CUSTOMER USER MODEL

IN MODELS.PY

KWANZA IMPORT HIVI VITU

```
from django.contrib.auth.models import User
from django.contrib.auth.models import AbstractBaseUser, BaseUserManager
from django.db import models
from django.urls import reverse
```

THEN TENGENEZA HIZI CODES

```
class MyUserManager(BaseUserManager):
  def create_user(self, email, username, password=None):
    if not email:
      raise ValueError("email is required")
    if not username:
      raise ValueError("Your user name is required")



    user=self.model(
      email=self.normalize_email(email),
      username=username,


    )
    user.set_password(password)
    user.save(using=self._db)
    return user
  def create_superuser(self, email, username, password=None):
    user=self.create_user(
      email=self.normalize_email(email),
      username=username,
      password=password,

    )
    user.is_admin=True
    user.is_staff=True

    user.is_superuser=True
    user.save(using=self._db)
    return user



 #HII NI PATH KWA AJILI YA KUHIFADHI HIZO IMAGE
def get_profile_image_filepath(self, filename):
  return f'profile_images/{self.pk}/{"44.jpg"}'
```

```
#HII NI KWA AJILI YA KUPATA DEFAULT IMAGE KM MTU ASIPO INGIZA IMAGE ILI ISILETE
ERRORS
def get_default_profile_image():
    return "media/44.jpg"
```

THEN BAADA YA HAPO TENGENEZA HII MODEL

```
class MyUser(AbstractBaseUser):
    email=models.EmailField(verbose_name="email", max_length=100, unique=True)
    first_name=models.CharField(verbose_name="first name", max_length=100, unique=False)
    username=models.CharField(verbose_name="user name", max_length=100, unique=True)
    middle_name=models.CharField(verbose_name="middle name", max_length=100,
unique=False)
    last_name=models.CharField(verbose_name="last name", max_length=100, unique=False)
    company_name=models.CharField(verbose_name="company name", max_length=100,
unique=False)
    phone=models.CharField(verbose_name="phone", max_length=15)
    profile_image = models.ImageField(upload_to='get_profile_image_filepath', blank=True,
null=True)
    date_joined=models.DateTimeField(verbose_name="date joined", auto_now_add=True)
    last_login=models.DateTimeField(verbose_name="last login", auto_now=True)
    is_admin=models.BooleanField(default=False)
    is_active=models.BooleanField(default=True)
    is_staff=models.BooleanField(default=True)
    is_superuser=models.BooleanField(default=False)
    hide_email = models.BooleanField(default=True)



    USERNAME_FIELD="email"
    REQUIRED_FIELDS=['username']

    objects=MyUserManager()

    def __str__(self):
        return self.username

    def get_profile_image_filename(self):
        return str(self.profile_image)[str(self.profile_image).index(f'profile_images/{self.pk}/'):]


    def has_perm(self, perm, obj=None):
        return True

    def has_module_perms(self, app_label):
        return True
```

THEN NENDA KWENYE FILE LA SETTINGS.PY KAANDIKE HIKI KITU

```
WSGI_APPLICATION = 'DimosoProject.wsgi.application'
AUTH_USER_MODEL="DimosoApp.MyUser"

# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases
```

IN ADMIN.PY

```
from django.contrib import admin
from django.contrib.auth.admin import UserAdmin as BaseUserAdmin
from DimosoApp.models import *


# Register your models here.
class MyUserAdmin(BaseUserAdmin):
   list_display=('username', 'email', 'company_name', 'profile_image', 'date_joined', 'last_login',
'is_admin', 'is_active')
   search_fields=('email', 'first_name', 'last_name')
   readonly_fields=('date_joined', 'last_login')
   filter_horizontal=()
   list_filter=('last_login',)
   fieldsets=()

   add_fieldsets=(
      (None,{
         'classes':('wide'),
         'fields':('email', 'username', 'first_name', 'middle_name', 'last_name', 'company_name',
'phone', 'profile_image', 'password1', 'password2'),
      }),
   )

   ordering=('email',)
```

