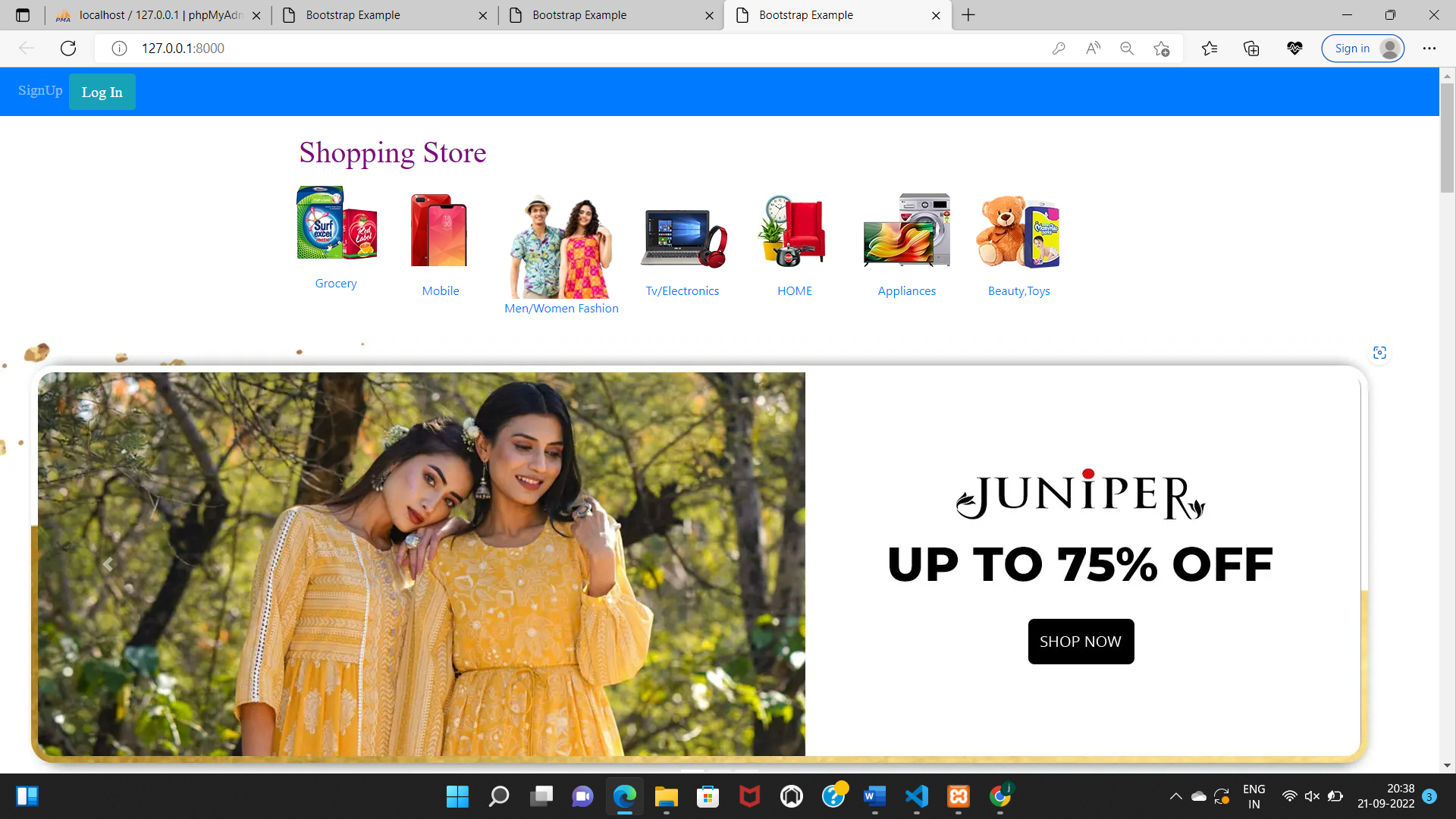
