**SAMPLE CODE**

**Urls.py**

from django.conf.urls import url

from django.contrib import admin

from django.conf import settings

from django.conf.urls.static import static

from django.urls import path

from adagency.views import \*

from searchadvertising.views import index, adminlogin, adminloginaction, logout, userdetails, activateuser, \

adagencydetails, activateadagency, accuracy, adminhome

from user.views import user, userlogincheck, userregistration, adds, viewdetails, search, usersearchresult, userpage,adds1,viewadagency

urlpatterns = [

url(r'^admin/', admin.site.urls),

url(r'^$', index, name="index"),

url(r'^index/', index, name="index"),

url(r'^adminlogin/', adminlogin, name="adminlogin"),

url(r'^adminhome/', adminhome, name="adminhome"),

url(r'^adminloginaction/', adminloginaction, name="adminloginaction"),

url(r'^userdetails/', userdetails, name="userdetails"),

url(r'^activateuser/', activateuser, name="activateuser"),

url(r'adagencydetails/', adagencydetails, name="adagencydetails"),

url(r'^activateadagency/', activateadagency, name="activateadagency"),

url(r'^accuracy/', accuracy, name="accuracy"),

url(r'^logout/', logout, name="logout"),

url(r'^user/', user, name="user"),

url(r'^userlogincheck/', userlogincheck, name="userlogincheck"),

url(r'^userregistration/', userregistration, name="userregistration"),

url(r'^userpage/', userpage, name="userpage"),

url(r'^adds/', adds, name="adds"),

url(r'^adds1/', adds1, name="adds1"),

url(r'^viewdetails/', viewdetails, name="viewdetails"),

url(r'^search/', search, name="search"),

url(r'^usersearchresult/', usersearchresult, name="usersearchresult"),

#url(r'^useresucces/',useresucces, name="useresucces"),

url(r'^adagency/', adagency, name="adagency"),

url(r'^adagencylogincheck/', adagencylogincheck, name="adagencylogincheck"),

url(r'^adagencyregistration/', adagencyregistration, name="adagencyregistration"),

url(r'^adagencypage/', adagencypage, name="adagencypage"),

url(r'^uploadadds/', uploadadds, name="uploadadds"),

url(r'^viewadds/', viewadds, name="viewadds"),

url(r'^viewadagency/',viewadagency,name='viewadagency')

]

if settings.DEBUG:

urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

main **Views.py**

from random import randint

import mysql.connector

import pandas as pd

from django.shortcuts import render

from django.contrib import messages

from django.http import HttpResponse

from adagency.models import adagencyregistrationmodel

from user.models import registrationmodel

from numpy import loadtxt

from xgboost import XGBClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

def index(request):

return render(request,"index.html")

def adminhome(request):

return render(request,'admin/adminhome.html')

def adminlogin(request):

return render(request, "admin/adminlogin.html")

def adminloginaction(request):

if request.method == "POST":

if request.method == "POST":

login = request.POST.get('loginid')

print(login)

pswd = request.POST.get('password')

if login == 'Admin' and pswd == 'Admin':

return render(request, 'admin/adminhome.html')

else:

messages.success(request, 'Invalid user id and password')

# messages.success(request, 'Invalid user id and password')

return render(request, 'admin/adminlogin.html')

def logout(request):

return render(request,'index.html')

def userdetails(request):

userdata = registrationmodel.objects.all()

return render(request,'admin/viewuserdetails.html', {'object': userdata})

def activateuser(request):

if request.method=='GET':

usid = request.GET.get('usid')

authkey = random\_with\_N\_digits(8)

status = 'activated'

print("USID = ",usid,authkey,status)

registrationmodel.objects.filter(id=usid).update(authkey=authkey , status=status)

userdata = registrationmodel.objects.all()

return render(request,'admin/viewuserdetails.html',{'object':userdata})

def adagencydetails(request):

adagencydata = adagencyregistrationmodel.objects.all()

return render(request,'admin/viewadagencydetails.html', {'object': adagencydata})

def activateadagency(request):

if request.method=='GET':

usid = request.GET.get('usid')

authkey = random\_with\_N\_digits(8)

status = 'activated'

print("USID = ",usid,authkey,status)

adagencyregistrationmodel.objects.filter(id=usid).update(authkey=authkey , status=status)

adagencydata = adagencyregistrationmodel.objects.all()

return render(request,'admin/viewadagencydetails.html',{'object':adagencydata})

def random\_with\_N\_digits(n):

range\_start = 10\*\*(n-1)

range\_end = (10\*\*n)-1

return randint(range\_start, range\_end)

def accuracy(request):

mydb = mysql.connector.connect(

host = "localhost",

user = "root",

password = "root",

database = "advertising"

