

**INSTITUTE OF TECHNOLOGY,**

**NIRMA UNIVERSITY**

**2CSDE86 Application Development Framework**

**PRACTICAL 6**

**Name :** Jaimik S. Chauhan

**Roll No. :** 20BCE503

**Semester :** 7th

**Teacher’s Signature :**

**Aim**

Continue with the social media application using the Django platform.

Admin module customization hands-on. After hands-on perform following tasks for the social media application.

Design an admin module with dashboard and grant management privileges like admin can update and delete the content of any user.

Preform CRUD operation and connect database with your social media website. Store all the data and retrieve the data according the user.

1. **…Continue from Practical 5**
2. **Admin Module Customisation on Users**

## /p3/social/models.py

import datetime

from django.db import models

# Create your models here.

class User(models.Model):

    CHOICE = (

        ("Male", "Male"),

        ("Female", "Female")

    )

    name = models.CharField(max\_length=250)

    email = models.EmailField()

    password = models.CharField(max\_length=18)

    phone = models.CharField(max\_length=200)

    username = models.CharField(max\_length=20)

    birthdate = models.DateField("Date",default=datetime.date.today())

    gender = models.CharField(choices=CHOICE, max\_length=100)

    is\_active = models.IntegerField(

        default = 1,

        blank = True,

        null = True,

        help\_text='1->active, 0->inactive',

        choices =((1, 'active'), (0,'inactive'))

    )

## /p3/social/admin.py

from django.contrib import messages

from django.contrib import admin

# jaimik : Admin@123

# Register your models here.

from .models import User

class UserAdmin(admin.ModelAdmin):

    list\_display = ('name', 'username', 'email', 'birthdate', 'gender', 'is\_active')

    ordering = ['name', 'birthdate']

    def active(self, obj):

        return obj.is\_active == 1

    active.boolean = True

    def make\_active(modeladmin, request, queryset):

        queryset.update(is\_active=1)

        messages.success(request, "Selected Record(s) Marked is Active Succesfully!!!")

    def make\_inactive(modeladmin, request, queryset):

        queryset.update(is\_active=0)

        messages.success(request, "Selected Record(s) Marked is Inactive Succesfully!!!")

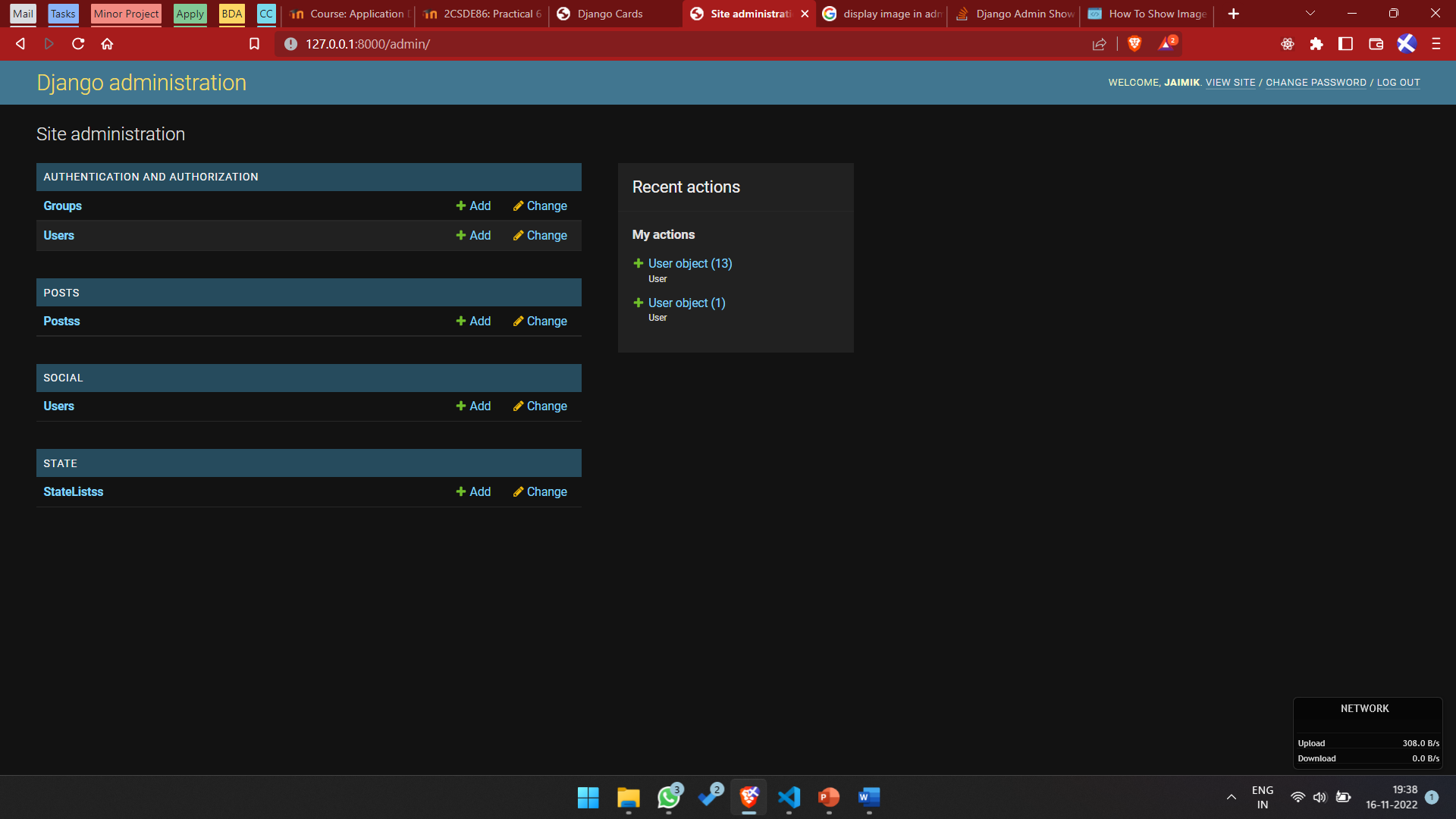
    admin.site.add\_action(make\_active,"Make Active")

