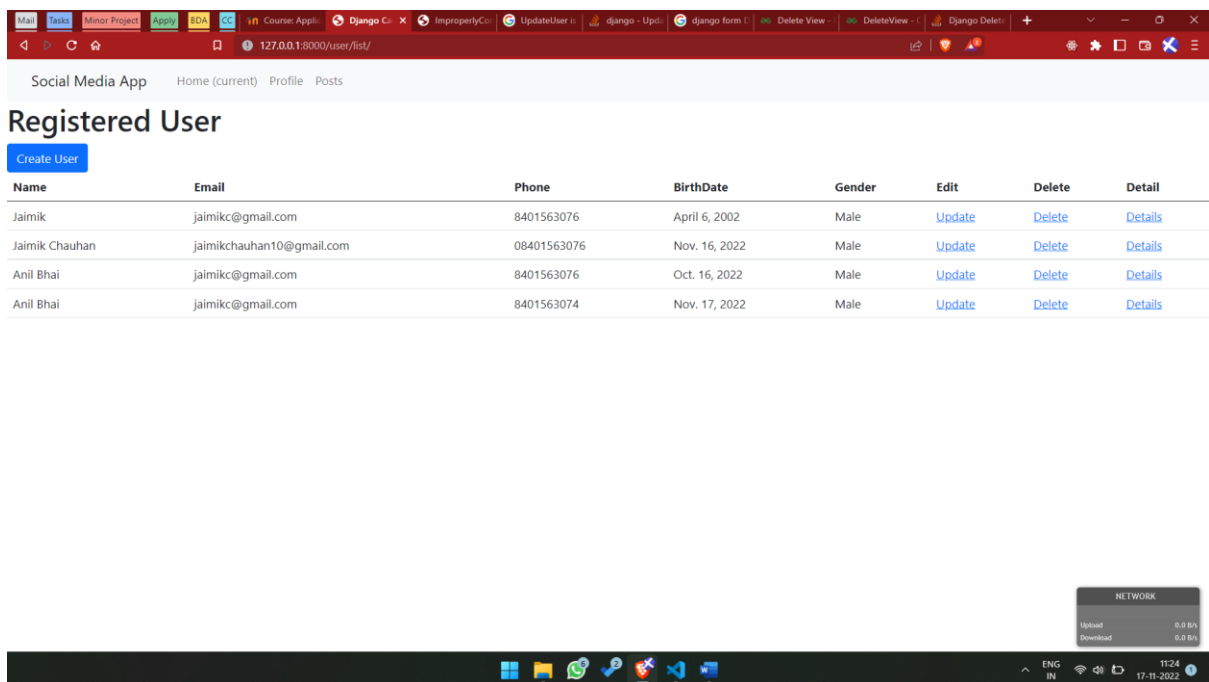
[Create User](#)

Name	Email	Phone	BirthDate	Gender	Edit	Delete	Detail
Jaimik	jaimik@gmail.com	8401563076	April 6, 2002	Male	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Details</a>
Jaimik Chauhan	jaimikchauhan10@gmail.com	08401563076	Nov. 16, 2022	Male	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Details</a>
Anil Bhai	jaimik@gmail.com	8401563076	Oct. 16, 2022	Male	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Details</a>
Anil Bhai	jaimik@gmail.com	8401563076	Nov. 17, 2022	Male	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Details</a>