)

mycursor = mydb.cursor()

mycursor.execute("SELECT price,rating FROM user\_viewdetailsmodel")

myresult = mycursor.fetchall()

dataset = pd.DataFrame(myresult)

#print("dataset",dataset)

mycursor.execute("SELECT rating FROM user\_viewdetailsmodel")

myresult1 = mycursor.fetchall()

dataset1 = pd.DataFrame(myresult1)

#print ("dataset1",dataset1)

#dataset = loadtxt('pima-indians-diabetes.csv', delimiter=",")

print(dataset.shape)

# split data into X and y

#X = dataset[:, 0:8]

#Y = dataset[:, 8]

X = dataset

print("X",len(X))

Y = dataset1

print("Y", len(Y))

#print("x",X)

# split data into train and test sets

seed = 2

test\_size = 5

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=test\_size, random\_state=seed)

"""print("X\_train",X\_train)

print(" X\_test", X\_test)

print("y\_train",y\_train)

print("y\_test",y\_test)"""

# fit model no training data

model = XGBClassifier()

print('Katti X= ',X\_train)

print('Katti y= ', y\_train)

model.fit(X\_train, y\_train)

# make predictions for test data

y\_pred = model.predict(X\_test)

predictions = [round(value) for value in y\_pred]

print("predictions",predictions)

# evaluate predictions

accuracy = accuracy\_score(y\_test, predictions)

print("Accuracy: %.2f%%" % (accuracy \* 100.0))

dict = {

"accuracy" : accuracy\* 100.0,

}

return render(request,"admin/accuracy.html",dict)

#return HttpResponse("wait for algoritham process")

Add agency

**forms.py**

from django import forms

from adagency.models import adagencyregistrationmodel, uploadmodel

from django.core import validators

# def name\_check(value):

# if value.isalpha!=True:

# raise forms.ValidationError("only strings allowed")

class adagencyregistrationform(forms.ModelForm):

#loginid = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100,validators=[name\_check])

loginid = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100)

password = forms.CharField(widget=forms.PasswordInput(), required=True, max\_length=100)

email = forms.EmailField(widget=forms.TextInput(),required=True)

mobile = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100,validators=[validators.MaxLengthValidator(10),validators.MinLengthValidator(10)])

place = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100)

city = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100)

authkey = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

status = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

class Meta:

model = adagencyregistrationmodel

fields = ['loginid','password','email','mobile','place','city','authkey','status' ]

class UploadaddForm(forms.ModelForm):

category = forms.ChoiceField(choices=[('Marketing','Marketing'),('Education','Education'),('Realestates','Realestates'),('Mobiles','Mobiles'),('Vehicles','Vehicles'

'')])

class Meta:

model = uploadmodel

fields = ('category','property','brand','city','price','file')

add agency

**models.py**

from django.db import models

# Create your models here.

class adagencyregistrationmodel(models.Model):

loginid = models.CharField(max\_length=100)

password = models.CharField(max\_length=100)

email = models.EmailField()

mobile = models.CharField(max\_length=100)

place = models.CharField(max\_length=100)

city = models.CharField(max\_length=100)

authkey = models.CharField(max\_length=100)

status = models.CharField(max\_length=100)

def \_\_str\_\_(self):

return self.email

class uploadmodel(models.Model):

category = models.CharField(max\_length=100)

property = models.CharField(max\_length=100)

brand = models.CharField(max\_length=100)

city = models.CharField(max\_length=100)

price = models.CharField(max\_length=100)

file = models.FileField(upload\_to='files/pdfs/')

def \_\_str\_\_(self):

return self.category

add agency

**views.py**

from django.contrib import messages

from django.http import HttpResponseRedirect

from django.shortcuts import render, redirect

# Create your views here.

from adagency.forms import adagencyregistrationform, UploadaddForm

from adagency.models import adagencyregistrationmodel, uploadmodel

def adagency(request):

return render(request,"adagency/adagency.html")

def adagencypage(request):

return render(request,'adagency/adagencypage.html')

def adagencylogincheck(request):

if request.method == 'POST':

usid = request.POST.get('loginid')

print(usid)

pswd = request.POST.get('password')

print(pswd)

try:

check = adagencyregistrationmodel.objects.get(loginid=usid, password=pswd)