    admin.site.add\_action(make\_inactive,"Make Inactive")

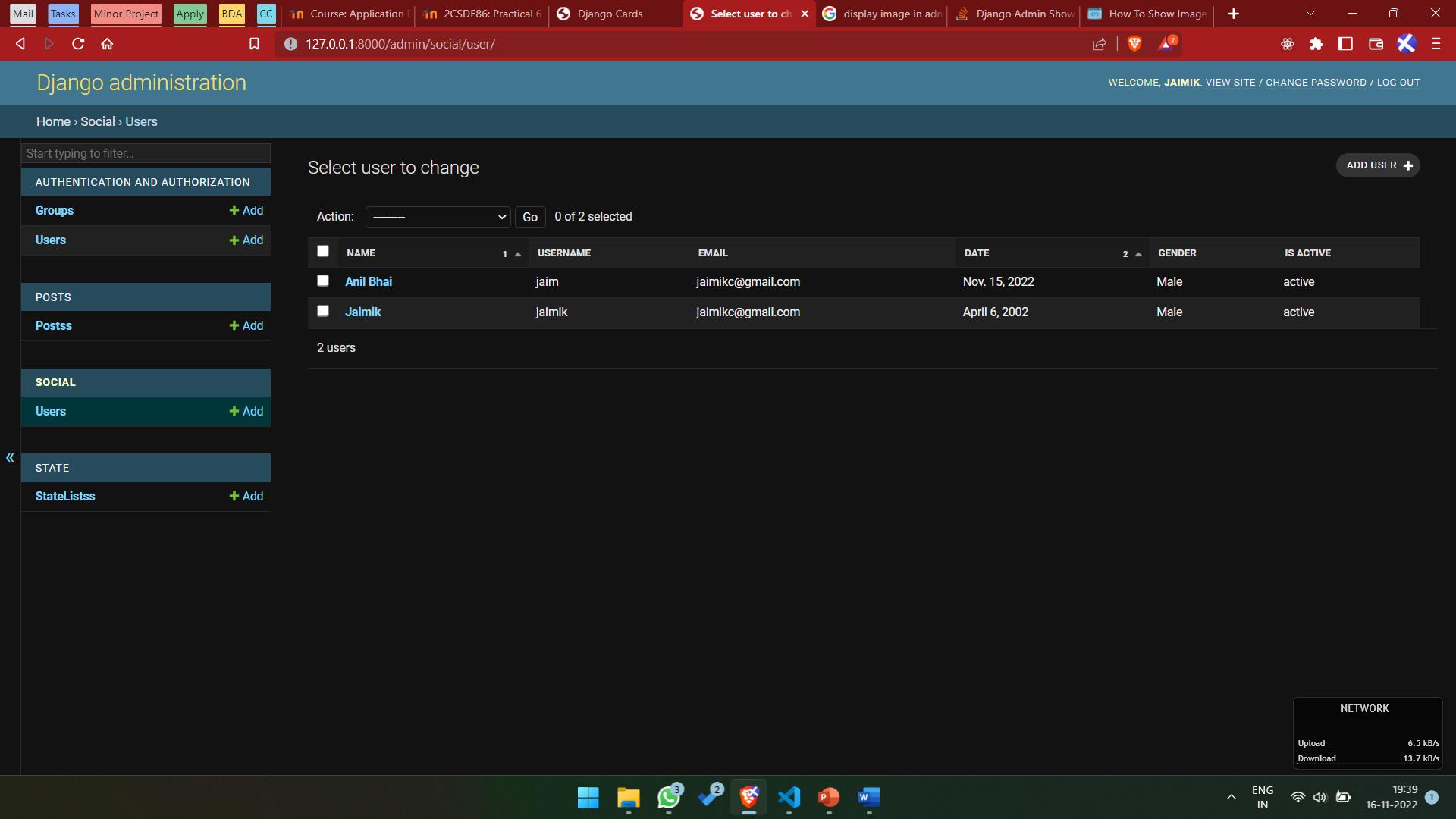
    db\_table = "New Users"

admin.site.register(User, UserAdmin)

