**Name- Rushikesh Sheshram Susar**

**Reg no.-2018bit040**

**Batch -C**

**Branch-Information Technology**

**Sub -CRM Django Project**

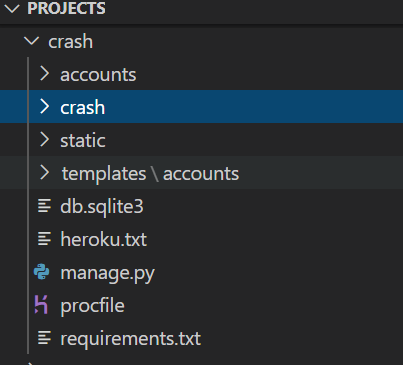
**Common Fetures:-**

1. URL routing
2. Templates inheriting
3. Models and Admin panel
4. CRUD Functionality
5. Filter form Table Search
6. Login Authentication
7. Image upload using pillow package

**Special features (New):-**

1. User roles management and permission using decorators.py
2. Django Signals-Creating Profiles with Django (when new user sign up that user automatically added to customer page)
3. Postgresql Database
4. Password Reset email
5. Contact form that sends mail to customer and website owner
6. Matplolib integration with Django
7. Barcode Generator

**Dir structure –**



**Models.py File :-**

from django.db import models

from django.contrib.auth.models import User

from django import forms

import barcode

from barcode.writer import ImageWriter

from io import BytesIO

from django.core.files import File

# Create your models here.

class Customers(models.Model):

    user = models.OneToOneField(User,null= True, on\_delete =models.CASCADE)

    name = models.CharField(max\_length=200, null=True)

    phone = models.CharField(max\_length=200, null=True)

    email = models.CharField(max\_length=200, null=True)

    profile\_pic = models.ImageField(default="profile1.png", null=True, blank=True)

    data\_created = models.DateTimeField(auto\_now\_add=True, null=True)

    def \_\_str\_\_(self):

        return str(self.name)

class Tag(models.Model):

    name = models.CharField(max\_length=200, null=True)

    def \_\_str\_\_(self):

        return self.name

class Product(models.Model):

    CATEGORY = (

        ('Indoor','Indoor'),

        ('Out door','Out door'),

    )

    name = models.CharField(max\_length=200, null=True)

    price = models.FloatField(null = True)

    category = models.CharField(max\_length=200, null=True, choices = CATEGORY)

    description = models.CharField(max\_length=200, null=True, blank = True)

    data\_created = models.DateTimeField(auto\_now\_add=True, null=True)

    tags = models.ManyToManyField(Tag)

    def \_\_str\_\_(self):

        return self.name

class Order(models.Model):

    STATUS = (

        ('Pending', 'Pending'),

        ('Out for delivery', 'Out for delivery'),

        ('Delivered', 'Delivered'),

    )

    customer = models.ForeignKey(Customers, null=True, on\_delete= models.SET\_NULL)

    product = models.ForeignKey(Product, null=True, on\_delete= models.SET\_NULL)

    data\_created = models.DateTimeField(auto\_now\_add=True, null=True)

    status = models.CharField(max\_length=200, null=True, choices = STATUS)

    note = models.CharField(max\_length=1000, null=True, choices = STATUS)

    def \_\_str\_\_(self):

        return self.product.name

class Contact(models.Model):

    sno = models.AutoField(primary\_key=True)

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

    phone = models.CharField(max\_length=13)

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

    content = models.TextField()

    timestamp = models.DateTimeField(auto\_now\_add=True,blank=True)

    def \_\_str\_\_(self):

        return 'message from  ' + self.name + '   -' + self.email

class Newsletter(models.Model):

    email = models.CharField(max\_length=100,default=False)

    timestamp = models.DateTimeField(auto\_now\_add=True,blank=True)

    def \_\_str\_\_(self):

        return 'Subscribed for your newsletter  ' + self.email

class Score(models.Model):

    result = models.PositiveIntegerField()

    def \_\_str\_\_(self):

        return str(self.result)

class Barcode(models.Model):

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

    barcode = models.ImageField(upload\_to='image/',blank=True)

    country\_id = models.CharField(max\_length=1,null=True)

    manufacturer\_id = models.CharField(max\_length=6,null=True)

    number\_id = models.CharField(max\_length=5,null=True)

    def \_\_str\_\_(self):

        return str(self.name)

    def save(self,\*args,\*\*kwargs):

        EAN = barcode.get\_barcode\_class('ean13')

        ean = EAN(f'{self.country\_id}{self.manufacturer\_id}{self.number\_id}',writer=ImageWriter())