# print('usid',usid,'pswd',pswd)

request.session['userid'] = check.loginid

status = check.status

if status == "activated":

request.session['email'] = check.email

#auth.login(request, usid)

return render(request,'adagency/adagencypage.html')

else:

messages.success(request, 'user is not activated')

return render(request,'adagency/adagency.html')

except Exception as e:

print('Exception is ', str(e))

messages.success(request,'Invalid user id and password')

return render(request,'adagency/adagency.html')

def adagencyregistration(request):

if request.method == 'POST':

form = adagencyregistrationform(request.POST)

if form.is\_valid():

# print("Hai Meghana")

form.save()

messages.success(request, 'you are successfully registred')

return HttpResponseRedirect('adagency')

else:

messages.success(request, 'Email Id already exist')

print('Invalid form')

else:

form = adagencyregistrationform()

return render(request, "adagency/adagencyregistration.html", {'form': form})

def uploadadds(request):

if request.method == 'POST':

form = UploadaddForm(request.POST, request.FILES)

if form.is\_valid():

form.save()

return redirect('adagency/uploadadd\_list.html')

else:

form = UploadaddForm()

return render(request, 'adagency/uploadadd.html', {'form': form})

def viewadds(request):

viewadds = uploadmodel.objects.all()

return render(request,'adagency/viewadds.html',{'adds':viewadds})

user side **forms.py**

from django import forms

from user.models import registrationmodel, viewdetailsmodel

from django.core import validators

def name\_check(value):

if value.isalpha()!=True:

raise forms.ValidationError("only strings allowed for name")

class registrationform(forms.ModelForm):

loginid = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100,validators=[name\_check])

password = forms.CharField(widget=forms.PasswordInput(), required=True, max\_length=100)

email = forms.EmailField(widget=forms.TextInput(),required=True)

mobile = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100,validators=[validators.MaxLengthValidator(10),validators.MinLengthValidator(10)])

gender = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100)

age = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100)

occupation = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100)

place = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100)

city = forms.CharField(widget=forms.TextInput(),required=True,max\_length=100)

authkey = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

status = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

class Meta:

model = registrationmodel

fields = ['loginid','password','email','mobile','gender','age','occupation','place','city','authkey','status' ]

class viewdetailsform(forms.ModelForm):

property = forms.CharField(widget=forms.TextInput(attrs={'readonly': 'readonly'}), required=True, max\_length=100)

brand = forms.CharField(widget=forms.TextInput(attrs={'readonly': 'readonly'}), required=True, max\_length=100)

city = forms.CharField(widget=forms.TextInput(attrs={'readonly': 'readonly'}), required=True, max\_length=100)

price = forms.IntegerField(widget=forms.TextInput(attrs={'readonly': 'readonly'}), required=True)

#file = forms.FileField(widget=forms.TextInput(attrs={'readonly': 'readonly'}), required=True, max\_length=100)

rating = forms.ChoiceField(choices=[('1','1'),('2','2'),('3','3'),('4','4'),('5','5')])

review = forms.CharField(widget=forms.TextInput(), required=True, max\_length=100)

class Meta:

model = viewdetailsmodel

fields = ('property', 'brand', 'city', 'price', 'rating','review')

User side **models.py**

from django.db import models

# Create your models here.

class registrationmodel(models.Model):

loginid = models.CharField(max\_length=100)

password = models.CharField(max\_length=100)

email = models.EmailField()

mobile = models.CharField(max\_length=100)

gender = models.CharField(max\_length=100)

age = models.CharField(max\_length=100)

occupation = models.CharField(max\_length=100)

place = models.CharField(max\_length=100)

city = models.CharField(max\_length=100)

authkey = models.CharField(max\_length=100)

status = models.CharField(max\_length=100)

def \_\_str\_\_(self):

return self.email

class viewdetailsmodel(models.Model):