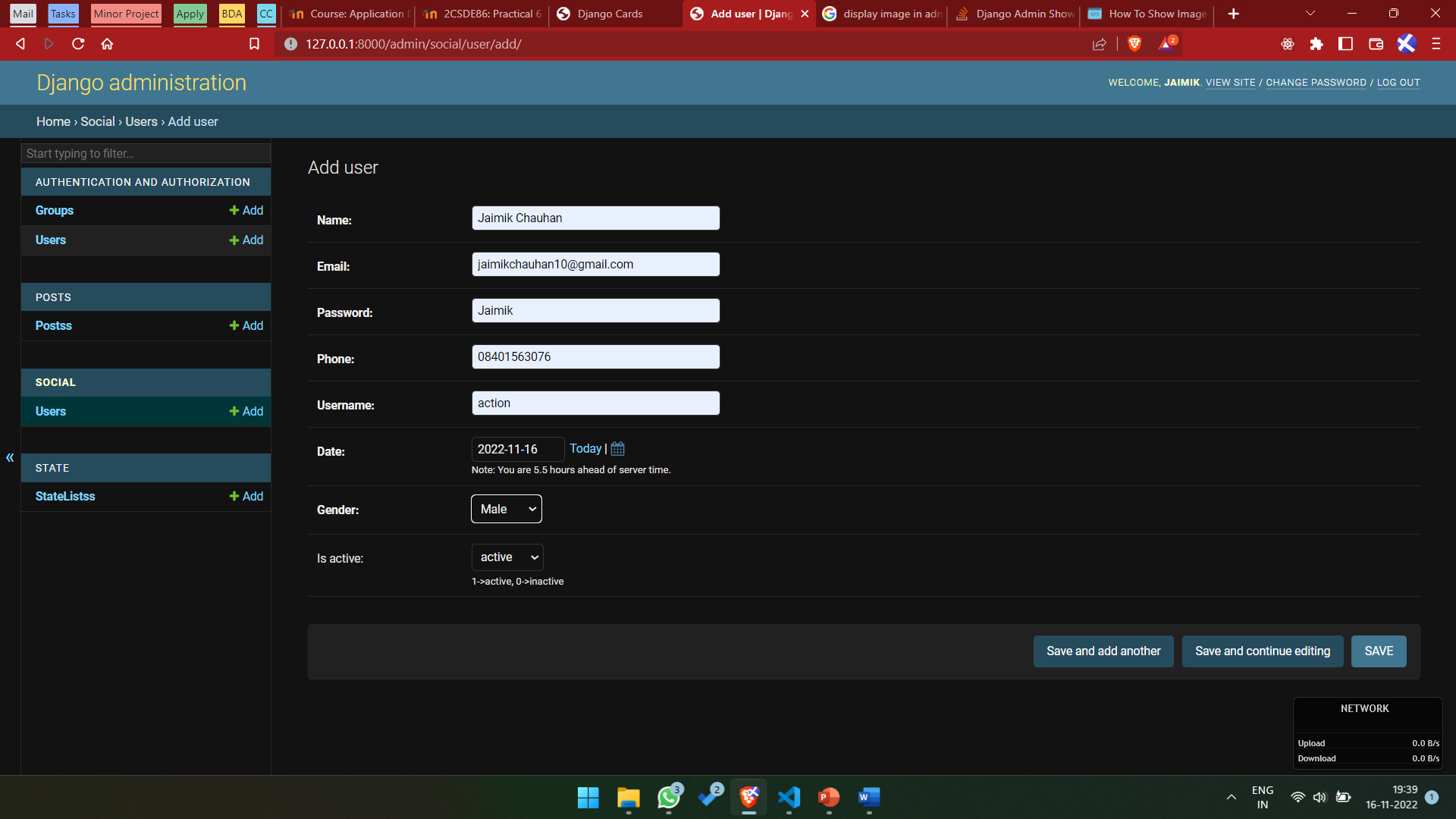
**Output :**



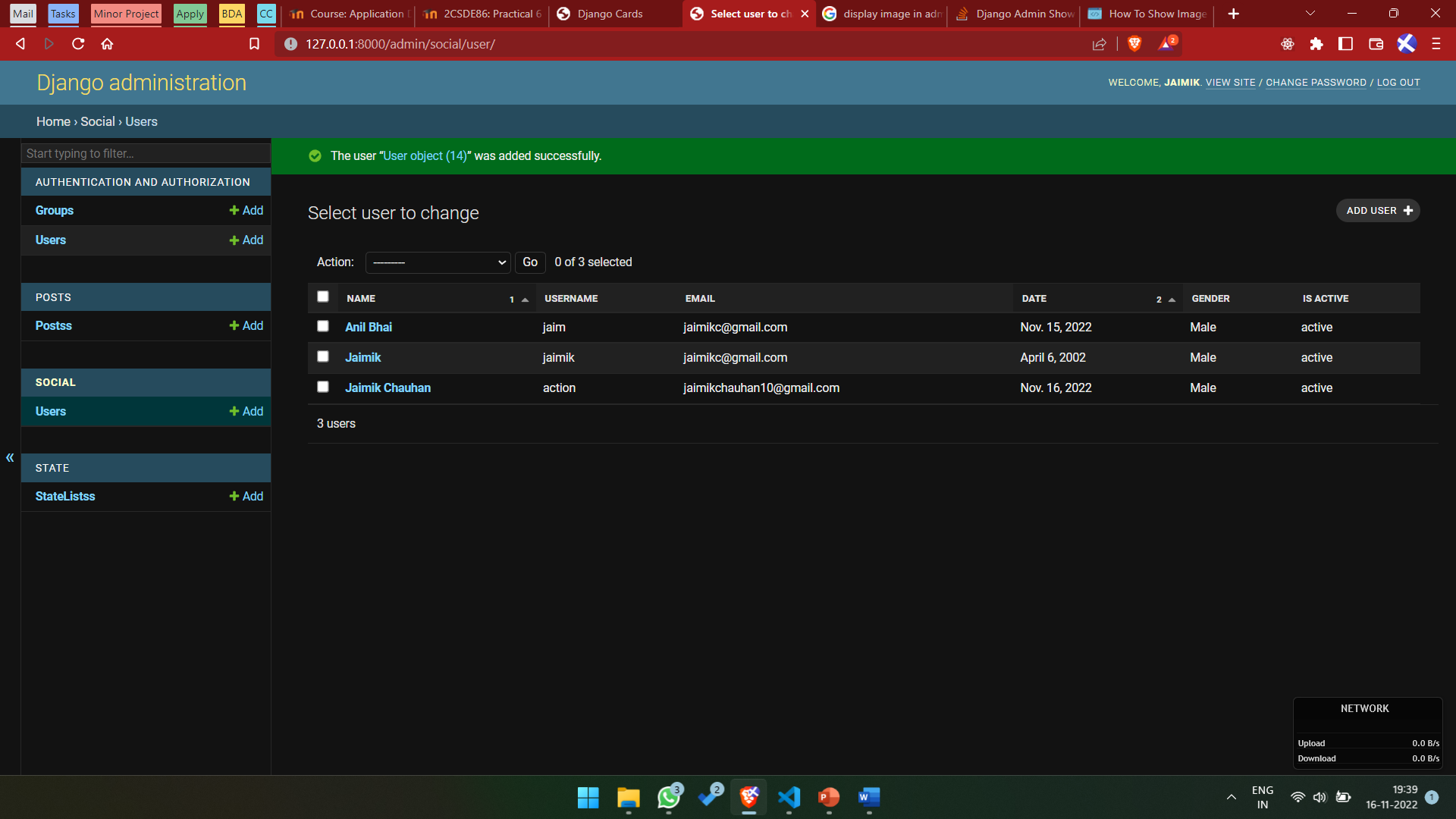
