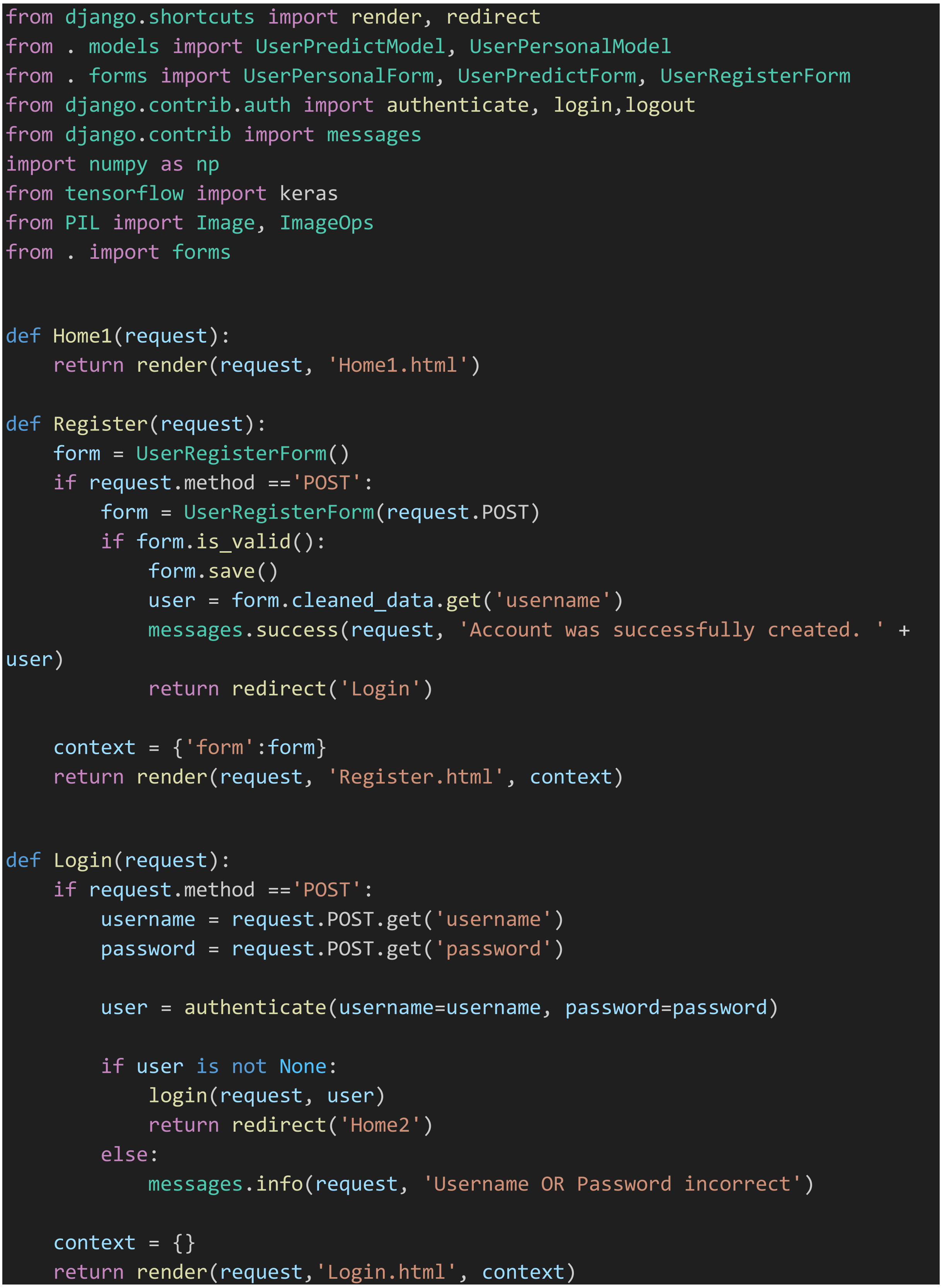
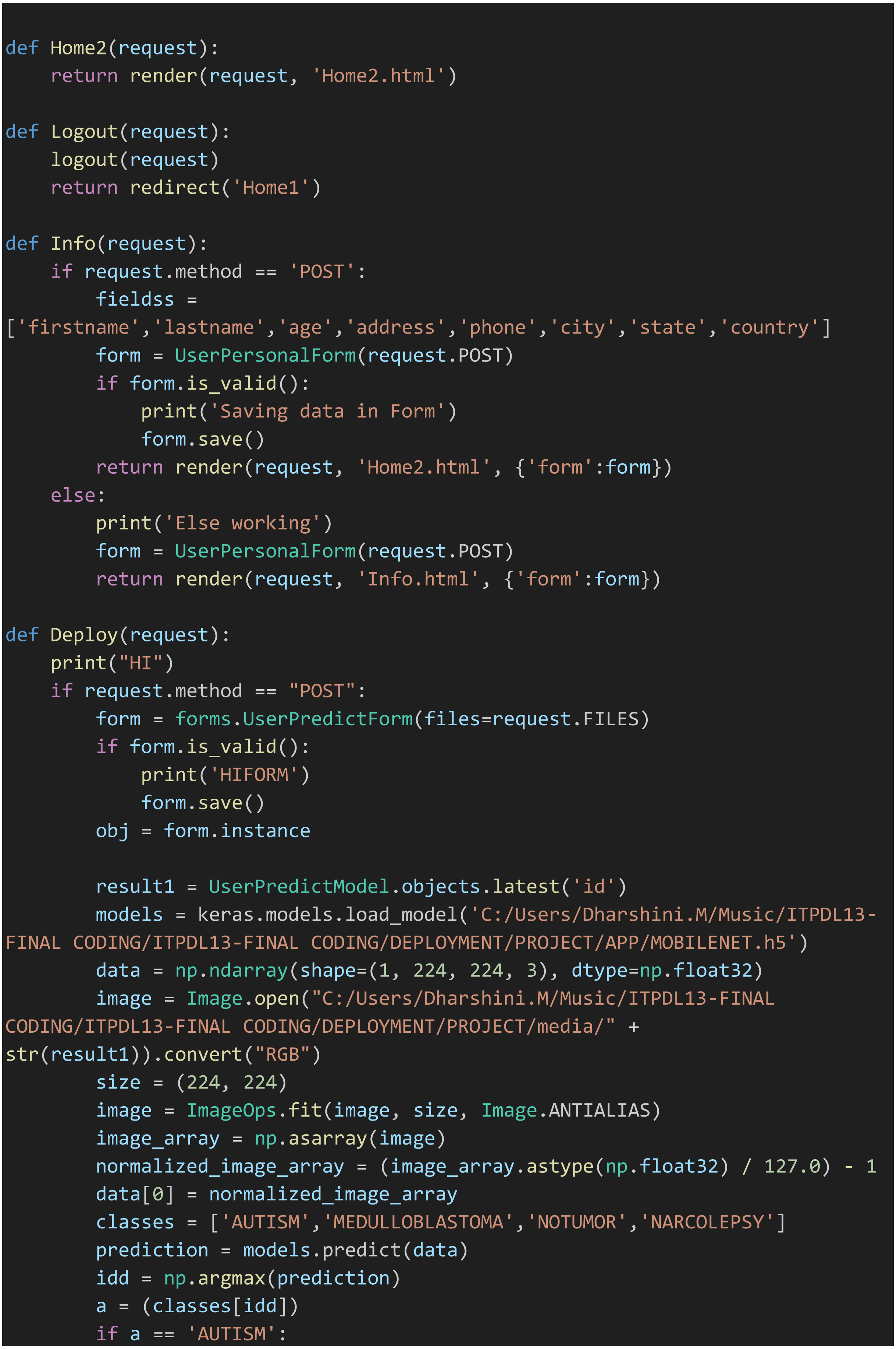
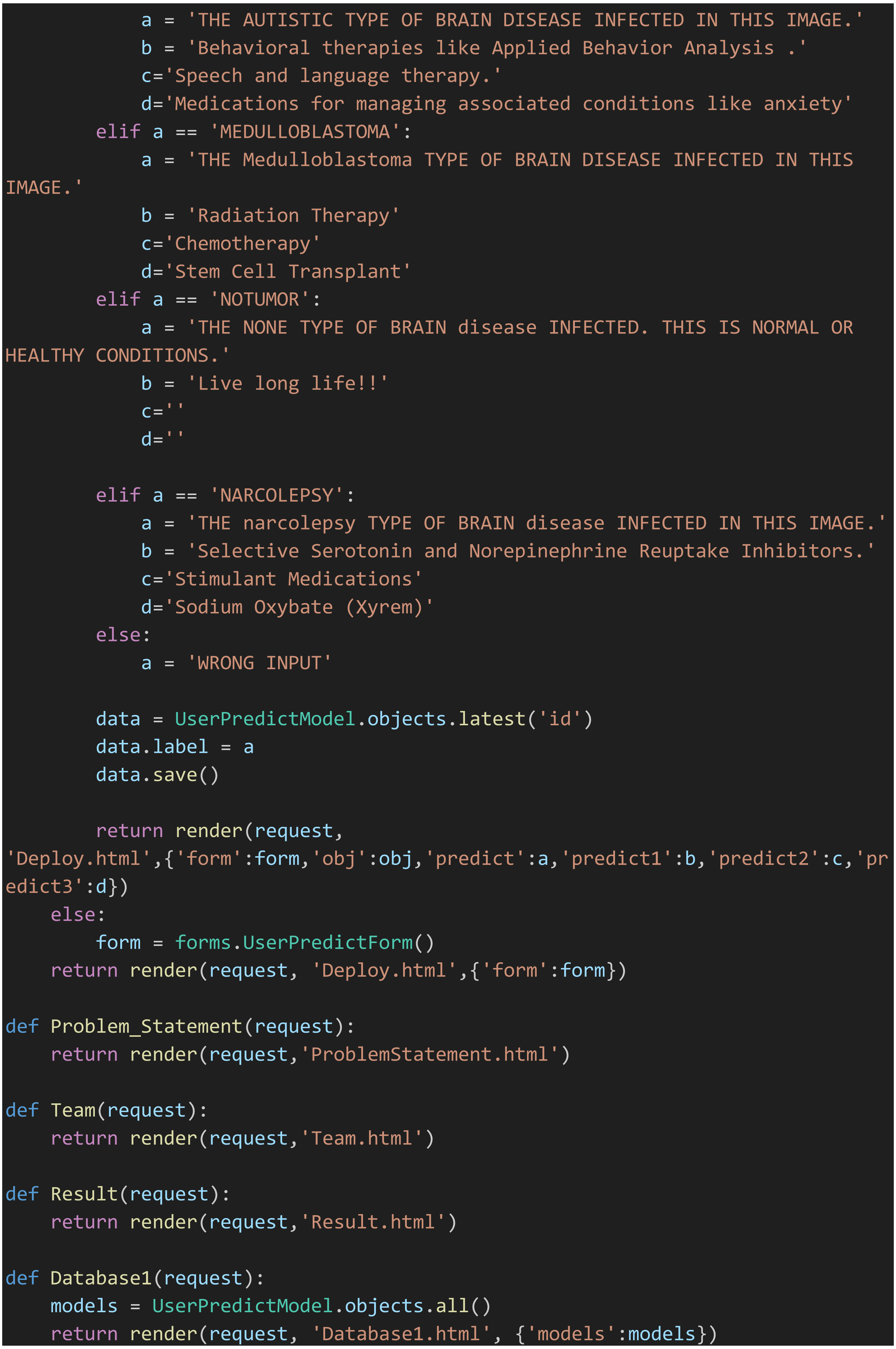
FRAMEWORK : DJANGO



```python
from django.shortcuts import render, redirect
from . models import UserPredictModel, UserPersonalModel
from . forms import UserPersonalForm, UserPredictForm, UserRegisterForm
from django.contrib.auth import authenticate, login,logout
from django.contrib import messages
import numpy as np
from tensorflow import keras
from PIL import Image, ImageOps
from . import forms


def Home1(request):
    return render(request, 'Home1.html')

def Register(request):
    form = UserRegisterForm()
    if request.method =='POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            form.save()
            user = form.cleaned_data.get('username')
            messages.success(request, 'Account was successfully created. ' +
user)
            return redirect('Login')

    context = {'form':form}
    return render(request, 'Register.html', context)


def Login(request):
    if request.method =='POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(username=username, password=password)

        if user is not None:
            login(request, user)
            return redirect('Home2')
