::SOURCE CODE::

from django.db import models

from django.contrib.auth.models import AbstractBaseUser, PermissionsMixin, BaseUserManager

from django.core.validators import RegexValidator

class NewUser(AbstractBaseUser, PermissionsMixin):

email = models.EmailField(unique=True)

username models.CharField(max\_length-150, unique=True) =

phone\_regex RegexValidator (regex=r'^\+?1?\d(9,15)$', message="Please enter a valid phone number") = phone\_number= models.CharField(validators-[phone\_regex], max\_length-17, blank=True)

is staff models.BooleanField(default-False)

is\_active models.BooleanField(default-False) =objects NewUserAccountManager()

USERNAME FIELD = 'email'

REQUIRED FIELDS ['username', 'phone\_number"]

def str\_(self):

return self.email

class NewUserAccountManager(BaseUserManager):

def create\_superuser(self, username, email, phone\_number, password, \*\*other\_fields): other fields.setdefault('is\_staff', True) other\_fields.setdefault('is\_superuser', True) other fields.setdefault('is\_active', True)

if other fields.get('is\_staff') is not True: raise ValueError('Superuser must be assigned to is\_staff=True')

if other fields.get('is\_superuser') is not True: raise ValueError('Superuser must be assigned to is superuser=True')

user = self.create\_user(username, email, phone number, password, other \_fields) user.set\_password(password)

user.save()

return user

def create\_user(self, username, email, phone\_number, password, "\*other\_fields): if not email:

raise ValueError("Email address is required!')

email = self.normalize\_email(email

if password is not None:

else:

user=self.model (username=username, email-email, phone\_number=phone\_number, password-password,\*\*user.set\_unusable\_password()

from django import forms

from django.contrib.auth import get\_user\_model

User = get\_user\_model()

class RegisterForm(forms.Form):

username =forms.CharField(label="Username")

email = forms. EmailField(label="Email")

phone\_number forms.CharField(label="Enter your phone number here")

user= self.model (username=username, email-email, phone\_number=phone\_number, password-password, \*\* user.save()

user.save()

return user

class RegisterForm(forms.Form):

username - forms.CharField(label="Username") email - forms.EmailField(label="Email") phone\_number forms.CharField(label="Enter your phone number here")

def clean\_username(self):

username" self.cleaned\_data.get('username') user\_details User.objects.filter(username=username)

if user\_details.exists():

raise forms.ValidationError("Username is already taken")

return username

def clean\_email(self):

email- self.cleaned\_data.get('email')

If user\_details.exists():

user\_details User.objects.filter(email-email) raise forms.ValidationError("There is already an account with this email, please try logging in!") return email

def clean\_phone\_number(self):

phone\_number = self.cleaned\_data.get('phone\_number')

user\_details User.objects.filter(phone\_number-phone\_number)

if user\_details.exists():

raise forms.ValidationError("An account with this phone number already exists! Please try registering

return phone\_number

]

class LoginForm(forms.Form):

email forms. EmailField(label-"Email")

def clean\_email(self):

email self.cleaned\_data.get('email') user\_details User.objects.filter(email-email) = if not user\_details.exists(): raise forms.ValidationError("No account exists with this email. Try registering") return email

from django.shortcuts import render, redirect from django.contrib.auth import authenticate, login, get\_user\_model

from forms import RegisterForm

User- get\_user\_model()

def register\_page(request):

form RegisterForm(request.POST or None) context = ["form": form]

}

if form.is\_valid(:

username form.cleaned\_data.get("username") email form.cleaned\_data.get("email") new\_user User.objects.create\_user(username, email, phone\_number, password=None)

`phone\_number form.cleaned\_data.get("phone\_number")

return redirect("/login”)

return render (request, "auth/register.html", context)

<html>

<head>

<style type="text/css">

body, p, div {

font-family: Helvetica, Arial, sans-serif;

font-size: 14px;

CHAN

VAJG

}

a {

text-decoration: none;

</style>

<title></title>

</head>

<body>

<center>

<p>

The verification code is: <strong> {{twilio\_code}} </strong>

</p>

<p>

<span style="font-size: 10px;"><a href=".">Email preferences</a></span>

</center>

</body> </html>

SECRET\_KEY = "add secret key here" EMAIL\_HOST = "smtp.sendgrid.net"

EMAIL\_PORT = 587

EMAIL\_USE\_TLS = True

EMAIL\_HOST\_USER = "apikey"

EMAIL\_HOST\_PASSWORD= "add sendgrid api key here"

DEFAULT\_FROM\_EMAIL "add from email address here"

LOGIN\_REDIRECT\_URL = "/home"

ACCOUNT\_SID = 'add account sid here'

AUTH\_TOKEN "add auth token here"

SERVICE\_ID I'add service id here'

TEMPLATE\_ID= 'add template id here'

from pathlib import Path

from dotenv import load\_dotenv

import os

load\_dotenv()

##email details

EMAIL\_HOST = os.getenv('EMAIL\_HOST')

EMAIL\_PORT = os.getenv('EMAIL\_PORT')

EMAIL\_USE\_TLS os.getenv('EMAIL\_TLS')

EMAIL\_HOST\_USER = os.getenv('EMAIL\_HOST\_USER')

EMAIL\_HOST\_PASSWORD os.getenv("EMAIL\_HOST\_PASSWORD")

DEFAULT\_FROM\_EMAIL = os.getenv('DEFAULT\_FROM\_EMAIL')