Output Admin Home Page



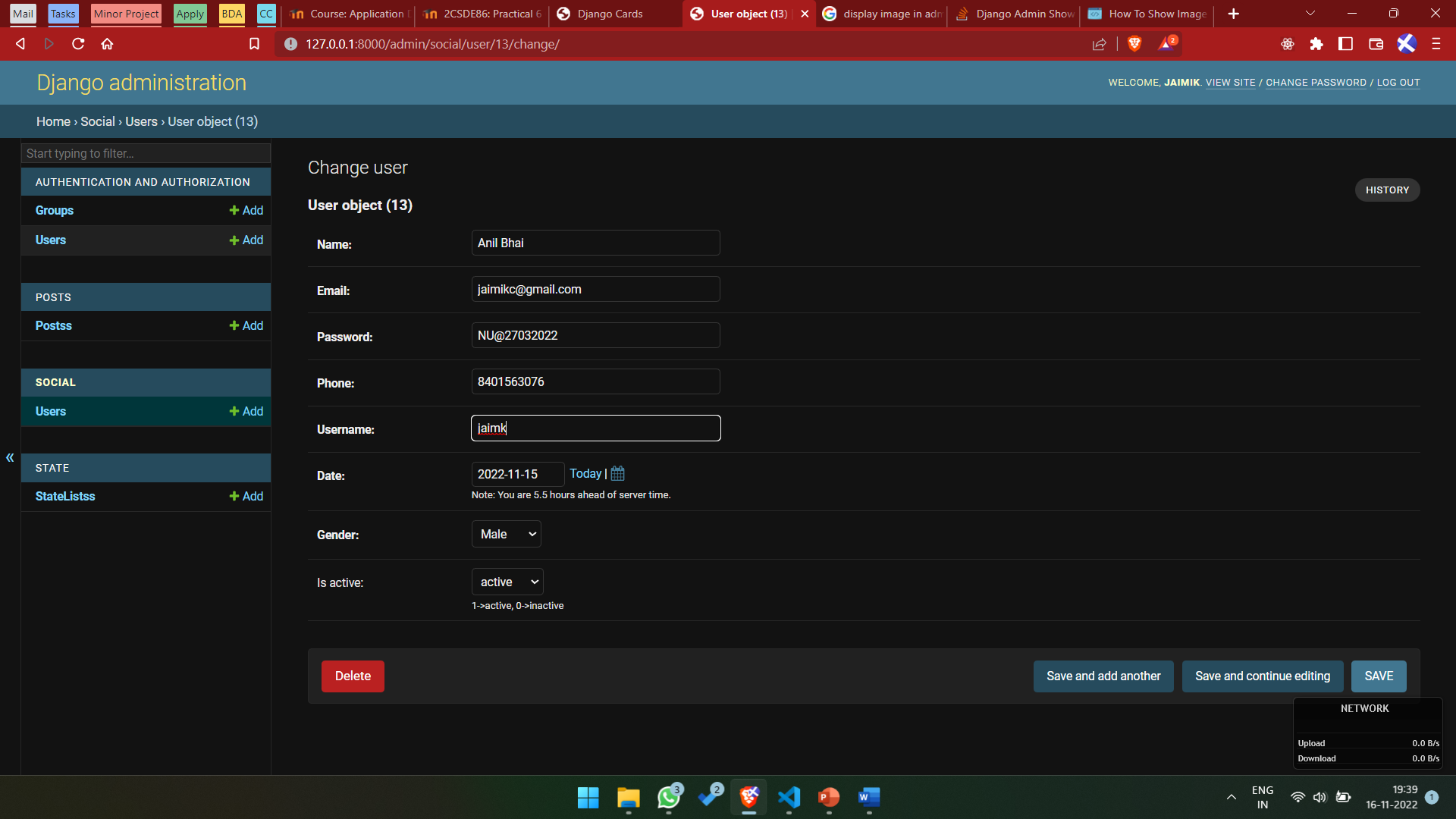
Output 2 List of Users



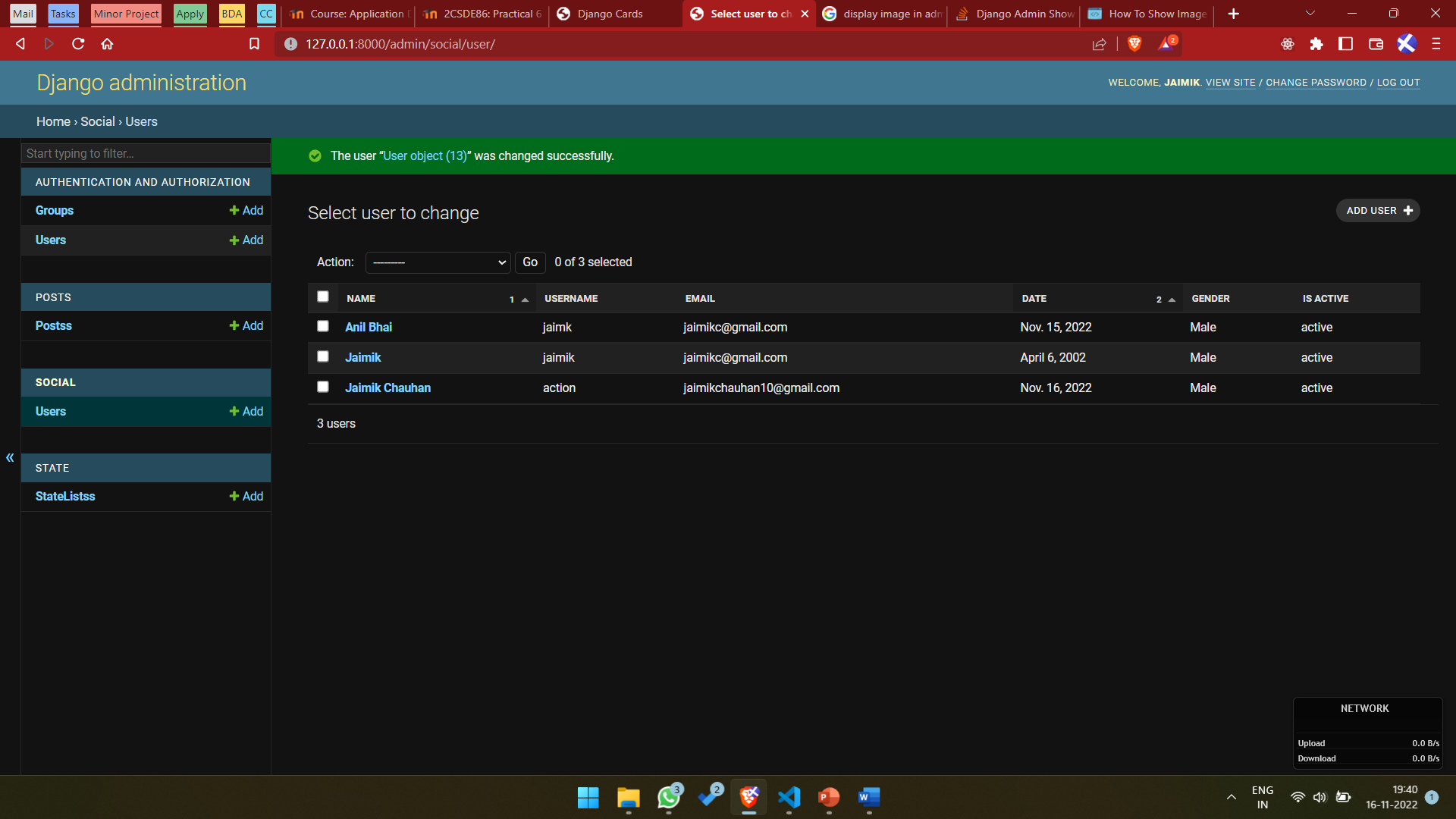