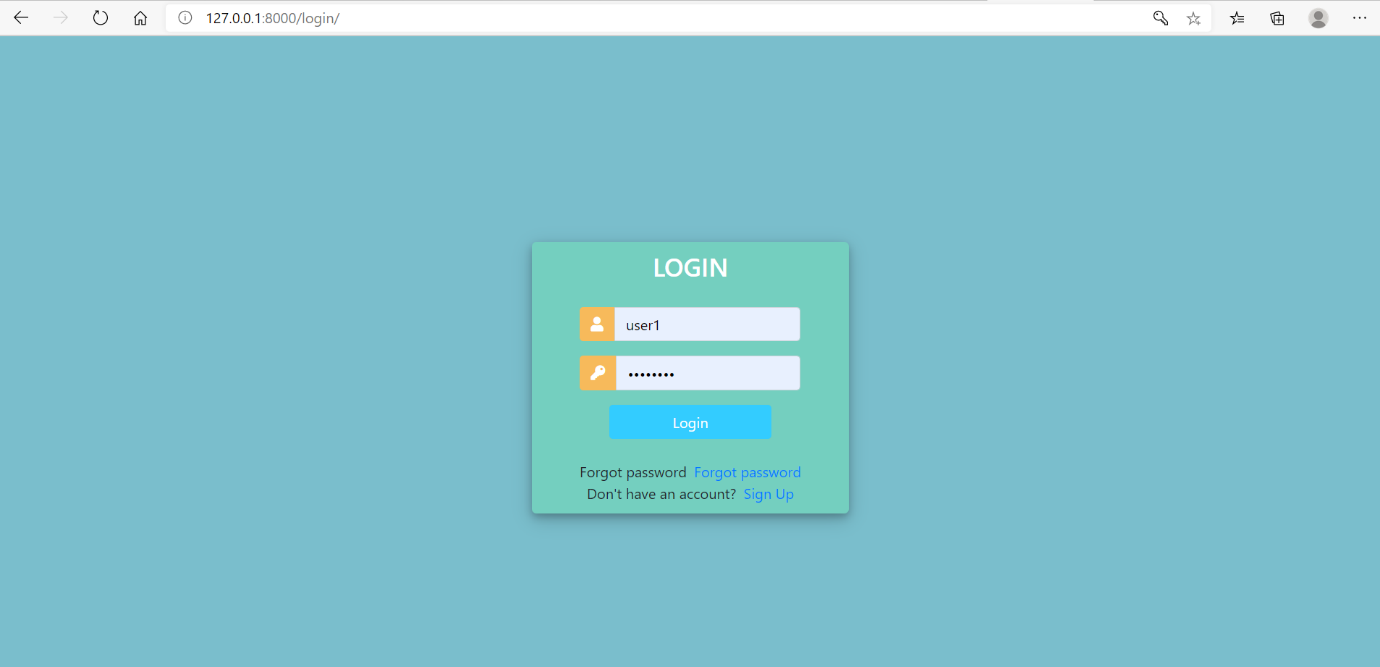
        buffer = BytesIO()

        ean.write(buffer)

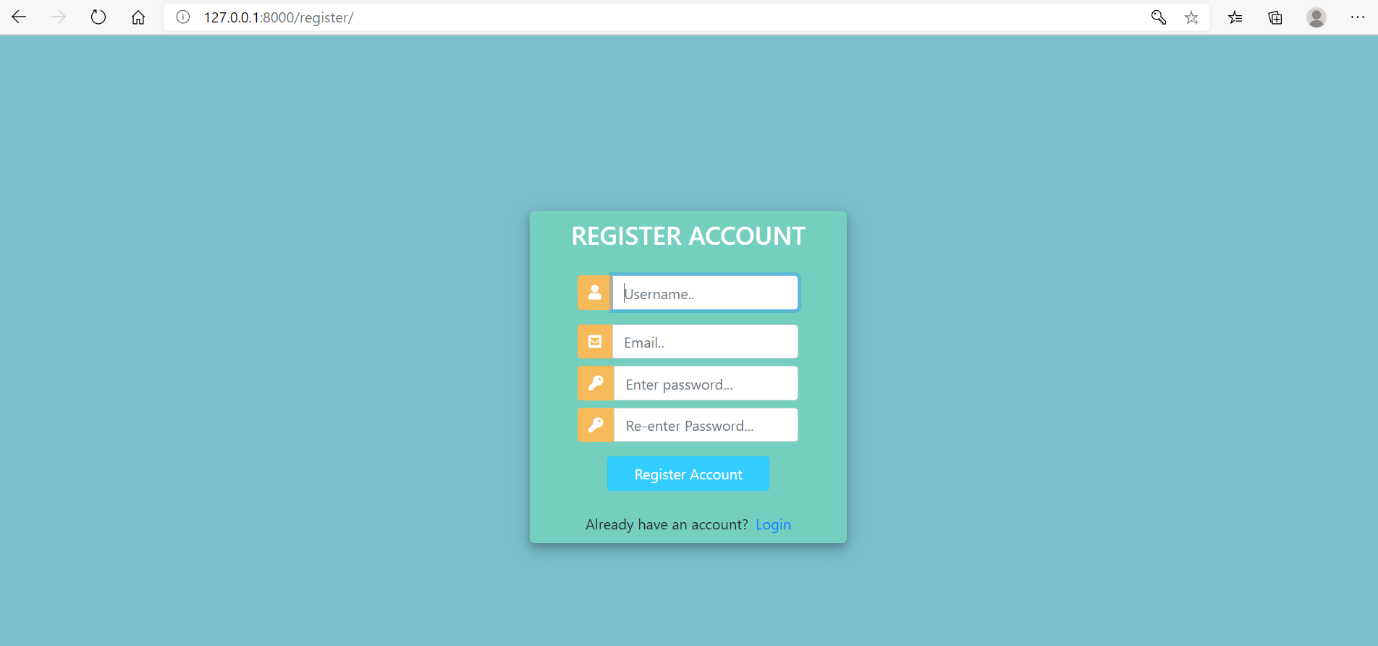
        self.barcode.save('barcode.png',File(buffer), save=False)

        return super().save(\*args, \*\*kwargs)