#category = models.CharField(max\_length=100)

property = models.CharField(max\_length=100)

brand = models.CharField(max\_length=100)

city = models.CharField(max\_length=100)

price = models.IntegerField()

#file = models.FileField(upload\_to='files/pdfs/')

rating=models.IntegerField()

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

User side **Views.py**

import uuid

from django.contrib import messages

from django.http import HttpResponseRedirect,HttpResponse

from django.shortcuts import render, redirect

# Create your views here.

from adagency.models import uploadmodel

from user.forms import registrationform, viewdetailsform

from user.models import registrationmodel, viewdetailsmodel

def user(request):

return render(request, "user/user.html")

def userpage(request):

return render(request,'user/userpage.html')

def userlogincheck(request):

if request.method == 'POST':

usid = request.POST.get('loginid')

# print(usid)

pswd = request.POST.get('password')

# print(pswd)

try:

check = registrationmodel.objects.get(loginid=usid, password=pswd)

# print('usid',usid,'pswd',pswd)

print(check)

request.session['userid'] = check.loginid

print(check.loginid)

request.session['id'] = check.id

print(check.id)

status = check.status

if status == "activated":

request.session['email'] = check.email

#auth.login(request, usid)

return render(request,'user/userpage.html')

else:

messages.success(request, 'user is not activated')

return render(request,'user/user.html')

except Exception as e:

print('Exception is ', str(e))

messages.success(request,'Invalid user id and password')

return render(request,'user/user.html')

def userregistration(request):

if request.method == 'POST':

form = registrationform(request.POST)

if form.is\_valid():

# print("Hai Meghana")

form.save()

#return HttpResponse("succesfully registered")

return render(request,"user/user.html")

else:

print('Invalid form')

else:

form = registrationform()

return render(request, "user/userregistration.html", {'form': form})

def adds(request):

#print(uuid.uuid1())

viewadds = uploadmodel.objects.all()

#print(uuid.viewadds())

return render(request,'user/userviewadds.html',{'adds':viewadds})

def adds1(request):

idno=request.session['id']

print(idno)

adds=uploadmodel.objects.filter(id=idno)

print(adds)

return render(request,"user/userviewadds1.html",{"adds":adds})

"""def viewdetails(request):

id = request.GET.get('id')

check = uploadmodel.objects.get(id=id)

request.session['id'] = check.id

id = request.session["id"]

viewadds = uploadmodel.objects.get(id=id)

return render(request, 'user/viewdetails.html', {'adds': viewadds})"""

def viewdetails(request):

if request.method == "POST":

forms = viewdetailsform(request.POST)

if forms.is\_valid():

forms.save()

#messages.success(request, 'successfully ')

return render(request,'user/useresucces.html')

else:

print('Invalid Form')

price = request.POST.get('price')

brand = request.POST.get('brand')

property = request.POST.get('property')

city = request.POST.get('city')

#file = request.POST.get('file')

rating = request.POST.get('rating')

review = request.POST.get('review')

#data = {'brand': brand, 'price': price, 'property': property, 'city': city, 'file': file, 'rating': rating, 'review': review}

#print("data",data)

form = viewdetailsmodel(price=price,brand=brand,property=property,city=city,rating=rating,review=review)

form.save()

print("form savved")

return render(request,'user/userviewadds.html',{'adds':form})

else:

if request.method =='GET':

print("get method taking")

id = request.GET.get('id')

print('Image Id = ',id)

price = request.GET.get('price')

brand = request.GET.get('brand')

property = request.GET.get('property')

city = request.GET.get('city')

file = request.GET.get('file')

#name= request.session['userid']

# print("name",name)

print('Price = ',price,' brand',brand)

data = uploadmodel.objects.get(id=id)

#data = {'brand': brand, 'price': price, 'property':property,'city':city, 'file':file}

print("data image url =",type(data.file))

data2 = {'brand': data.brand, 'price': data.price, 'property':data.property,'city':data.city}

viewadds = viewdetailsform(data2)

#print('Image Localtion Katti ',file)

#file ='media/'+file

return render(request, 'user/viewdetails.html', {'adds': viewadds,'image':data.file})

def search(request):

return render(request, 'user/usersearch.html')

def viewadagency(request):

viewadds = uploadmodel.objects.all()

return render(request,'admin/viewadagency.html',{'adds':viewadds})

def usersearchresult(request):

property = request.GET.get('property')

print('product is', property, ' and its type ', type(property))

dict = {}

check = uploadmodel.objects.filter(property=property)

print(check)

object = check.filter(property=property)

return render(request, 'user/usersearchresult.html', {"object": object})

