**Difference between Abstract User and Abstract Base User?**

* If u want to do everything from scratch, like is\_login, is\_active everything u need to define by your own. – AbstractBaseUser
* If u want to just add some field without changing the whole default User model, like u want to add profile pic field etc.

**Test code for custom user model?**

* Create a directory named “tests” add \_\_init\_\_.py file.. Now add the test\_custom\_user\_model.py
* ***Code***

from django.test import TestCase

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

class ModelTests(TestCase):

def test\_create\_user\_with\_email\_successful(self):

""" First Test"""

email = 'test@example.com'

password = 'User1234'

user = get\_user\_model().objects.create\_user(

email=email,

password=password)

self.assertEqual(user.email, email)

self.assertTrue(user.check\_password(password))

def test\_normalize\_email\_succesfull(self):

"""Second Test"""

sample\_email = [('Abcd@Gmail.com', 'abcd@gmail.com'),

('abc@Exampl.Com', 'abc@exampl.com'),

('xyz@EXAMPLE.com', 'xyz@example.com'),

('XYZK@EXAMPLE.COM', 'xyzk@example.com')

]

password = 'ABCD1234'

for email, expected\_email in sample\_email:

user = get\_user\_model().objects.create\_user(

email=email,

password=password

)

self.assertEqual(user.email, expected\_email)

**Model Code:-**

"""

Database Models

"""

from django.db import models

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

# Create your models here.

class CustomUserManager(BaseUserManager):

def create\_user(self, email, password=None, \*\*extra\_fields):

if not email:

raise ValueError('Please provide a valid email address')

email = self.normalize\_email(email)

email = email.lower()

user = self.model(email=email, \*\*extra\_fields)

user.set\_password(password)

user.save(using=self.\_db)

return user

def create\_superuser(self, email, password=None):

user = self.create\_user(email, password)

user.is\_superuser = True

user.is\_staff = True

user.save(using=self.\_db)

return user

class CustomUserModel(AbstractBaseUser, PermissionsMixin):

email = models.EmailField(max\_length=255, unique=True)

name = models.CharField(max\_length=255)

is\_active = models.BooleanField(default=True)

is\_staff = models.BooleanField(default=False)

objects = CustomUserManager()

USERNAME\_FIELD = 'email'