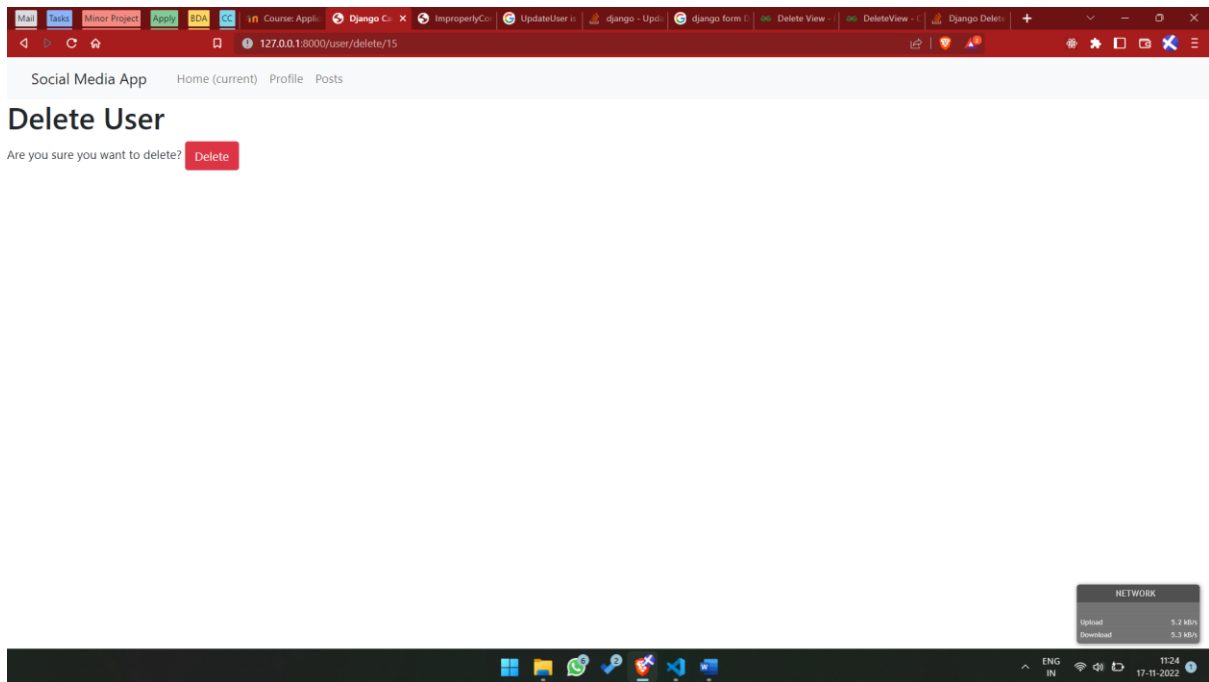
Output 1. 3 New User Created details



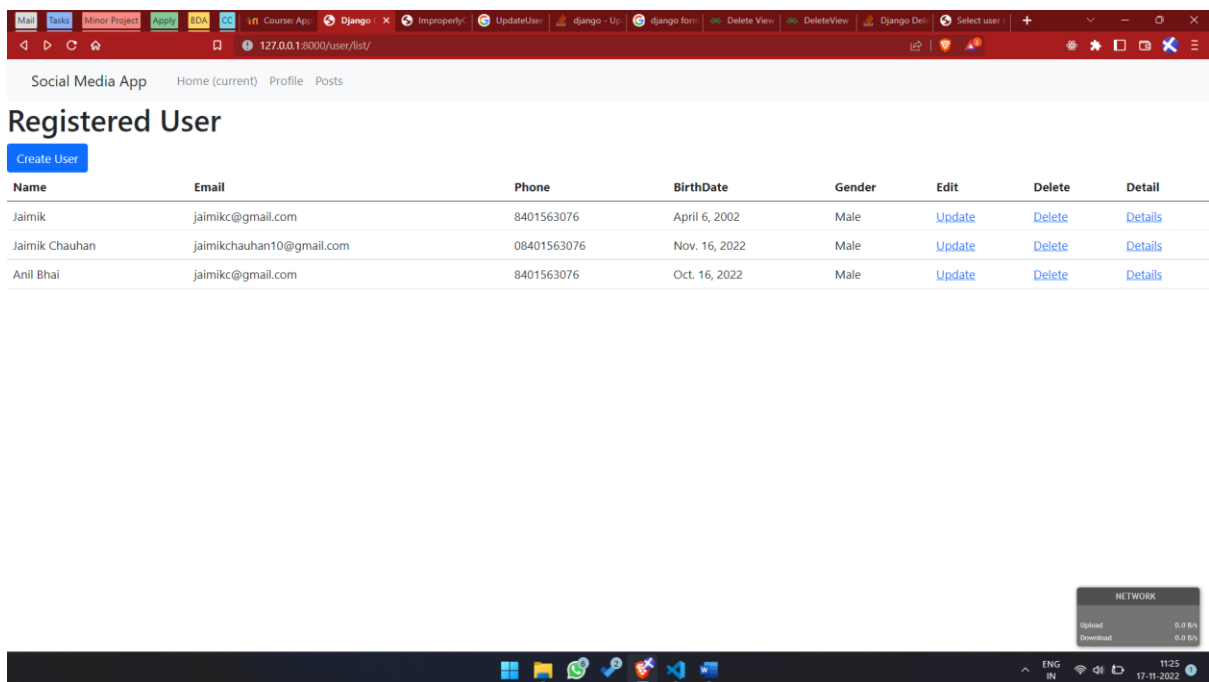
Output 1. 4 Update User Details Page



Output 1. 5 After Updating User Details



Output 1. 6 Delete User Page



Output 1. 7 After Deletion of User

### 3. Forms.py in Post App

/p3/posts/forms.py

```
from django import forms
from .models import Posts

class CreatePostForm(forms.ModelForm):
    class Meta:
        model = Posts
        fields= '__all__'
```

/p3/posts/views.py

```
from django.shortcuts import render