Output Creating a User



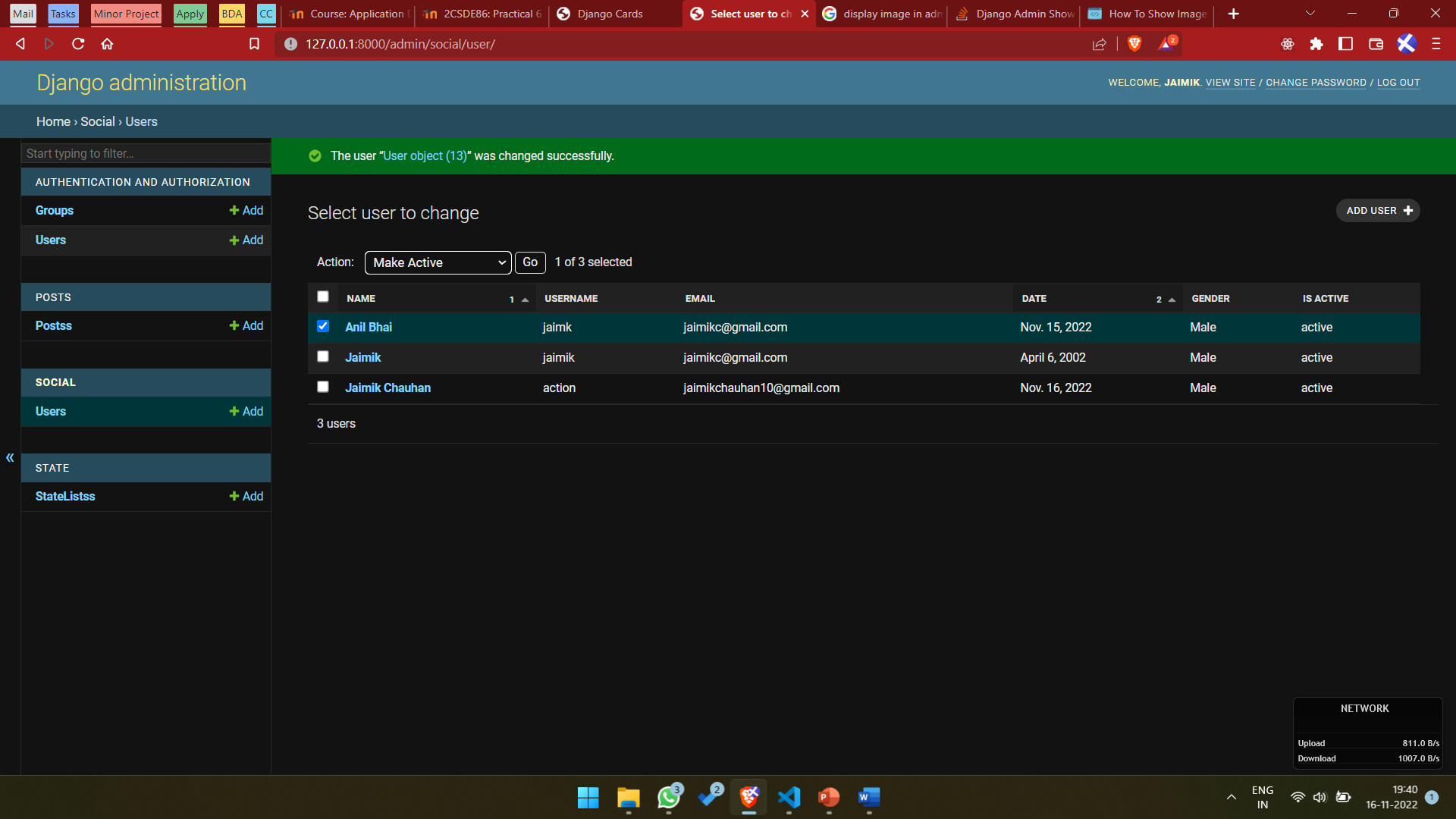
Output After Inserting User