##other details

LOGIN\_REDIRECT\_URL os.getenv('LOGIN\_REDIRECT\_URL')

ACCOUNT\_SID = os.getenv('ACCOUNT\_SID')

AUTH\_TOKEN- os.getenv('AUTH\_TOKEN') SERVICE\_ID = os.getenv('SERVICE\_ID')

TEMPLATE ID os.getenv('TEMPLATE\_ID')

from django.contrib.auth.backends import ModelBackend from django.contrib.auth import get\_user\_model

class PasswordlessAuthBackend (ModelBackend): """Log in to Django without providing a password.

def authenticate(self, request, email): User get\_user\_model() try:

user User.objects.get(email-email) = except User.DoesNotExist:

return user

return None

def get\_user(self, user\_id): User get\_user\_model() try:

return User.objects.get(pk=user\_id) except User.Does NotExist:

return None

from django.contrib.auth.backends import ModelBackend from django.contrib.auth import get\_user\_model

class PasswordlessAuthBackend (ModelBackend): """Log in to Django without providing a password.

def authenticate(self, request, email): User get\_user\_model() try: user User.objects.get(email-email) = except User.DoesNotExist:

return user

return None

def get\_user(self, user\_id): User get\_user\_model() try:

return User.objects.get(pk=user\_id) except User.Does NotExist:

return None

from django.contrib.auth.backends import ModelBackend from django.contrib.auth import get\_user\_model

class PasswordlessAuthBackend (ModelBackend):

"""Log in to Django without providing a password.”””

def authenticate(self, request, email): User get\_user\_model() try:

user User.objects.get(email-email) = except User.DoesNotExist:

return user

return None

def get\_user(self, user\_id): User get\_user\_model() try:

return User.objects.get(pk=user\_id) except User.Does NotExist:

return None

import uuid

from django.shortcuts import render, redirect

from django.contrib.auth import authenticate, login, get\_user\_model

from django.contrib import messages

from .forms import RegisterForm, LoginForm▸

from verify import SendOTP

def login page(request):

form - LoginForm(request.POST or None)

context "form": form

if form.is\_valid():

email - form.cleaned\_data.get('email')

try:

new " User.objects.get(email-email)

## if user exists

##first: send otp to the user

SendOTP.send\_code(email)

##second:redirect to the page to enter otp

temp uuid.uuid4()

return redirect("/otp/{}/{}".format(new.pk, temp))

except Exception as e:

messages.error(request, "No such user exists!")

return render(request, "auth/login.html", context)

def generate\_otp(request, pk, uuid): return render(request, 'otp.html')

import os

from twilio.rest import Client

from django.conf import settings

# Find your Account SID and Auth Token at twilio.com/console # and set the environment variables. See http://twil.io/secure

class CheckOTP:

def check\_otp(email, secret):

account\_sid = settings.ACCOUNT\_SID

auth\_token settings.AUTH\_TOKEN

client Client(account\_sid, auth\_token) =verification check client.verify \ verification\_check

.services(settings.SERVICE\_ID)\

.verification\_checks \

.create(to email, code-secret)

return verification\_check.status

def home page(request):

return render(request, "home\_page.html")

from django.contrib import admin

from django.urls import path

from verification.views import register\_page, home page, login page, generate\_otp, check\_otp

urlpatterns = [

path('admin/', admin.site.urls),

path('register/', register\_page, name="register"),

path('check/, check\_otp, name="check\_otp"),

path('login/', login page, name="login"),

path("otp/<int:pk>/<uuid>/', generate\_otp),

path("home/', home\_page)

TEMPLATES = [

{

'DIRS': ['templates'],

'BACKEND': 'django.template.backends.django.DjangoTemplates', "OPTIONS": {

'APP\_DIRS': True,

'context\_processors': [

'django.template.context\_processors.debug', 'django.template.context\_processors.request', 'django.contrib.auth.context\_processors.auth', 'django.contrib.messages.context\_processors.messages',

#html code

<IDOCTYPE html>

<html>

<head>

</head>

<body style="background-color: cadetblue;">

<br>

<h1 style="text-align: center;"> Register yourself for an awesome shopping experience 1</h1>

<form method="POST" style="text-align:center;"> {% csrf\_token %} <div class="container" style="background-color: grey; height: 400px; width: 600px; text-align:center; borde

</br>

</br>

</br>

{{ form.as\_p }}

<button type="submit" style="text-align:center;">Submit</button> <p><b>Already have an account? </b><a href="{% url 'login' %}">Sign in</a>.</p>

</div>

</body>

</html>

!DOCTYPE html>

<html>

<head>

</head>

<body style="background-color:khaki;">

(X if messages X)

(X for message in messages X) (% if message.tags == "error" %)

<div class="alert alert-danger">

chi style="border: 5px red;"> ((message)) </h1>

</div>

{% else %)

cdiv class="alert alert-{{ message.tags }}">

<h1 style="border: 5px red;"> {{ message }} </h1>

</div>

(X endif %)

(% endfor %) {% endif %)

chi style="text-align:center;"> Welcome Back!!</h1>

ch2 style="text-align: center;"> Login to continue</h2> <div class="container" style="background-color: grey; height: 400px; width:600px; text-align:center; borde

</br>

</br>

<form method="POST" style="text-align:center;"> {% csrf\_token %)

((form.as\_p))

<button type="submit" class="btn btn-default" style="text-align:center; background-color:white;">Submit</but

<p><b> Create an account? </b><a href="{% url 'register' \*)">Register here</a>.</p>

</div>

</form>

</body>

</html>