from django.shortcuts import render
from django.views.generic import ListView, DetailView
from django.views.generic.edit import CreateView, UpdateView, DeleteView
from .models import Posts
from django.urls import reverse_lazy
from django.http import HttpResponseRedirect
from .forms import CreatePostForm

class ListPosts(ListView):
    model = Posts
    query_results = Posts.objects.all()
    template_name = "ListPosts.html"

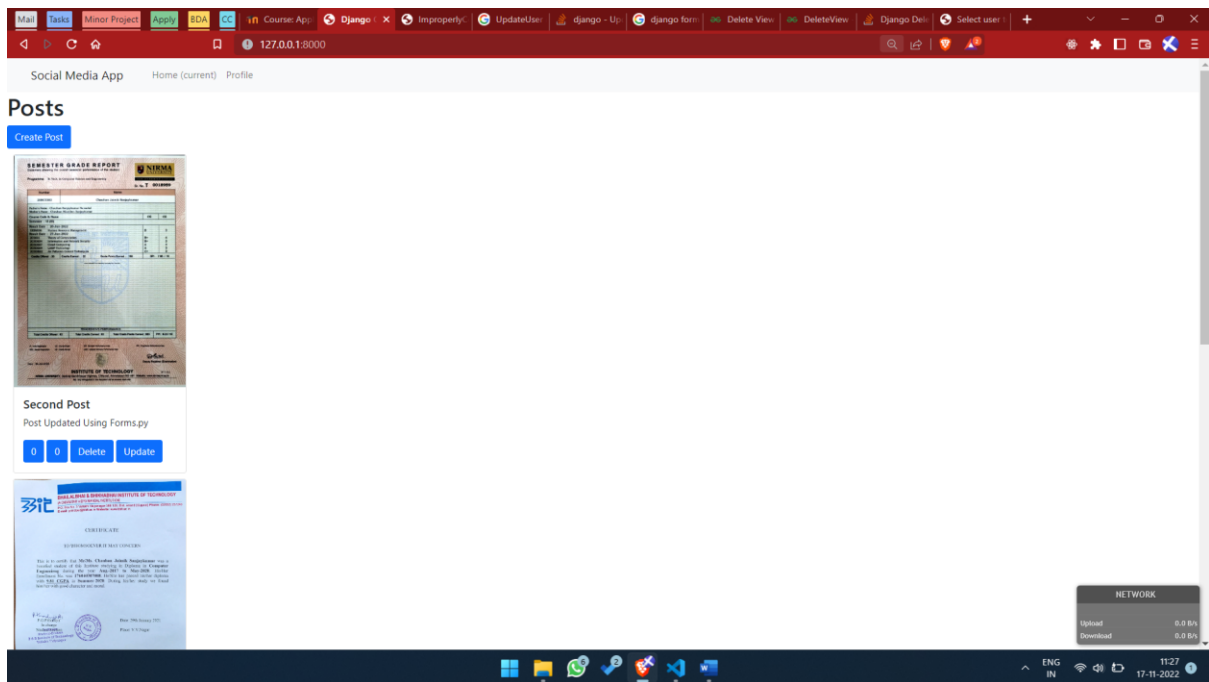
class CreatePost(CreateView):
    # model = Posts