SASA KWA MFANO UKITAKA KUMREGISTER MTU KWA KUTUMIA COSTOMER USER MODEL

KWENYE VIEWS.PY YA HIYO VIEW YA REGISTRATION ANDIKA HAYA

HIZO KODE ZA EMAILACHANA NAZO HZO NI KWA AJILI YA MTU AKIJISAJILI ATUMIWE EMAIL CODE ZAKE UTAZIPATA KWENYE HOW TO SEND EMAIL WHEN USER REGISTER.

```
#HII FORM NI KWA AJILI YA USER AKITAKA KUJISALI MWENYEWE
def registration(request, *args, **kwargs):
   user = request.user
   if user.is_authenticated:
        return HttpResponse(f"you are already authenticated as {user.email}")
    context = {}

    #form = MyUserForm()

    if request.POST:

        form=MyUserForm(request.POST)
        if form.is_valid():
            form.save()

            #HIZI NI KWA AJILI KUTUMA EMAIL ENDAPO MTU AKIJISAJILI
            username = request.POST['username']
            #last_name = request.POST['last_name']
            email = request.POST['email']
            subject = "Welcome to Dimoso El Blog"
            message = f"Hey {username}  we will help you become the best coder you can ever be"
            from_email = settings.EMAIL_HOST_USER
            recipient_list = [email]
            send_mail(subject, message, from_email, recipient_list, fail_silently=True)

            #ZINAISHIA HAPA ZA KUTUMA EMAIL



            email = form.cleaned_data.get('email').lower()
            raw_password = form.cleaned_data.get('password1')
            myuser = authenticate(email=email,password=raw_password)
            if myuser is not None:
                        login(request,myuser)
                        #messages.success(request,"account created Successfully, now you can login.")

                        return redirect('home')



        else:
            context['registration_form'] = form






    return render(request,'account/registration.html', context)
```

USER_LOGIN

```
def user_login(request):
    context={}
    if request.POST:
        form=UserLoginForm(request.POST)
        if form.is_valid():
            email=request.POST['email']
            password=request.POST['password']
            user = authenticate(request, email=email,password=password)

            if user is not None:
                login(request,user)
                return redirect('home')
            messages.success(request, "password or username is incorrect")
        else:
            context['login_form']=form

    else:
        #messages.success(request, "password or username is incorrect")
        form=UserLoginForm(request.POST)
        context['login_form']=form

    return render(request,'account/user_login.html', context)
```

USER_LOGOUT

```
def user_logout(request):
    logout(request)
    return redirect('user_login')
    return render(request,'account/logout.html')
```

URLS.PY OF APPLICATION

```
from django.urls import path
from .import views

urlpatterns = [


    path('', views.user_login, name="user_login"),
    #path('user_register', views.user_register.as_view(), name="user_register"),
    path('user_logout', views.user_logout, name="user_logout"),
    path('registration', views.registration, name="registration"),



]
```

