Python Code

```python
from django.shortcuts import render
from django.http import HttpResponse
from django.contrib.auth.models import User
from django.contrib import messages
from django.shortcuts import render, redirect
from django.contrib.auth.models import User
from django.contrib.auth import authenticate, login as auth_login
from django.contrib.auth.decorators import login_required
from django.contrib import messages
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth import logout
from django.contrib.auth.hashers import make_password
import requests


# Create your views here.


def register(request):
    if request.method == "POST":
        u_fname = request.POST.get("first_name")
        u_lname = request.POST.get("last_name")
        u_name = request.POST.get("username")
        u_email = request.POST.get("email")
        u_pass = request.POST.get("password")
        print(u_fname,u_lname,u_name,u_email,u_pass)
        User.objects.create_user(first_name=u_fname,last_name=u_lname,username=u_name, email=u_email, password=u_pass)
        return redirect('login')

    elif request.method == "GET":
          print("come to get methord")

    return render(request,'register.html')


def login(request):
    u_name = request.POST.get("name")
    u_pass = request.POST.get("password")

    user = authenticate(request, username=u_name, password=u_pass)
    print(user)
    if user is not None:

            auth_login(request, user)
            return redirect('dashboard')
    else:

            messages.error(request, "Invalid username or password.")
            return render(request, 'login.html')

@login_required
```

```python
def dashboard(request):
      return render(request,'dashboard.html')

def user_list(request):
    users = User.objects.all()
    return render(request, 'alluser.html', {'users': users})


def edit_user(request,id):

      if request.method == "POST":
        user  = User.objects.get(id=id)
        print(user)
        u_fname = request.POST.get("first_name")
        u_lname = request.POST.get("last_name")

        u_email = request.POST.get("email")
        print(user, "hg")

        user.first_name = u_fname
        user.last_name = u_lname
        user.email = u_email
        user.save()
        print(user,"sdsda")

        messages.success(request, "User updated successfully!")
        return redirect('userlist')

      elif request.method == "GET":
        user = User.objects.get(id = id)
        return render(request,'edit_user.html',{'user_obj': user})




def delete_user(request, id):
    if request.method == "GET":
        user = User.objects.get(id=id)


        user.delete()
        messages.success(request, "User deleted successfully!")
        print('ghghgjg')
        return redirect('userlist')
# def add_user(request):
#      return HttpResponse("ffjhf")
def add_user(request):
    if request.method == "POST":
        u_fname = request.POST.get("first_name")
        u_lname = request.POST.get("last_name")
        u_name = request.POST.get("username")
        u_email = request.POST.get("email")
        u_pass = request.POST.get("confirm_password")

```

```
107
108             # hashed_password = make_password(u_pass)
109             # print( hashed_password)
110
111             # User.objects.create_user(first_name=u_fname,last_name=u_lname,username=u_
name, email=u_email,password=u_pass)
112             User.objects.create_user(first_name=u_fname,last_name=u_lname,username=u_na
me, email=u_email,password=u_pass)
113
114             return redirect('userlist')
115
116         return render(request,'add.html')
117     @login_required
118     def logout_user(request):
119         logout(request)
120         return redirect('login')
121
122
123
124
125     def postal(request):
126         result = None
127         if request.method == "POST":
128             pincode = request.POST.get('pincode')
129             try:
130                 # CORRECT URL STRUCTURE
131                 url = f"https://api.postalpincode.in/pincode/{pincode}"
132
133                 # Use headers and a timeout for stability
134                 headers = {'User-Agent': 'Mozilla/5.0'}
135                 response = requests.get(url, headers=headers, timeout=5)
136
137                 # Parse the JSON data
138                 result = response.json()
139                 print(result) # Look at your terminal to see the data structure
140             except Exception as e:
141                 print(f"Error fetching data: {e}")
142                 result = None
143
144         return render(request, 'postal.html', {'result': result})
145
146
147
148
149
150     def weather_home(request):
151         weather_data = None
152         city = request.GET.get('city')
153
154         if city:
155             # 1. Geocoding: Convert City Name to Lat/Lon
156             geo_url = f"https://geocoding-api.open-meteo.com/v1/search?name={city}&coun
t=1"
157             geo_res = requests.get(geo_url).json()
158
```

```
            if geo_res.get('results'):
                res = geo_res['results'][0]
                lat, lon = res['latitude'], res['longitude']

                # 2. Get Weather
                w_url = f"https://api.open-meteo.com/v1/forecast?latitude={lat}&longitu
de={lon}&current_weather=true&timezone=Asia/Kolkata"
                data = requests.get(w_url).json()

                weather_data = {
                    'city': res['name'],
                    'state': res.get('admin1', 'India'),
                    'temp': data['current_weather']['temperature'],
                    'wind': data['current_weather']['windspeed'],
                    'time': data['current_weather']['time']
                }

        return render(request, 'weather.html', {'weather': weather_data})
```