    # fields = '__all__'
    form_class = CreatePostForm
    template_name = 'CreatePost.html'
    success_url = reverse_lazy('ListPosts')

class UpdatePost(UpdateView):
    model = Posts
    # fields = '__all__'
    form_class = CreatePostForm
    template_name = 'UpdatePost.html'
    success_url = reverse_lazy('ListPosts')

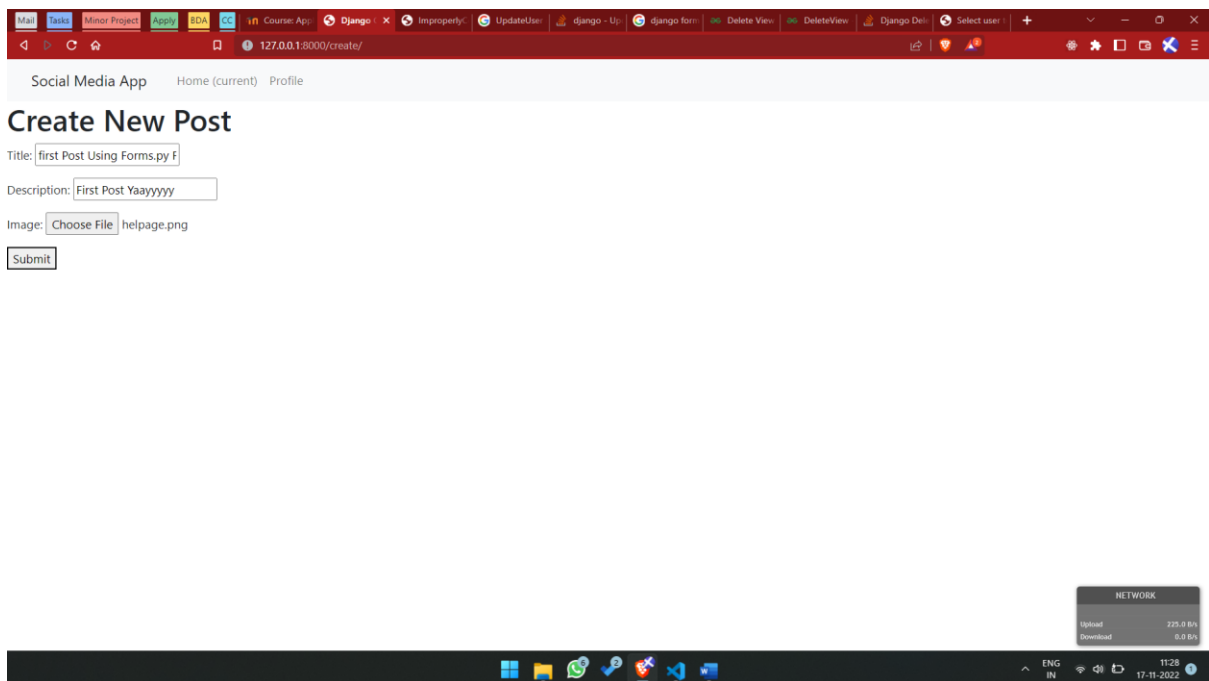
class RemovePost(DeleteView):
    model= Posts
    form_class = CreatePostForm
    template_name = 'RemovePost.html'
    success_url = reverse_lazy('ListPosts')
```