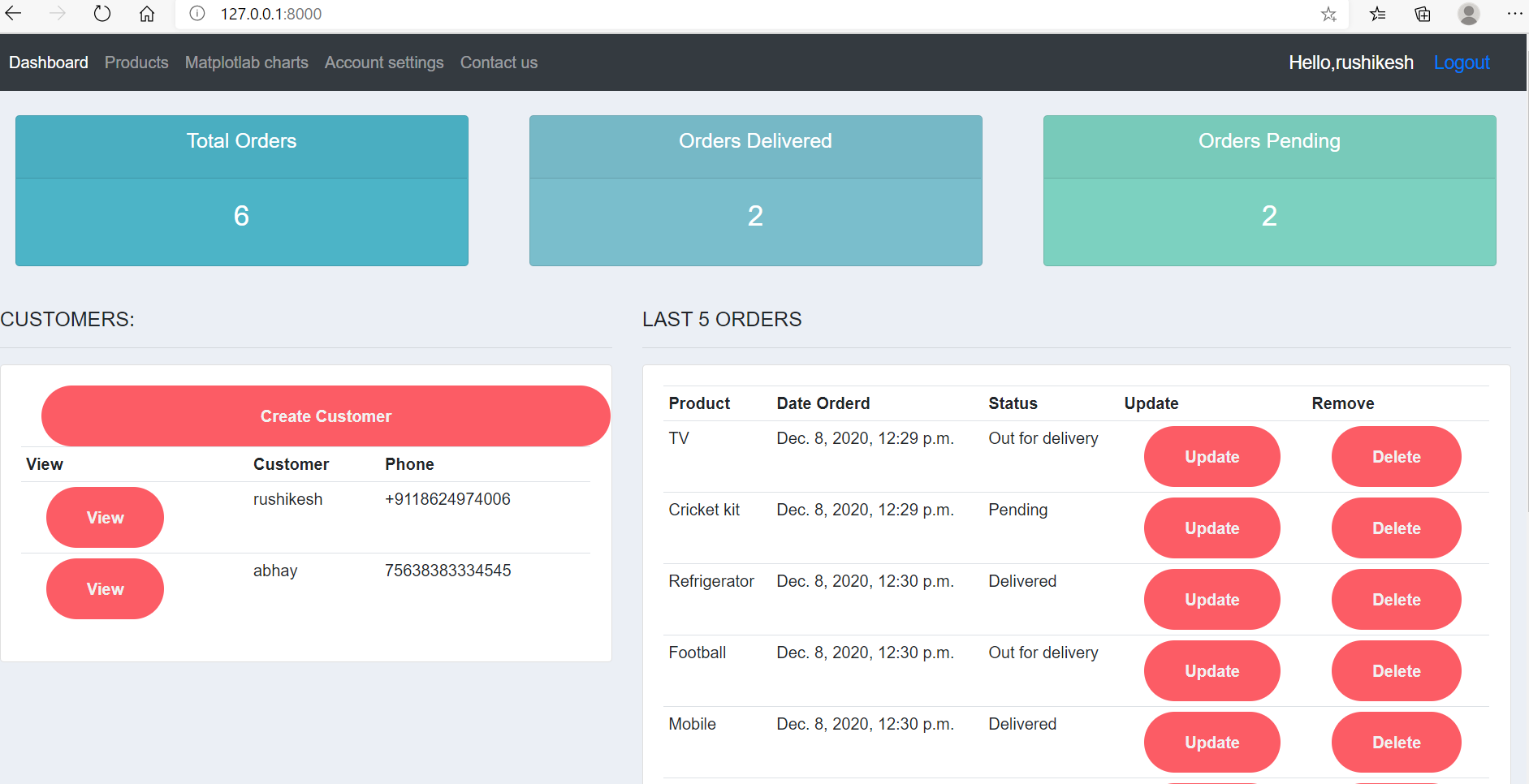
**GUI :- Login page**

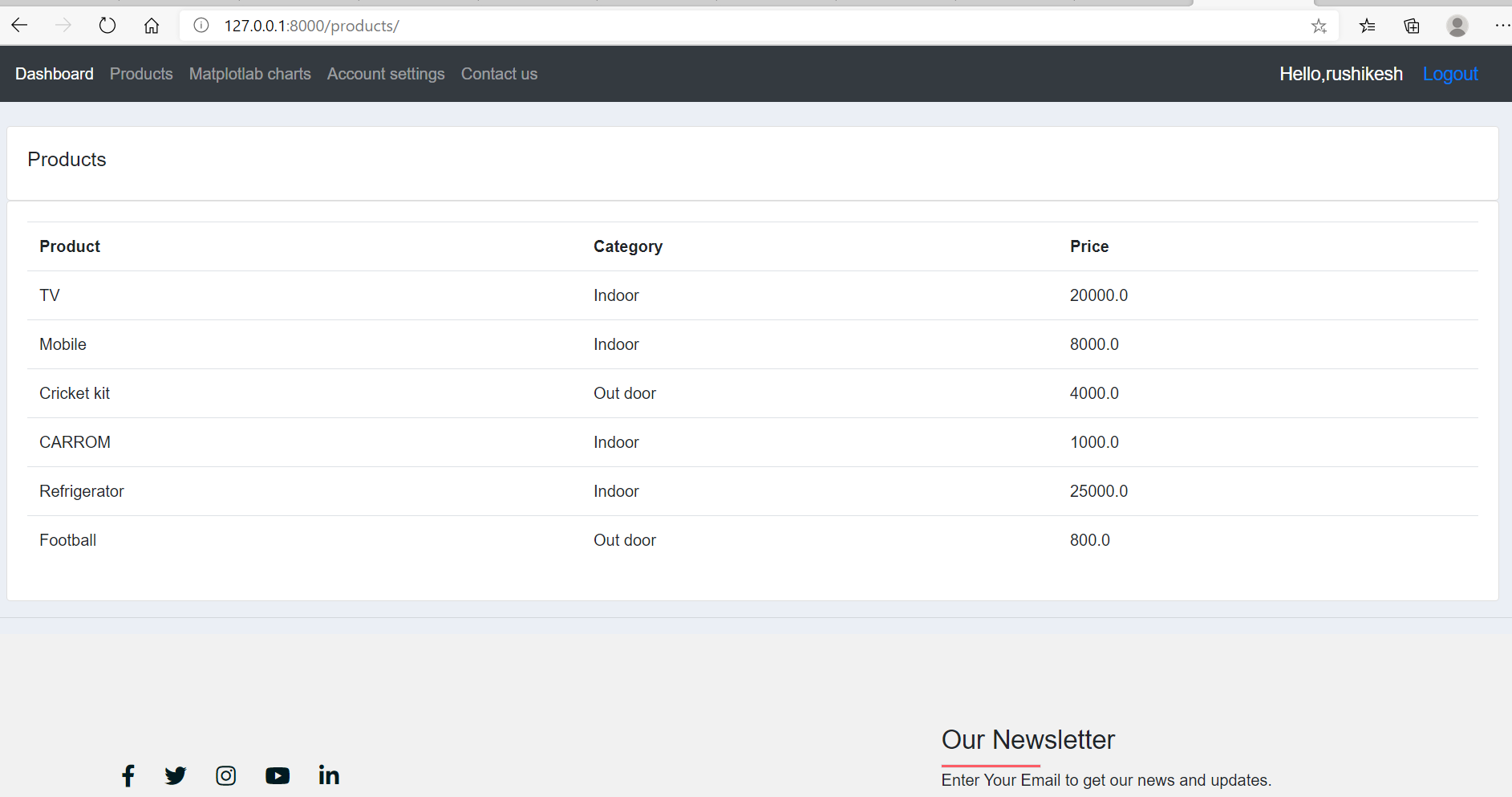