TEMPLATES

REGISTRATION.PY

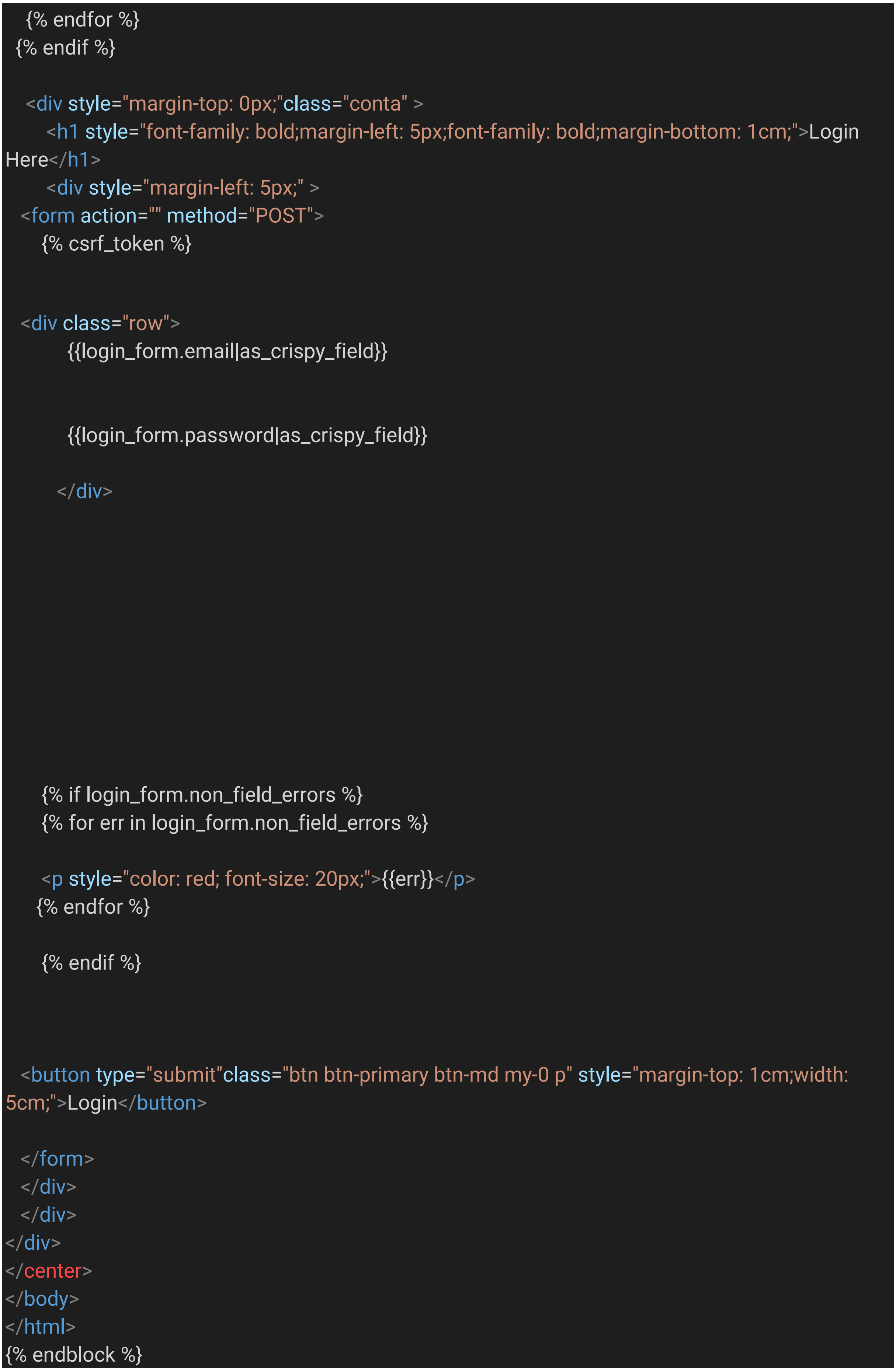
```
{% extends 'DimosoApp/base.html' %}
{% load static %}
{% load crispy_forms_tags %}
{% block title %}Register {% endblock %}


{% block content %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style type="text/css">
      body{
 background-color: wheat;
  overflow-x: hidden;
}


input{
  width: 100%;
  padding: 15px;
  margin: 5px;
  border-radius: 10px;
  border: 2px solid blue;
}
label{
  color: black;
  font-weight: bold;
  font-size: 20px;
}
input[type=submit]{
  width: 6cm;
  padding: 10px;
  margin-top: : 1cm;
  border-radius: 10px;

}


    </style>
</head>
<body>
```

```
 <div class="container-fluid" style="margin-top: 1cm;">

  <center><h1 style="font-family: bold;">USER REGISTRATION FORM</h1></center>







<center>



    <form  method="POST" class="form-group">
      {% csrf_token %}
      <div class="d-flex flex-column pb-3">

<img src="">

      </div>
      <label>Email</label><br>

      <input type="email" name="email" id="inputEmail" class="" placeholder="Email address"
required><br>
      <ul>
  <li style="color: blue;font-weight: bold;">Required. Add a valid email address.</li>
</ul>


   <label>UserName</label><br>
<input type="text" name="username" id="inputUsername" class="" placeholder="Your
username"><br>

<ul>
  <li style="color: blue;font-weight: bold;">Your username can’t be too similar to your
password</li>
</ul>





 <label>Password</label><br>
<input type="password" name="password1" id="inputPassword1" class="" placeholder="Enter
Password" required><br>

   <ul>
  <li style="color: blue;font-weight: bold;">Your password can’t be too similar to your other
personal information.</li>
```

```
 <li style="color: blue;font-weight: bold;">Your password must contain at least 8 characters.</li>
 <li style="color: blue;font-weight: bold;">Your password can’t be a commonly used 
password.</li>
 <li style="color: blue;font-weight: bold;">Your password can’t be entirely numeric.</li>

</ul>




 <label>Comfirm Password</label><br>
<input type="password" name="password2" id="inputPassword2" class="" placeholder="confirm 
Password" required><br>
<ul>
 <li style="color: blue;font-weight: bold;">Enter the same password as before, for verification.</li>
</ul>

{% for field in registration_form %}

<p>
 {% for error in field.errors %}
 <p style="color: red">{{error}}</p>
 {% endfor %}
</p>
{% endfor %}


 <input type="submit" class="btn-primary" value="Submit">
   </form>




  </center>









</div>


</body>
</html>
{% endblock %}
```