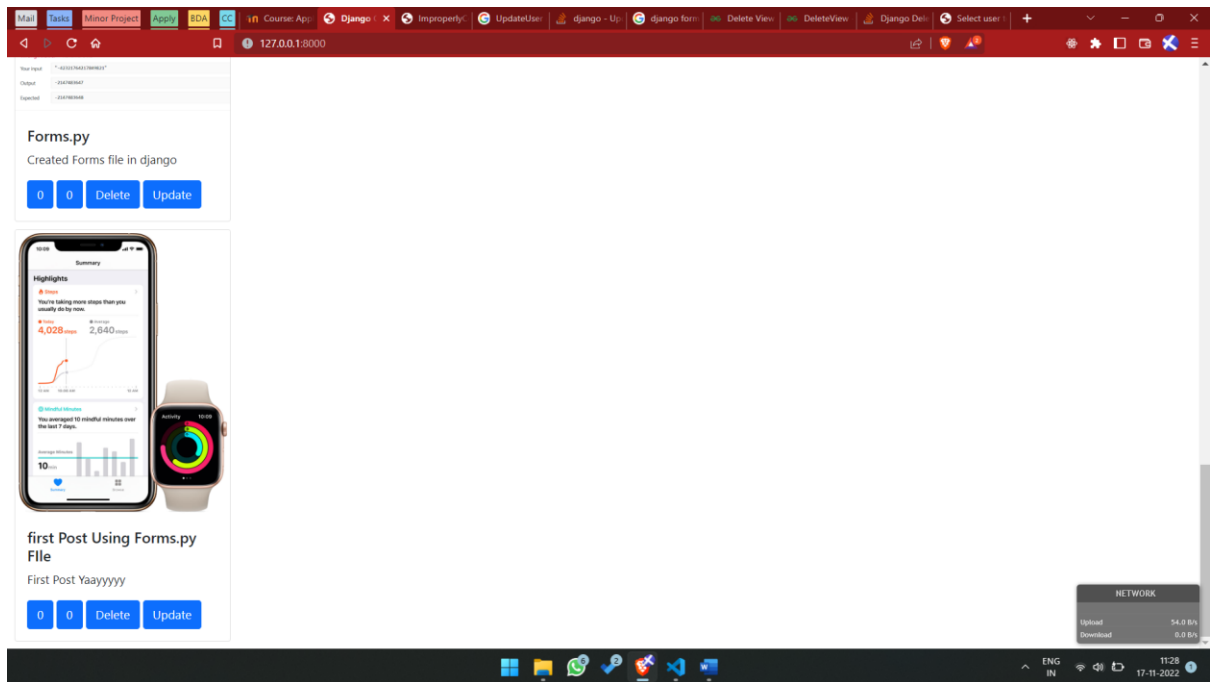
## Output :