Register Page:-

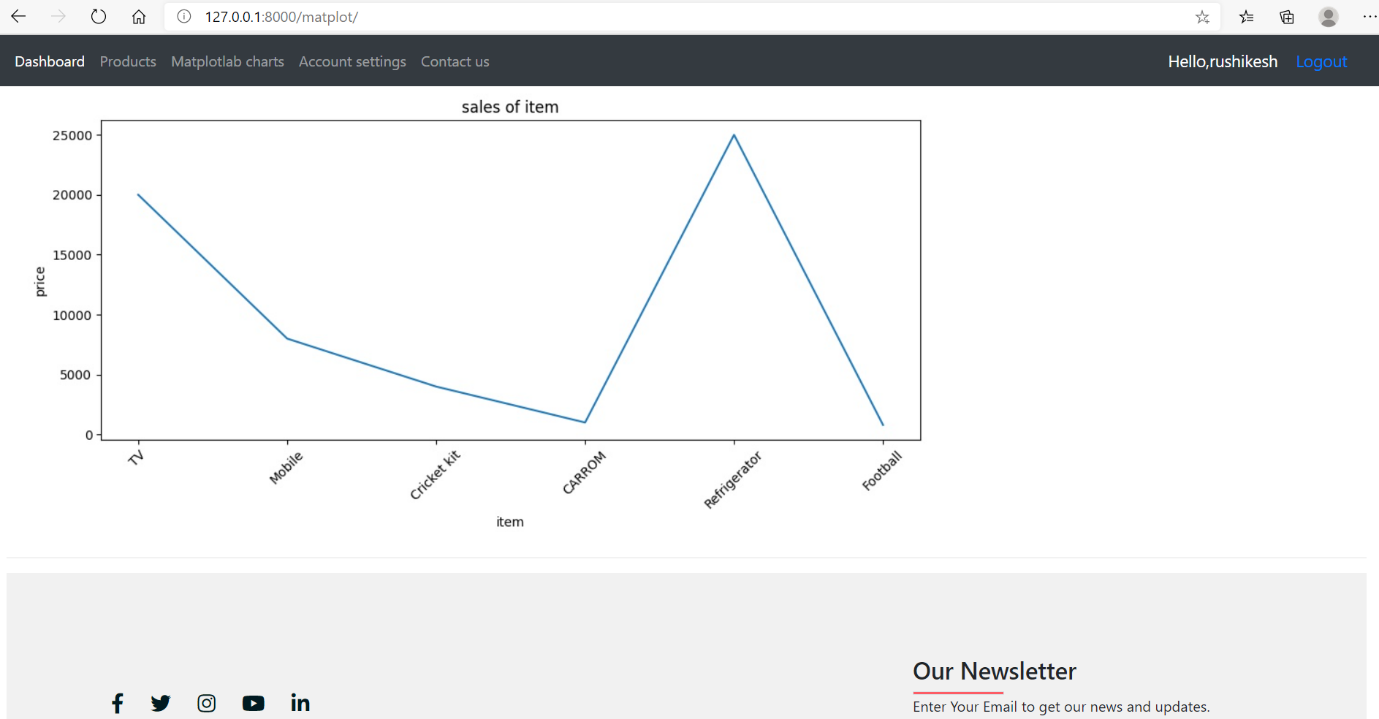


**Dashboard page:\_**