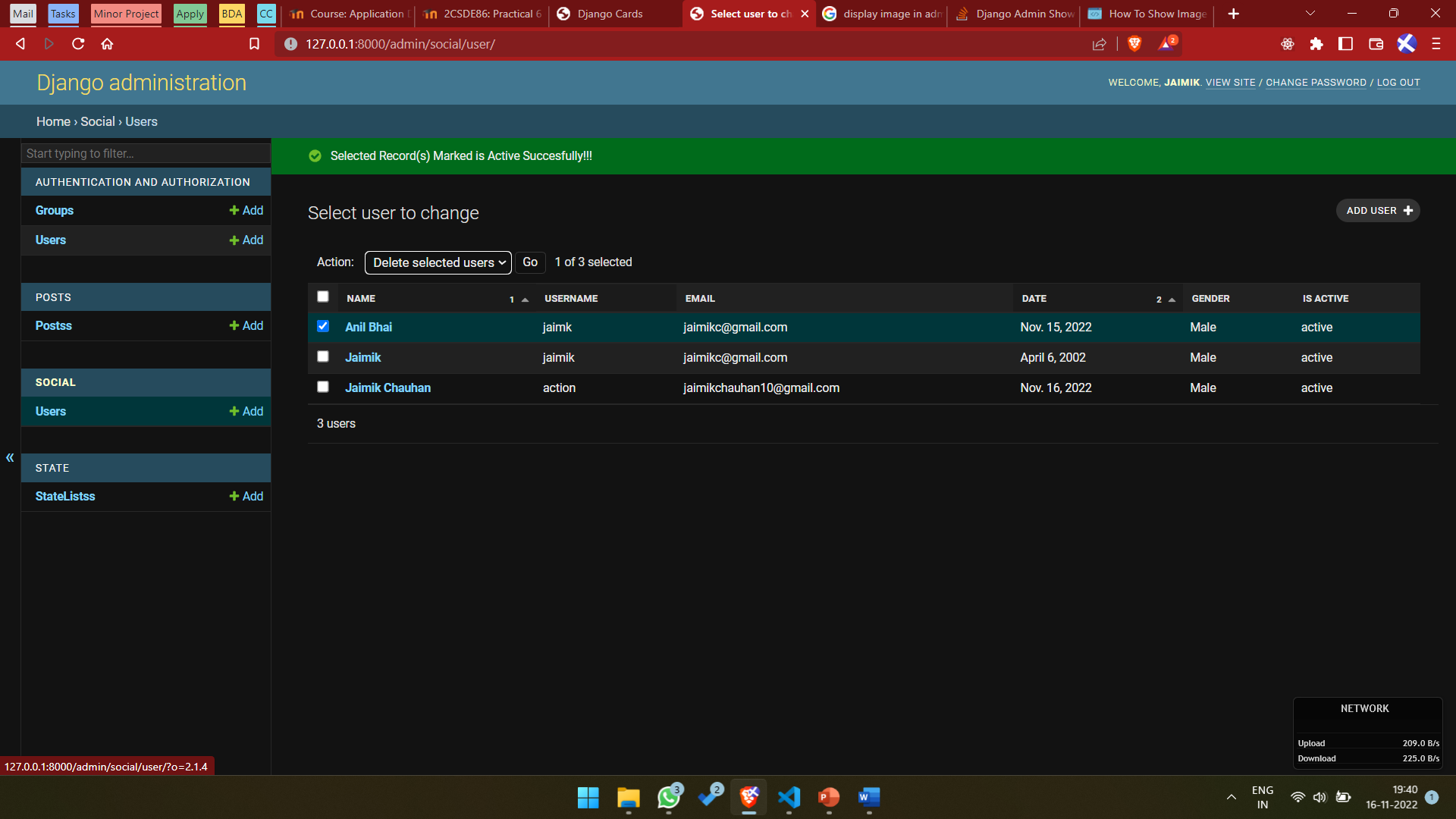
Output Update User Details



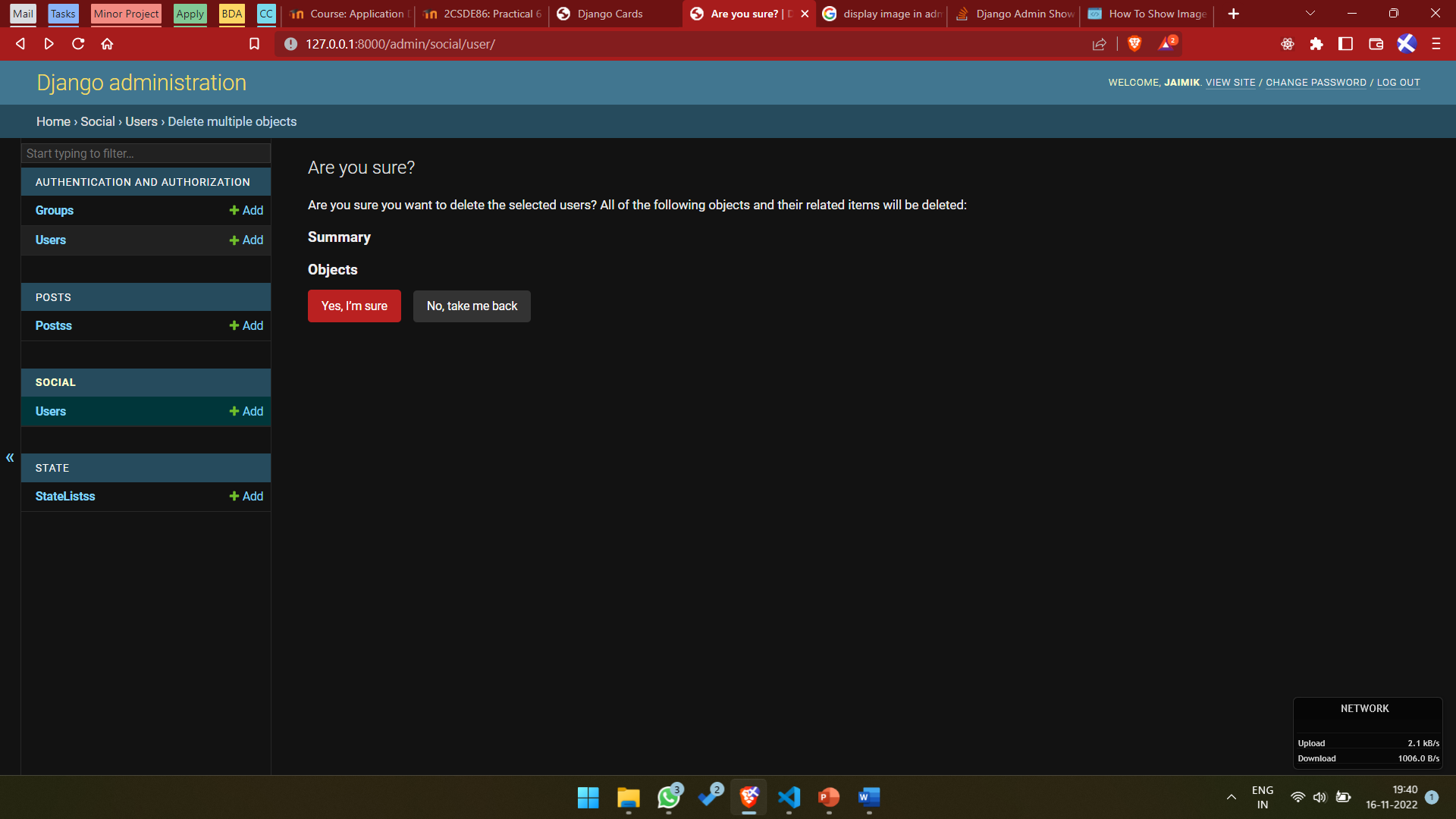