**Addagencybase.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<title>Searchadvertising</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link href="https://fonts.googleapis.com/css?family=Muli:300,400,700,900" rel="stylesheet">

<link rel="stylesheet" href="{% static 'fonts/icomoon/style.css' %}">

<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">

<link rel="stylesheet" href="{% static 'css/jquery-ui.css' %}">

<link rel="stylesheet" href="{% static 'css/owl.carousel.min.css' %}">

<link rel="stylesheet" href="{% static 'css/owl.theme.default.min.css' %}">

<link rel="stylesheet" href="{% static 'css/owl.theme.default.min.css' %}">

<link rel="stylesheet" href="{% static 'css/jquery.fancybox.min.css' %}">

<link rel="stylesheet" href="{% static 'css/bootstrap-datepicker.css' %}">

<link rel="stylesheet" href="{% static 'fonts/flaticon/font/flaticon.css' %}">

<link rel="stylesheet" href="{% static 'vcss/aos.css' %}">

<link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body data-spy="scroll" data-target=".site-navbar-target" data-offset="300">

<div class="site-wrap">

<div class="site-mobile-menu site-navbar-target">

<div class="site-mobile-menu-header">

<div class="site-mobile-menu-close mt-3">

<span class="icon-close2 js-menu-toggle"></span>

</div>

</div>

<div class="site-mobile-menu-body"></div>

</div>

<header class="site-navbar py-4 js-sticky-header site-navbar-target" role="banner">

<div class="container-fluid">

<div class="d-flex align-items-center">

<div class="site-logo"><a href="index.html">SearchAdvertising</a></div>

<div>

<nav class="site-navigation position-relative text-right" role="navigation">

<ul class="site-menu main-menu js-clone-nav mr-auto d-none d-lg-block">

<li><a href="{% url 'adagencypage' %}" class="nav-link">HOME</a></li>

<li><a href="{% url 'uploadadds' %}" class="nav-link">UPLOADADDS</a></li>

<li><a href="{% url 'viewadds' %}" class="nav-link">VIEWADDS</a></li>

<li><a href="{% url 'logout' %}" class="nav-link">LOGOUT</a></li>

</ul>

</nav>

</div>

<div class="ml-auto">

<nav class="site-navigation position-relative text-right" role="navigation">

<ul class="site-menu main-menu site-menu-dark js-clone-nav mr-auto d-none d-lg-block">

<li class="cta"><a href="{% url 'logout' %}" class="nav-link"><span class="rounded border border-primary">{{request.session.userid}}</span></a></li>

</ul>

</nav>

<a href="#" class="d-inline-block d-lg-none site-menu-toggle js-menu-toggle text-black float-right"><span class="icon-menu h3"></span></a>

</div>

</div>

</div>

</header>

<!--<div class="intro-section" id="home-section">

<div class="container">

<div class="row align-items-center">

<div class="col-lg-4 mr-auto" data-aos="fade-up">

<h1>Feature Engineering for Search Advertising Recognition.</h1>

<h1><p class="mb-5">Feature Engineering for Search Advertising Recognition.</p></h1>

<p><a href="#" class="btn btn-outline-light py-3 px-5">Get Started</a></p>

</div>-->

<!--<div class="col-lg-2 ml-auto" data-aos="fade-up" data-aos-delay="100">

<figure class="img-absolute">

<img src="{% static 'images/person\_0.jpg' %}" alt="Image" class="img-fluid" width="500">

<img src="{% static 'images/person.jpg' %}" alt="Image" class="img-fluid" width="500">