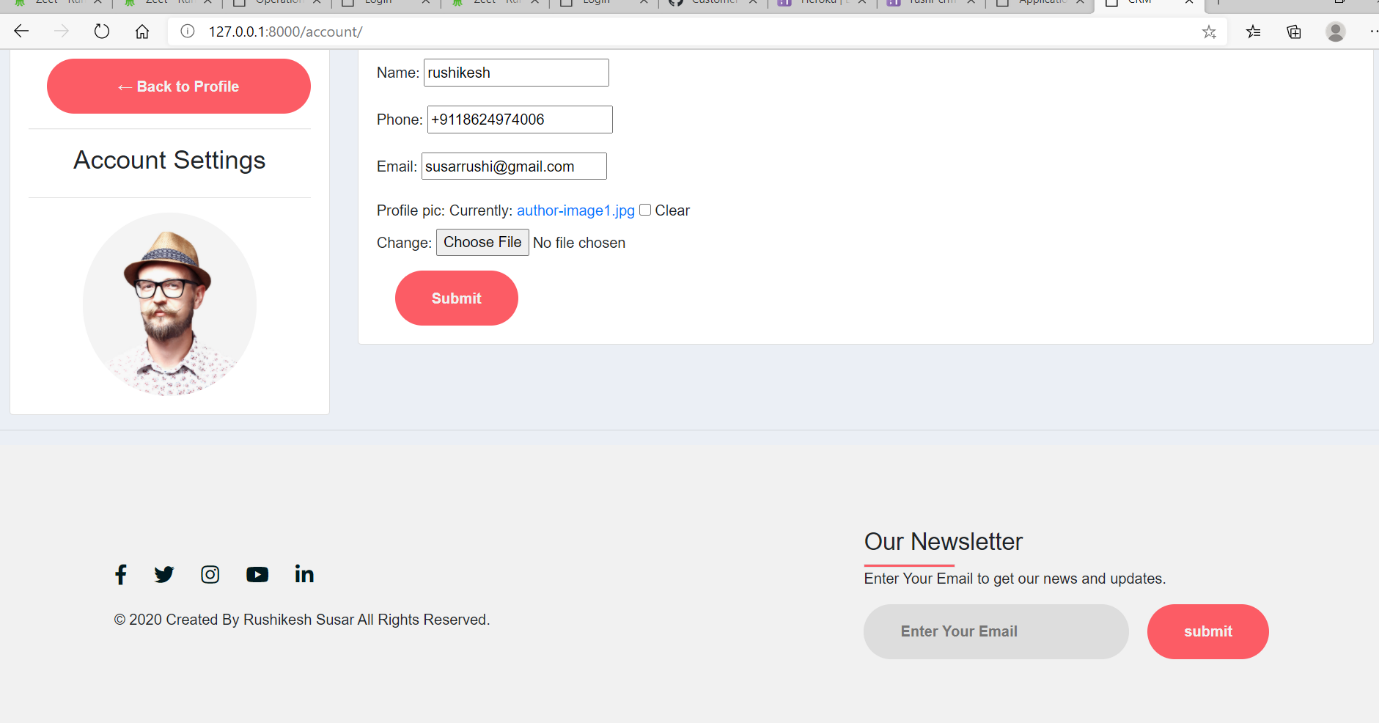


Products Page:-

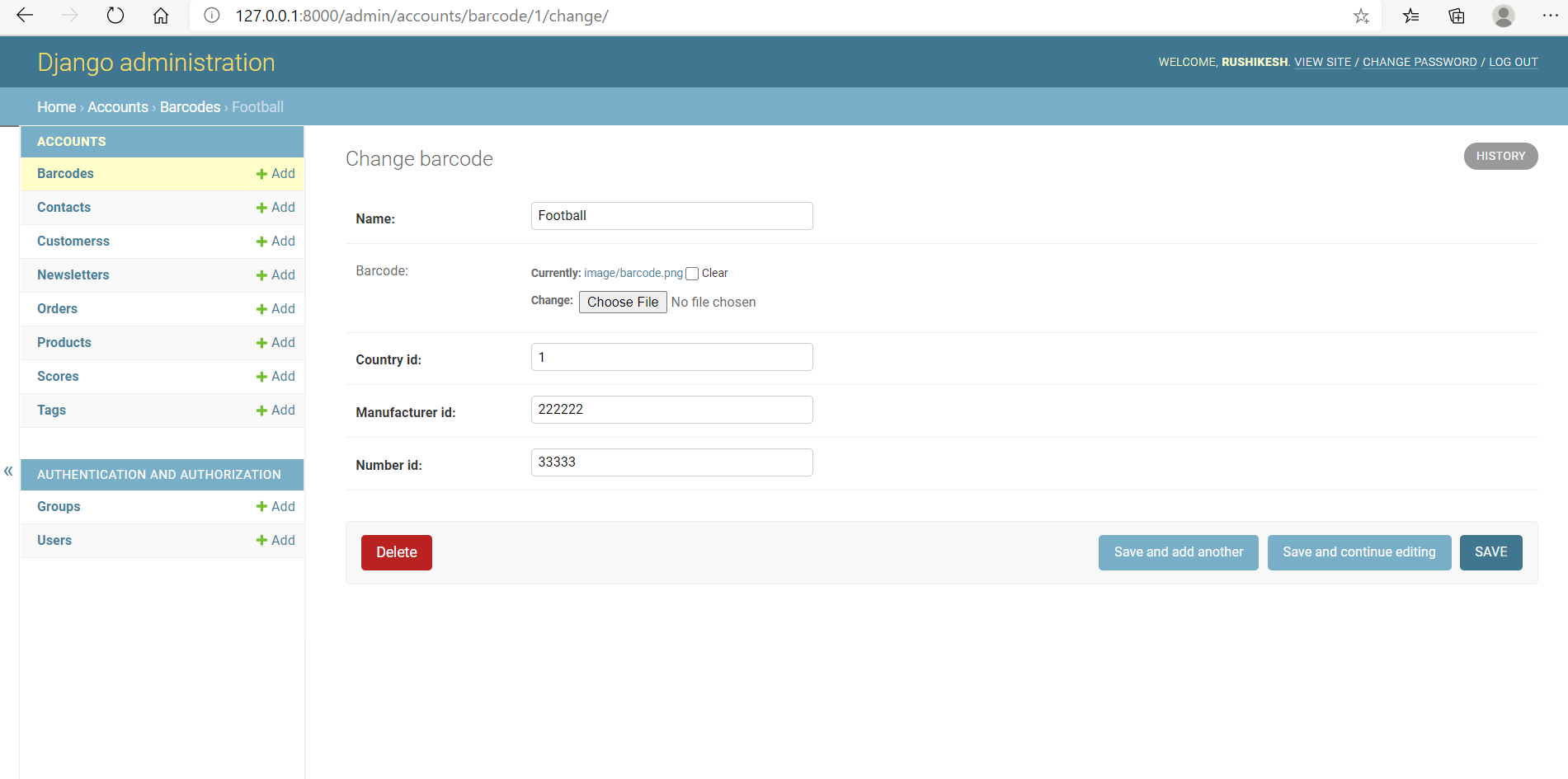