USER_LOGIN.HTML

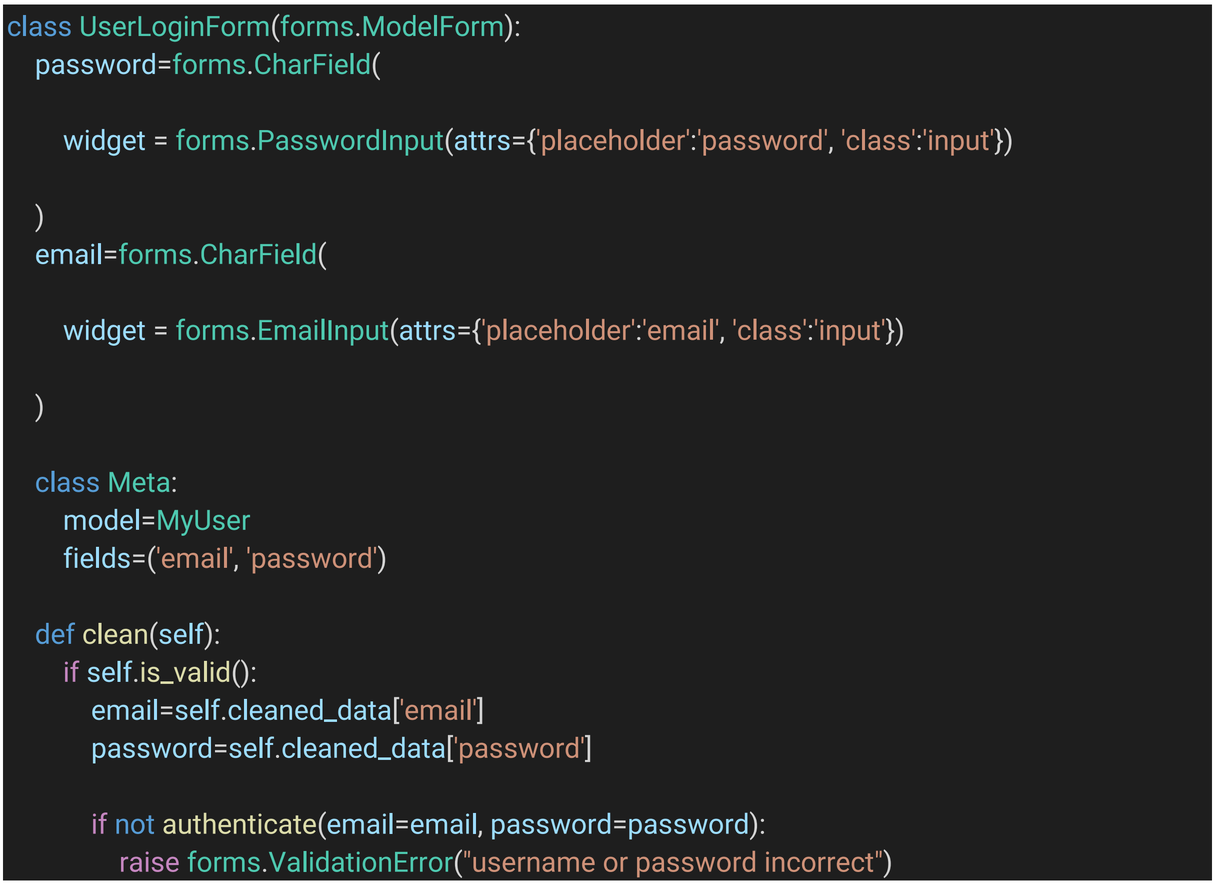
```
{% extends 'DimosoApp/base.html' %}
{% load static %}
{% load crispy_forms_tags %}
{% block title %}login {% endblock %}


{% block content %}


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

   <style type="text/css">
    body{
      overflow-x: hidden;
    }

      .input{
    width: 100%;
     margin-bottom: 2px;
    padding: 20px;
    border-radius: 15px;

  }
.conta{
  background-color: wheat;
}
   </style></head>
<body>
    <center>
<div style="margin-top: 0px;">


     {% if messages %}
    {% for message in messages %}


          <div class="alert alert-warning alert-dismissible fade show" role="alert">
  <strong>{{message}}</strong>
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div>
```



```
   {% endfor %}
  {% endif %}

   <div style="margin-top: 0px;"class="conta" >
       <h1 style="font-family: bold;margin-left: 5px;font-family: bold;margin-bottom: 1cm;">Login
Here</h1>
       <div style="margin-left: 5px;" >
  <form action="" method="POST">
      {% csrf_token %}


  <div class="row">
           {{login_form.email|as_crispy_field}}



           {{login_form.password|as_crispy_field}}

         </div>











      {% if login_form.non_field_errors %}
      {% for err in login_form.non_field_errors %}

      <p style="color: red; font-size: 20px;">{{err}}</p>
     {% endfor %}

      {% endif %}




  <button type="submit"class="btn btn-primary btn-md my-0 p" style="margin-top: 1cm;width:
5cm;">Login</button>

  </form>
  </div>
  </div>
</div>
</center>
</body>
</html>
{% endblock %}
```

IN FORMS.PY

```
from DimosoApp.models import *
from django.forms import ModelForm
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import authenticate


class MyUserForm(UserCreationForm):
    email = forms.EmailField(max_length=255, help_text="Required. Add a valid email address.")



    class Meta:
        model = MyUser
        fields = (
        "email",
        "username",
        "password1",
        "password2"


         )
    def clean_email(self):
        email = self.cleaned_data['email'].lower()
        try:
            myuser = MyUser.objects.get(email=email)
        except Exception as e:
            return email
        raise forms.ValidationError(f"Email {email} is already exist.")

    def clean_username(self):
        username = self.cleaned_data['username']
        try:
            myuser = MyUser.objects.get(username=username)
        except Exception as e:
            return username
        raise forms.ValidationError(f"username {username} is already exist.")
```



```
class UserLoginForm(forms.ModelForm):
    password=forms.CharField(

        widget = forms.PasswordInput(attrs={'placeholder':'password', 'class':'input'})

    )
    email=forms.CharField(

        widget = forms.EmailInput(attrs={'placeholder':'email', 'class':'input'})

    )

    class Meta:
        model=MyUser
        fields=('email', 'password')

    def clean(self):
        if self.is_valid():
            email=self.cleaned_data['email']
            password=self.cleaned_data['password']

            if not authenticate(email=email, password=password):
                raise forms.ValidationError("username or password incorrect")
```