</figure>

</div>

</div>

</div>

</div>-->{% block contents %}

{% endblock %}

<footer class="footer-section">

<div class="row pt-5 mt-5 text-center">

<div class="col-md-12">

<div class="border-top pt-5">

<p>

Copyright &copy;<script>document.write(new Date().getFullYear());</script>

</p>

</div>

</div>

</div>

</footer>

</div> <!-- .site-wrap -->

<script src="{% static 'js/jquery-3.3.1.min.js' %}"></script>

<script src="{% static 'js/jquery-migrate-3.0.1.min.js' %}"></script>

<script src="{% static 'js/jquery-ui.js' %}"></script>

<script src="{% static 'js/popper.min.js' %}"></script>

<script src="{% static 'js/bootstrap.min.js' %}"></script>

<script src="{% static 'js/owl.carousel.min.js' %}"></script>

<script src="{% static 'js/jquery.stellar.min.js' %}"></script>

<script src="{% static 'js/jquery.countdown.min.js' %}"></script>

<script src="{% static 'js/bootstrap-datepicker.min.js' %}"></script>

<script src="{% static 'js/jquery.easing.1.3.js' %}"></script>

<script src="{% static 'js/aos.js' %}"></script>

<script src="{% static 'js/jquery.fancybox.min.js' %}"></script>

<script src="{% static 'js/jquery.sticky.js' %}"></script>

<script src="{% static 'js/main.js' %}"></script>

</body>

</html>

**Usersearch.html**

{% extends 'userbase.html'%}

{% load static %}

{% block contents %}

<div class="intro-section" id="home-section">

<div class="container">

<div class="row align-items-center">

<div class="col-lg-4 mr-auto" data-aos="fade-up">

<h3><form method="GET" action="{% url 'usersearchresult' %}">

{% csrf\_token %}

<table style="width:30%">

<tr>

<td><input type="text" name="property"> </td>

<td><input type="submit" value="Search"></td><br>

<P>we can search based upon property</P>

</tr>

</table></h3>

<!--<h1><p class="mb-5">Feature Engineering for Search Advertising Recognition.Meghana</p></h1>

<p><a href="#" class="btn btn-outline-light py-3 px-5">Get Started</a></p>-->

</div>

<div class="col-lg-2 ml-auto" data-aos="fade-up" data-aos-delay="100">

<figure class="img-absolute">

<img src="{% static 'images/search.jpg' %}" alt="Image" class="img-fluid" width="500">

</figure>

</div>

</div>

</div>

</div>

{% endblock %}

**Userregister.html**

{% extends 'base.html'%}

{% load static %}

{% block contents %}

<div class="intro-section" id="home-section">

<div class="container">

<div class="row align-items-center">

<div class="col-lg-4 mr-auto" data-aos="fade-up">

<h4 ><form action="{% url 'userregistration' %}" method="POST" class="form-box">

{% csrf\_token %}

<table>

<tr><td style="color:white">Login ID</td><td >{{form.loginid}}</td></tr>

<tr><td style="color:white">Password</td><td>{{form.password}}</td></tr>

<tr><td style="color:white">Email</td><td>{{form.email}}</td></tr>

<tr><td style="color:white">Mobile</td><td>{{form.mobile}}</td></tr>

<tr><td style="color:white">Gender</td><td>{{form.gender}}</td></tr>

<tr><td style="color:white">Age</td><td>{{form.age}}</td></tr>

<tr><td style="color:white">Occupation</td><td>{{form.occupation}}</td></tr>

<tr><td style="color:white">Place</td><td>{{form.place}}</td></tr>

<tr><td style="color:white">City</td><td>{{form.city}}</td></tr>

<tr><td></td><td>{{form.authkey}}</td></tr>

<tr><td></td><td>{{form.status}}</td></tr>

</table>

<div class="form-group">

<input type="submit" value="Sign up" style="color:Blue">

</div>

</form></h4>

<!--<h1><p class="mb-5">Feature Engineering for Search Advertising Recognition.Meghana</p></h1>

<p><a href="#" class="btn btn-outline-light py-3 px-5">Get Started</a></p>-->

</div>

<div class="col-lg-2 ml-auto" data-aos="fade-up" data-aos-delay="100">

<figure class="img-absolute">

<img src="{% static 'images/registration.jpg' %}" alt="Image" class="img-fluid" width="500" >