**Matplotlib page:- (New feature)**



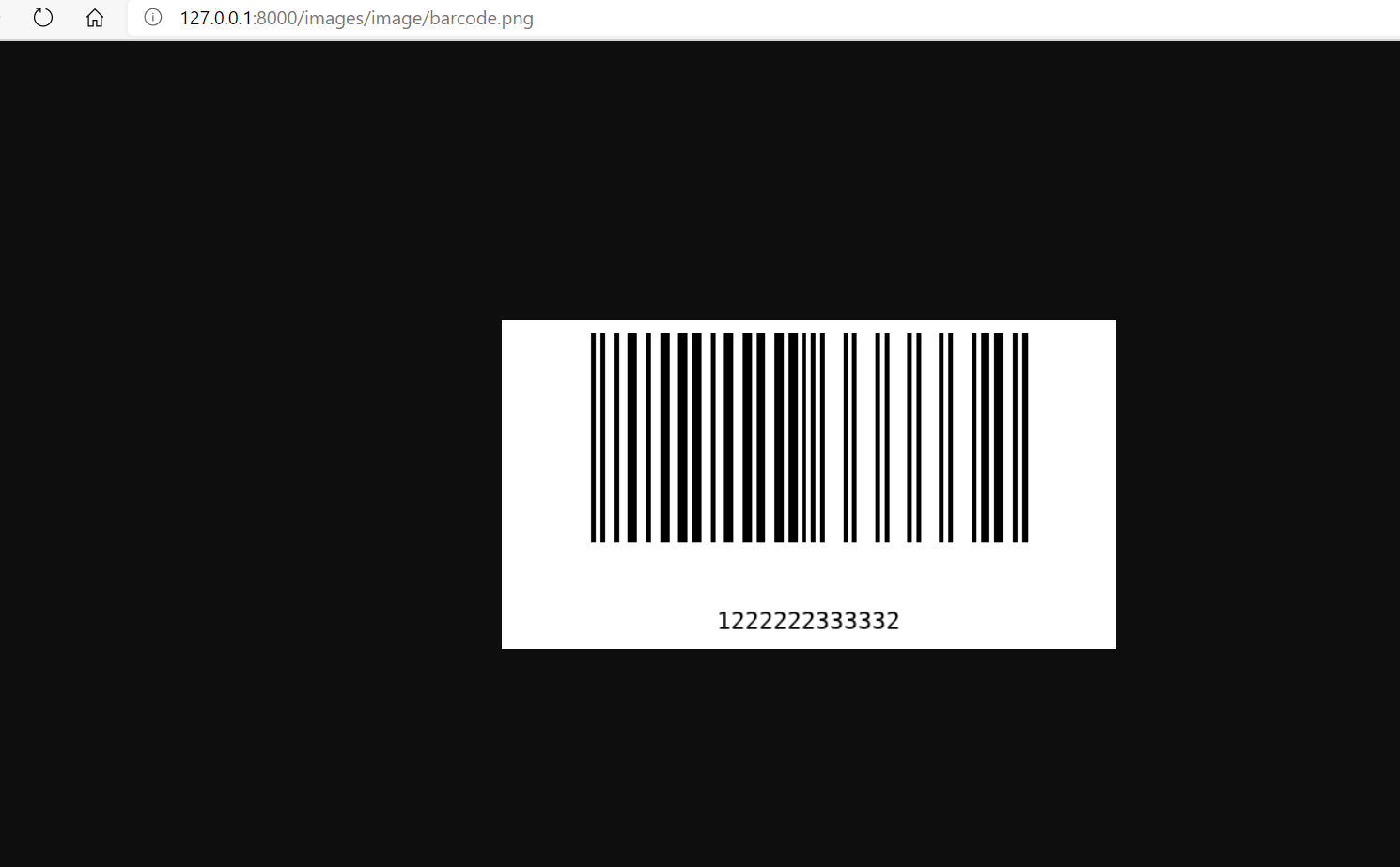
**Account settings Page:-**