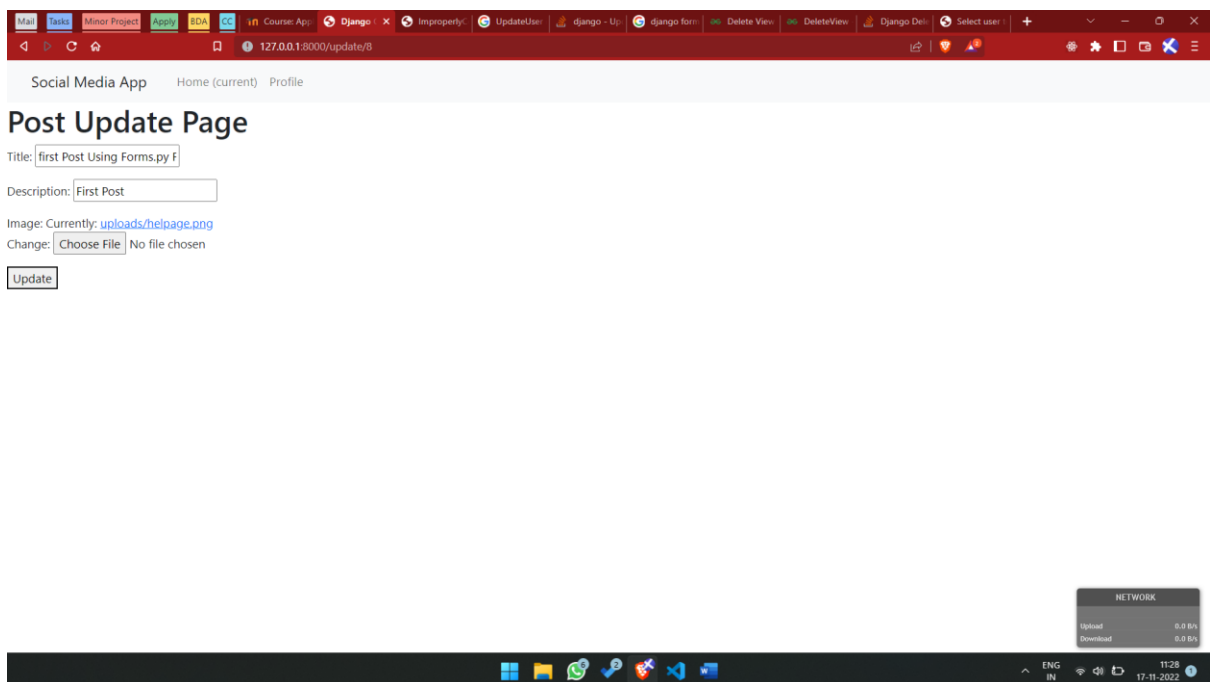
Output 2.1 Post Home Page



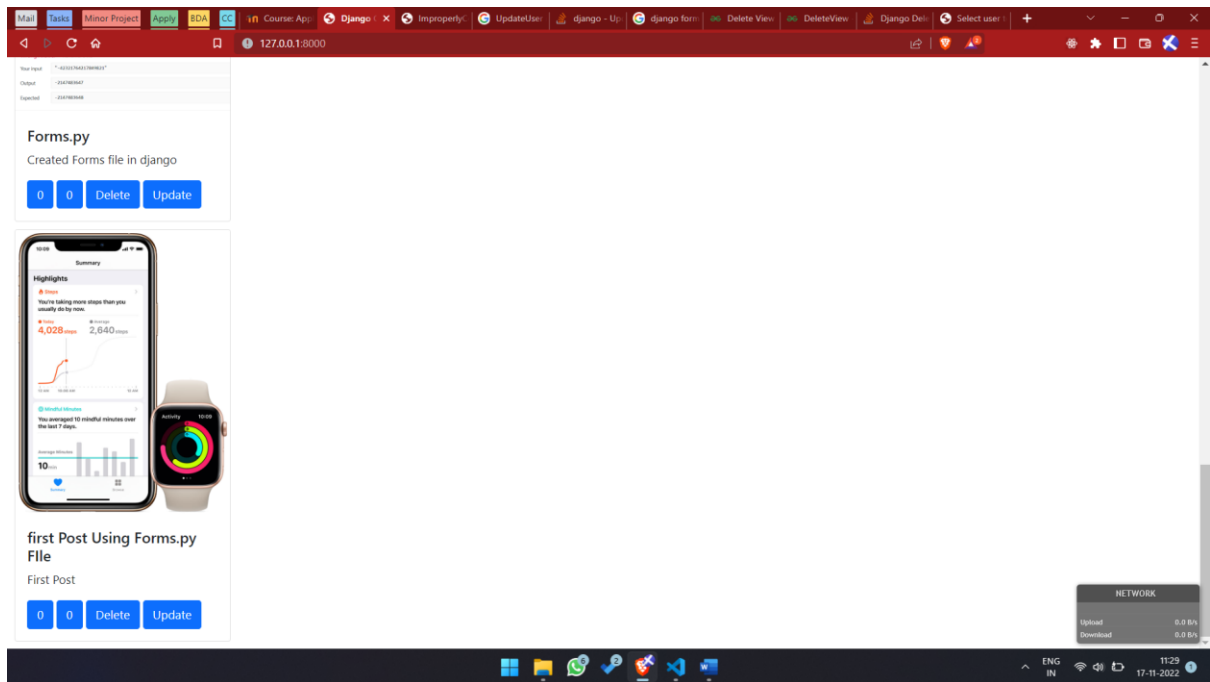
Output 2. 2 Create New Post Form



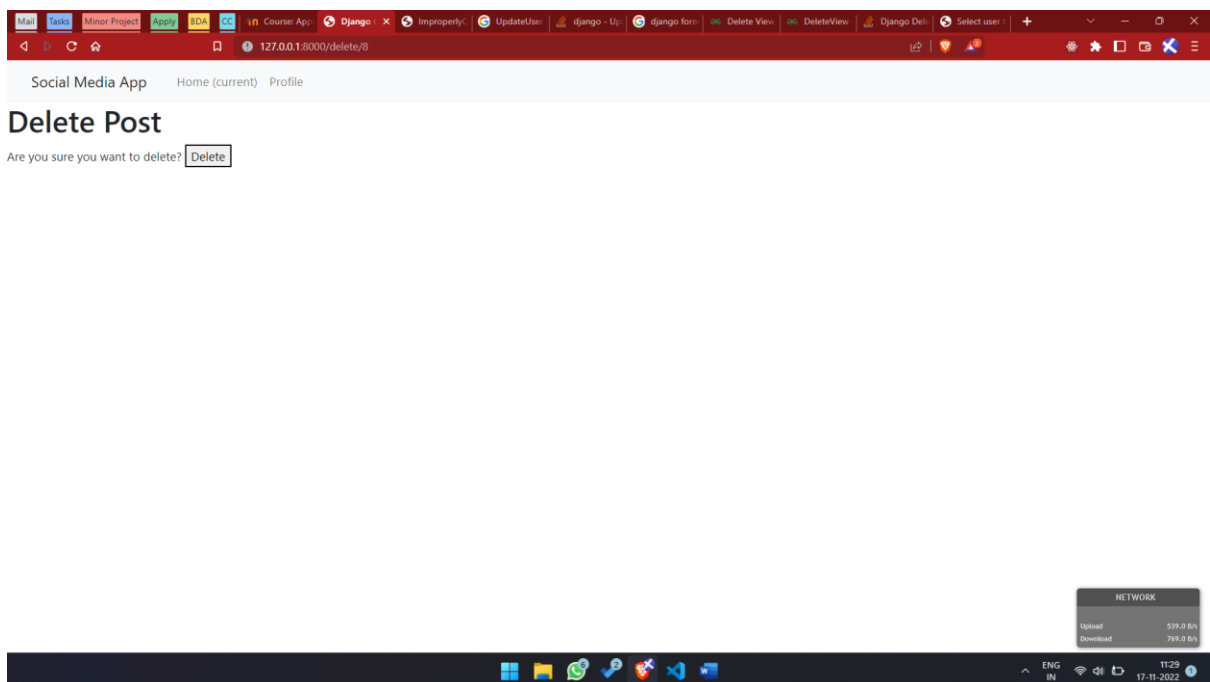