</figure>

</div>

</div>

</div>

</div>

{% endblock %}

**Userviewadds.html**

{% extends 'userbase.html'%}

{% load static %}

{% block contents %}

<div class="intro-section" id="home-section">

<div class="container">

<div class="row align-items-center">

<div class="col-lg-4 mr-auto" data-aos="fade-up">

<h1>USER VIEW THE ADDS</h1>

</div>

<div class="col-lg-2 ml-auto" data-aos="fade-up" data-aos-delay="100">

<figure class="img-absolute">

<table border="2px solid blue">

<tr> <th>add</th>

{% for x in adds%}

<tr>

<td><marquee behavior="scroll" direction="up"><a href="/viewdetails/?id={{x.id}}&price={{x.price}}&brand={{x.brand}}&property={{x.property}}&city={{x.city}}&file={{x.file}}"> <img src="{{x.file.url}}" style="width:25em;height:10em;"></a></marquee></td>

</tr>

{% endfor %}

</table>

</figure>

</div>

</div>

</div>

</div>

</div>

{% endblock %}

**Viewadd.html**

{% extends 'userbase.html'%}

{% load static %}

{% block contents %}

<div class="intro-section" id="home-section">

<div class="container">

<div class="row align-items-center">

<div class="col-lg-4 mr-auto" data-aos="fade-up">

<div class="col-sm-4" style="background-color:;border:;">

<form action="{% url 'viewdetails' %}" method="POST" enctype="multipart/form-data" >

{% csrf\_token %}

<table align="center" style="margin-top:10px;">

<tr>

<td colspan="3" align="center">

<img src="{{image.url}}" style="width:25em;height:10em;">

</td>

<!--<tr><td style="color:black">Category</td>

<td style="color:black">:</td>

<td height="30px" style="color:black">{{adds.file}}</td></tr>-->

<tr><td style="color:black">Property</td>

<td style="color:black">:</td>

<td height="30px" style="color:black">{{adds.property}}</td></tr>

<tr><td style="color:black">Brand</td>

<td style="color:black">:</td>

<td height="30px" style="color:black">{{adds.brand}}</td></tr>

<tr><td style="color:black">City</td>

<td style="color:black">:</td>

<td height="30px" style="color:black">{{adds.city}}</td></tr>

<tr><td style="color:black">Price</td>

<td style="color:black">:</td>

<td height="30px" style="color:black">{{adds.price}}</td></tr>

<tr><td style="color:black">Rating</td>

<td style="color:black">:</td>

<td height="30px" style="color:black">{{adds.rating}}</td></tr>

<tr><td style="color:black">Review</td>

<td style="color:black">:</td>

<td height="30px" style="color:black">{{adds.review}}</td></tr>

</tr>

<tr>

<td align="center"><input style="color:Black" type="submit" name="submit" ></td>

</tr>

</table></div>

</div>

<!--<h1><p class="mb-5">Feature Engineering for Search Advertising Recognition.Meghana</p></h1>

<p><a href="#" class="btn btn-outline-light py-3 px-5">Get Started</a></p>-->

</div>

<div class="col-lg-2 ml-auto" data-aos="fade-up" data-aos-delay="100">

<figure class="img-absolute">

</figure>

</div>

</div>

</div>

</div>

{% endblock %}

**Agentviewadds.html**

{% extends 'adagencybase.html'%}

{% load static %}

{% block contents %}

<div class="intro-section" id="home-section">

<div class="container">

<div class="row align-items-center">

<div class="col-lg-4 mr-auto" data-aos="fade-up">

<br><br><br><br><h3 style="color:white">View Upload adds:</h3>

<div class="col-sm-4" style="background-color:;border:;">

<table align="center" style="margin-top:10px;">

<thead>

<tr style="color:White">

<th>Image</th>

<th>category</th>

<th>property</th>

<th>brand</th>

<th>city</th>

<th>price</th>

</tr>

</thead>

{% for adds in adds %}

<tr style="color:Black">

<td><img src="{{adds.file.url}}" style="width:10em;height:10em;"></td>

<td>{{adds.category}}</td>

<td>{{adds.property}}</td>

<td>{{adds.brand}}</td>

<td>{{adds.city}}</td>

<td>{{adds.price}}</td>

</tr>

{% endfor %}

</table>

</div>

<!--<h1><p class="mb-5">Feature Engineering for Search Advertising Recognition.Meghana</p></h1>

<p><a href="#" class="btn btn-outline-light py-3 px-5">Get Started</a></p>-->

</div>

<div class="col-lg-2 ml-auto" data-aos="fade-up" data-aos-delay="100">

<figure class="img-absolute">

</figure>

</div>

</div>

</div>

</div>

{% endblock %}