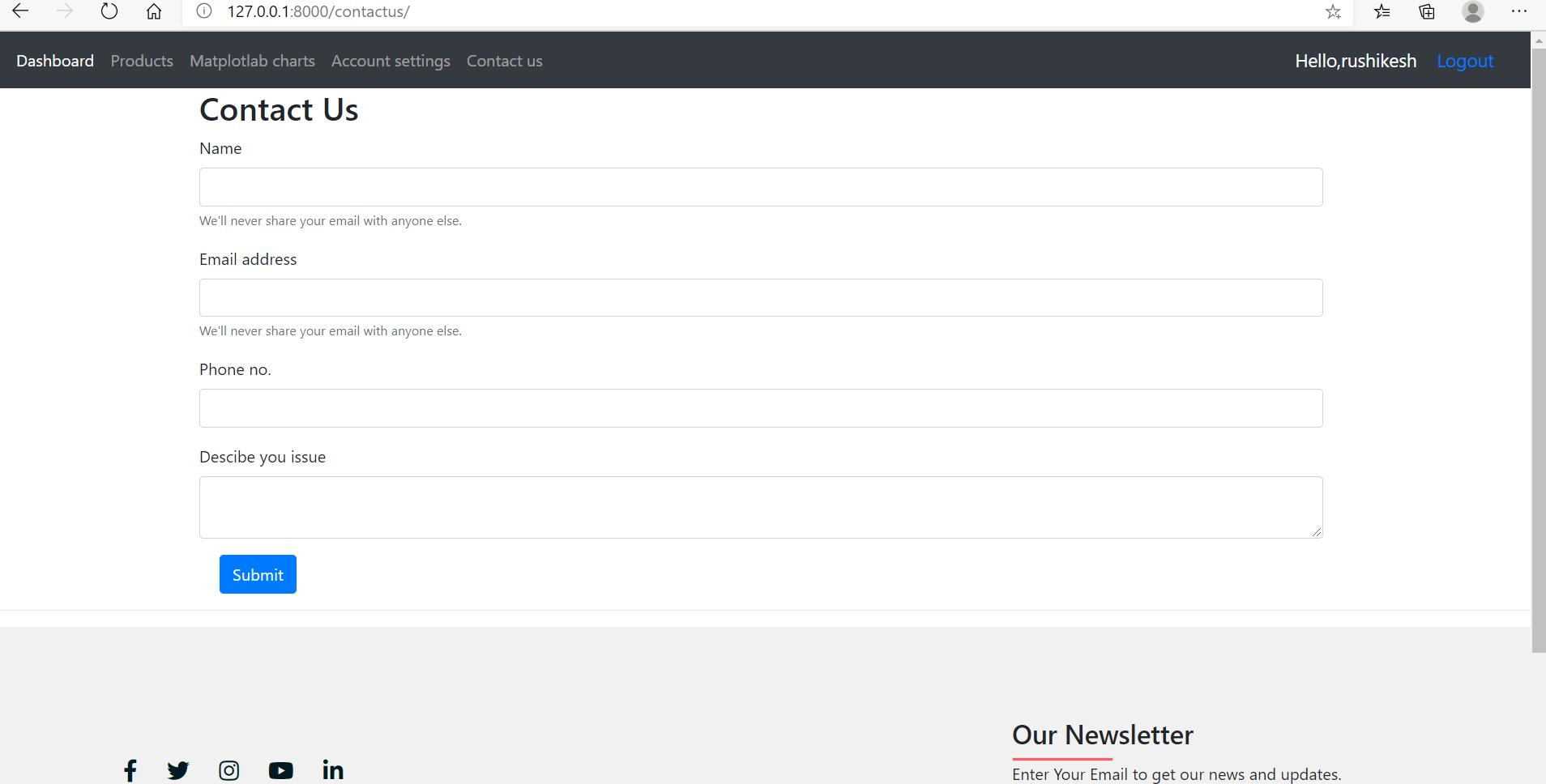
**Django barcode app admin page:- (new feature)**