Output User Details Updated



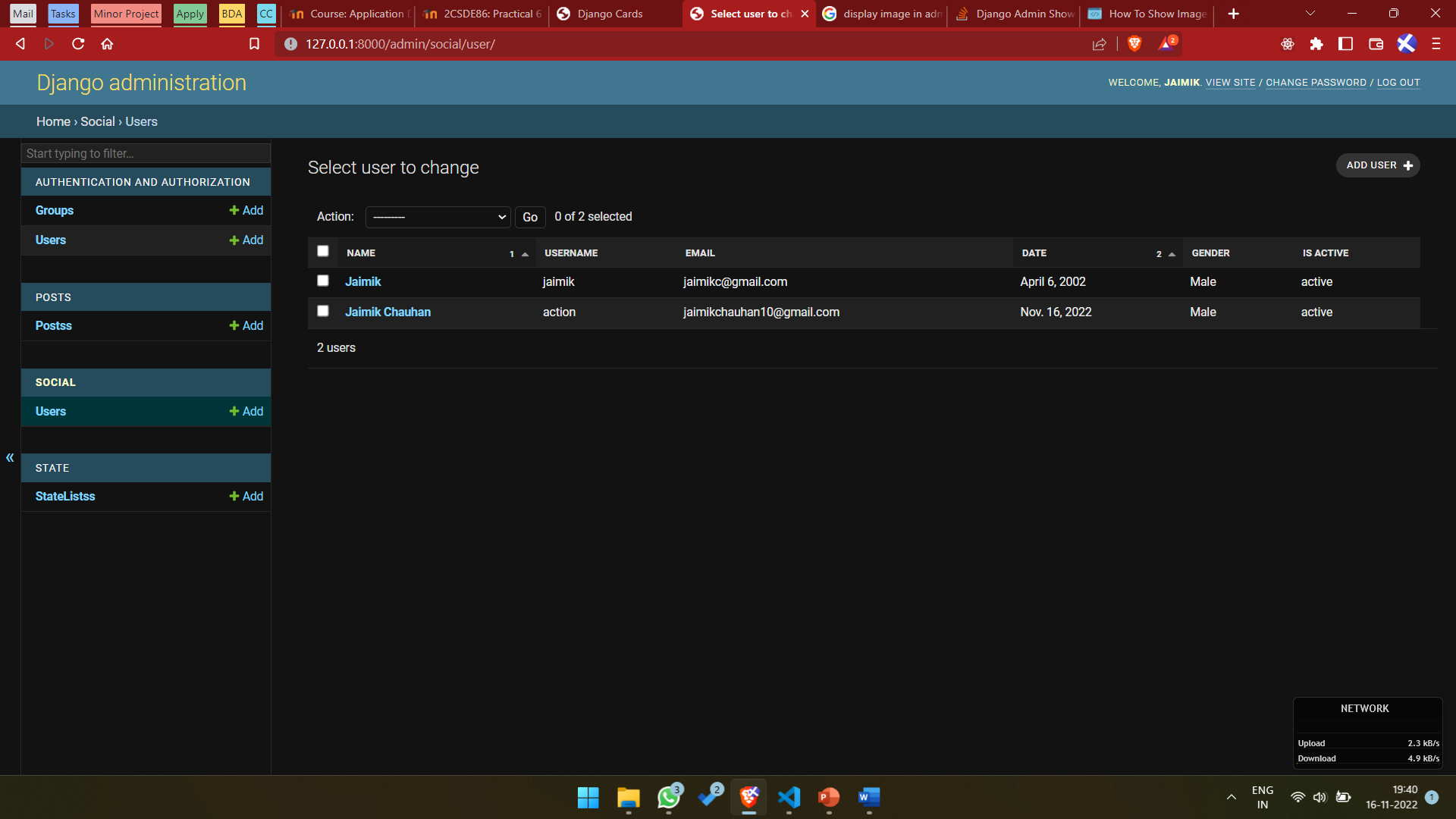
Output Selected Option Make Active User



Output Delete User



Output Confirm Delete User Page



Output Updated List of Users