Output 2. 3 Post Created



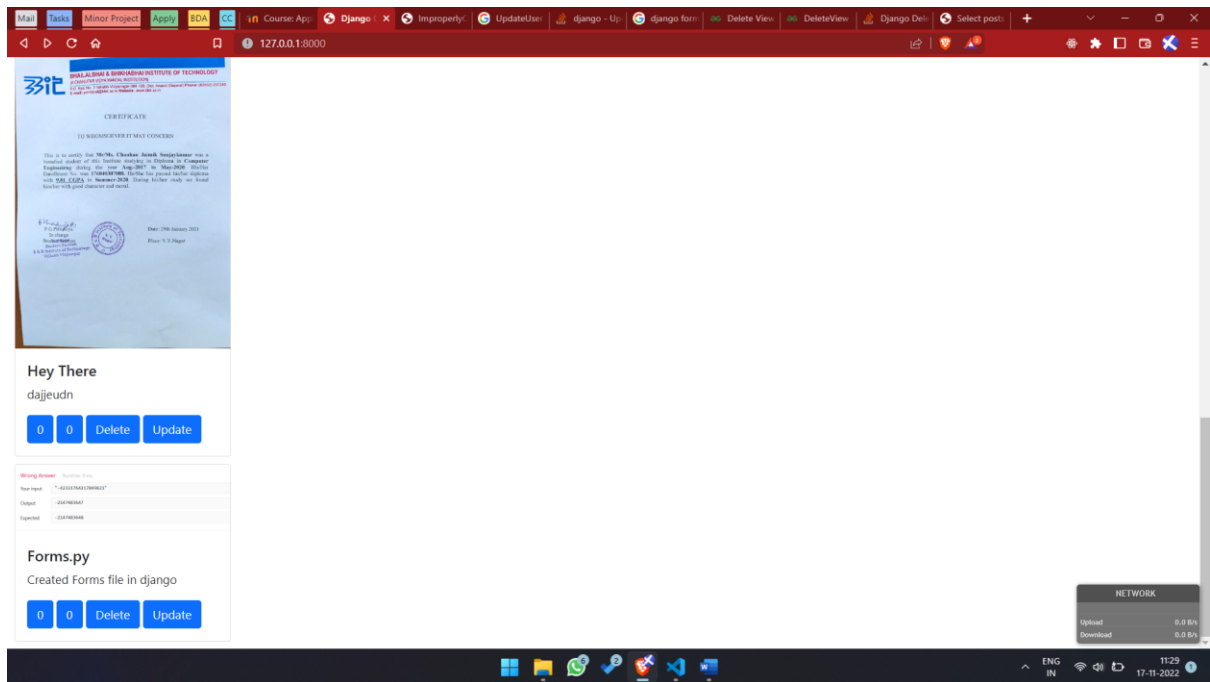
Output 2. 4 Update Post Page



Output 2. 5 Post Updated



Output 2. 6 Delete Post page



Output 2. 7 After Post Deletion, Home Page