        else:
            messages.info(request, 'Username OR Password incorrect')

    context = {}
    return render(request,'Login.html', context)
```



```
def Home2(request):
    return render(request, 'Home2.html')

def Logout(request):
    logout(request)
    return redirect('Home1')

def Info(request):
    if request.method == 'POST':
        fieldss =
['firstname','lastname','age','address','phone','city','state','country']
        form = UserPersonalForm(request.POST)
        if form.is_valid():
            print('Saving data in Form')
            form.save()
        return render(request, 'Home2.html', {'form':form})
    else:
        print('Else working')
        form = UserPersonalForm(request.POST)
        return render(request, 'Info.html', {'form':form})

def Deploy(request):
    print("HI")
    if request.method == "POST":
        form = forms.UserPredictForm(files=request.FILES)
        if form.is_valid():
            print('HIFORM')
            form.save()
        obj = form.instance

        result1 = UserPredictModel.objects.latest('id')
        models = keras.models.load_model('C:/Users/Dharshini.M/Music/ITPDL13-
FINAL CODING/ITPDL13-FINAL CODING/DEPLOYMENT/PROJECT/APP/MOBILENET.h5')
        data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
        image = Image.open("C:/Users/Dharshini.M/Music/ITPDL13-FINAL
CODING/ITPDL13-FINAL CODING/DEPLOYMENT/PROJECT/media/" +
str(result1)).convert("RGB")
        size = (224, 224)
        image = ImageOps.fit(image, size, Image.ANTIALIAS)
        image_array = np.asarray(image)
        normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1
        data[0] = normalized_image_array
        classes = ['AUTISM','MEDULLOBLASTOMA','NOTUMOR','NARCOLEPSY']
        prediction = models.predict(data)
        idd = np.argmax(prediction)
        a = (classes[idd])
        if a == 'AUTISM':
```



```
            a = 'THE AUTISTIC TYPE OF BRAIN DISEASE INFECTED IN THIS IMAGE.'
            b = 'Behavioral therapies like Applied Behavior Analysis .'
            c='Speech and language therapy.'
            d='Medications for managing associated conditions like anxiety'
        elif a == 'MEDULLOBLASTOMA':
            a = 'THE Medulloblastoma TYPE OF BRAIN DISEASE INFECTED IN THIS
IMAGE.'
            b = 'Radiation Therapy'
            c='Chemotherapy'
            d='Stem Cell Transplant'
        elif a == 'NOTUMOR':
            a = 'THE NONE TYPE OF BRAIN disease INFECTED. THIS IS NORMAL OR
HEALTHY CONDITIONS.'
            b = 'Live long life!!'
            c=''
            d=''

        elif a == 'NARCOLEPSY':
            a = 'THE narcolepsy TYPE OF BRAIN disease INFECTED IN THIS IMAGE.'
            b = 'Selective Serotonin and Norepinephrine Reuptake Inhibitors.'
            c='Stimulant Medications'
            d='Sodium Oxybate (Xyrem)'
        else:
            a = 'WRONG INPUT'

        data = UserPredictModel.objects.latest('id')
        data.label = a
        data.save()

        return render(request,
'Deploy.html',{'form':form,'obj':obj,'predict':a,'predict1':b,'predict2':c,'pr
edict3':d})
    else:
        form = forms.UserPredictForm()
    return render(request, 'Deploy.html',{'form':form})

def Problem_Statement(request):
    return render(request,'ProblemStatement.html')

def Team(request):
    return render(request,'Team.html')

def Result(request):
    return render(request,'Result.html')

def Database1(request):
    models = UserPredictModel.objects.all()
    return render(request, 'Database1.html', {'models':models})
```

```python
def Database2(request):
    models = UserPersonalModel.objects.all()
    return render(request, 'Database2.html', {'models':models})
```