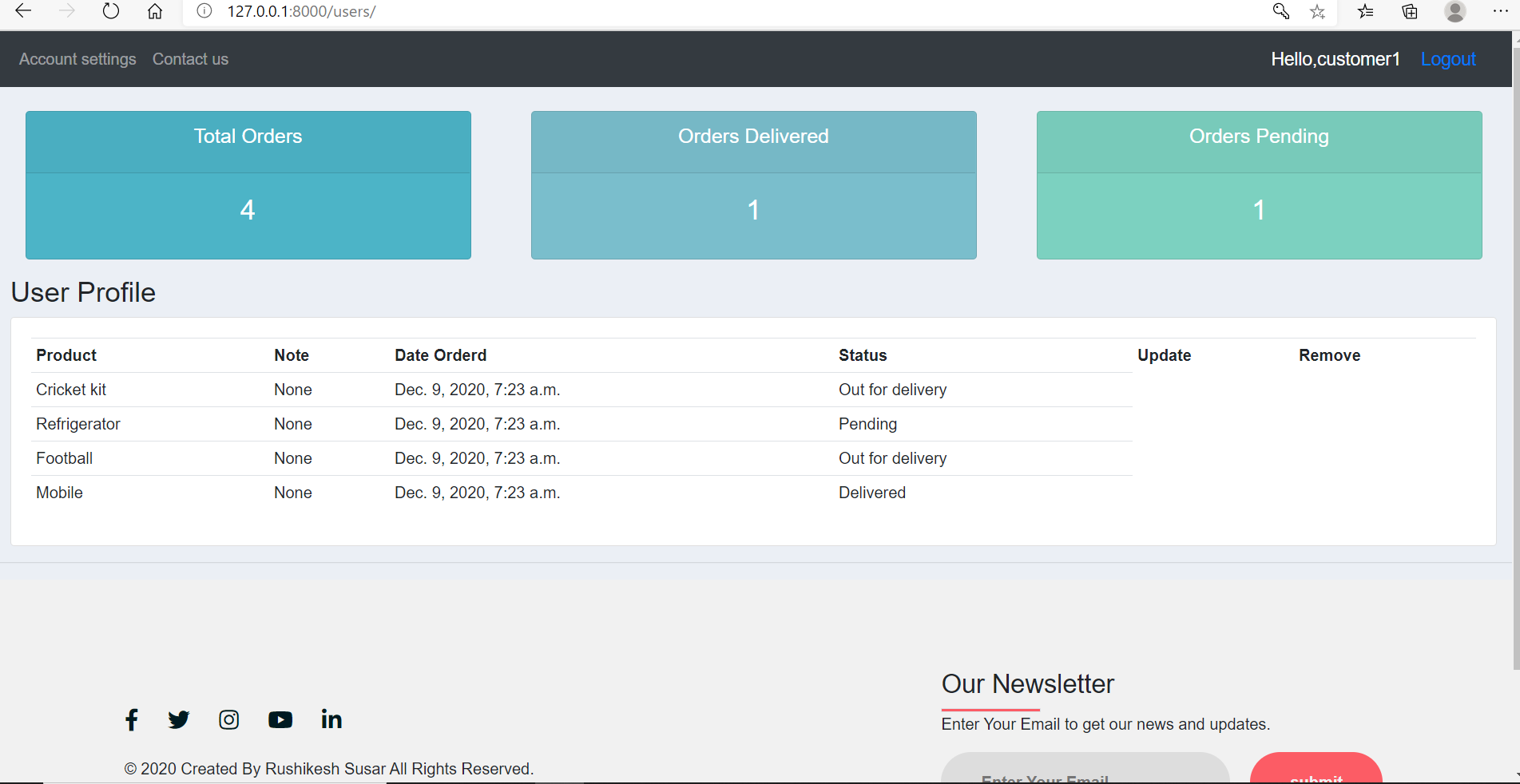
**Barcode Generated:-**



**Contact us:- (new feature)**



**User Dashboard:-**



**New Features Code:-**

* **Signals.py:-**

from django.db.models.signals import post\_save

from django.contrib.auth.models import User

from django.contrib.auth.models import Group

from .models import Customers

def customer\_profile(sender,instance,created,\*\*kwargs):

    if created:

        group = Group.objects.get(name='customer')

        instance.groups.add(group)

        Customers.objects.create(

            user=instance,

            name =instance.username

            )

        print('profile was created')

post\_save.connect(customer\_profile,sender=User)

* **utils.py :- (matplotlib code)**
* import matplotlib.pyplot as plt
* import base64
* from io import BytesIO
* def get\_graph():
* buffer = BytesIO()
* plt.savefig(buffer,format = 'jpg')
* buffer.seek(0)
* image\_png = buffer.getvalue()
* graph = base64.b64encode(image\_png)
* graph = graph.decode('utf-8')
* buffer.close()
* return graph
* def get\_plot(x,y):
* plt.switch\_backend('AGG')
* plt.figure(figsize=(10,5))
* plt.title('sales of item')
* plt.plot(x,y)
* plt.xticks(rotation=45)
* plt.xlabel('item')
* plt.ylabel('price')
* plt.tight\_layout()
* graph = get\_graph()
* return graph