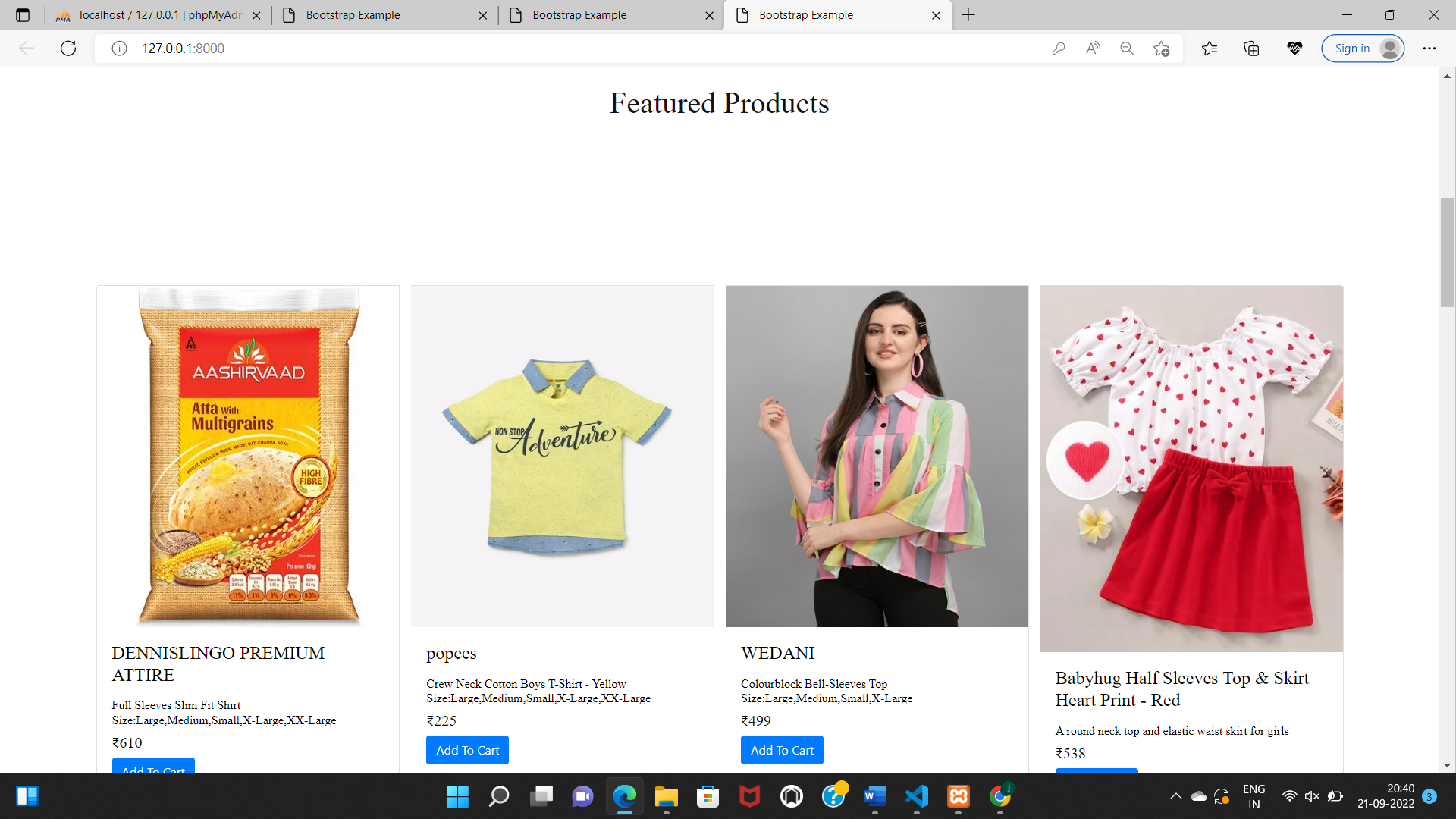
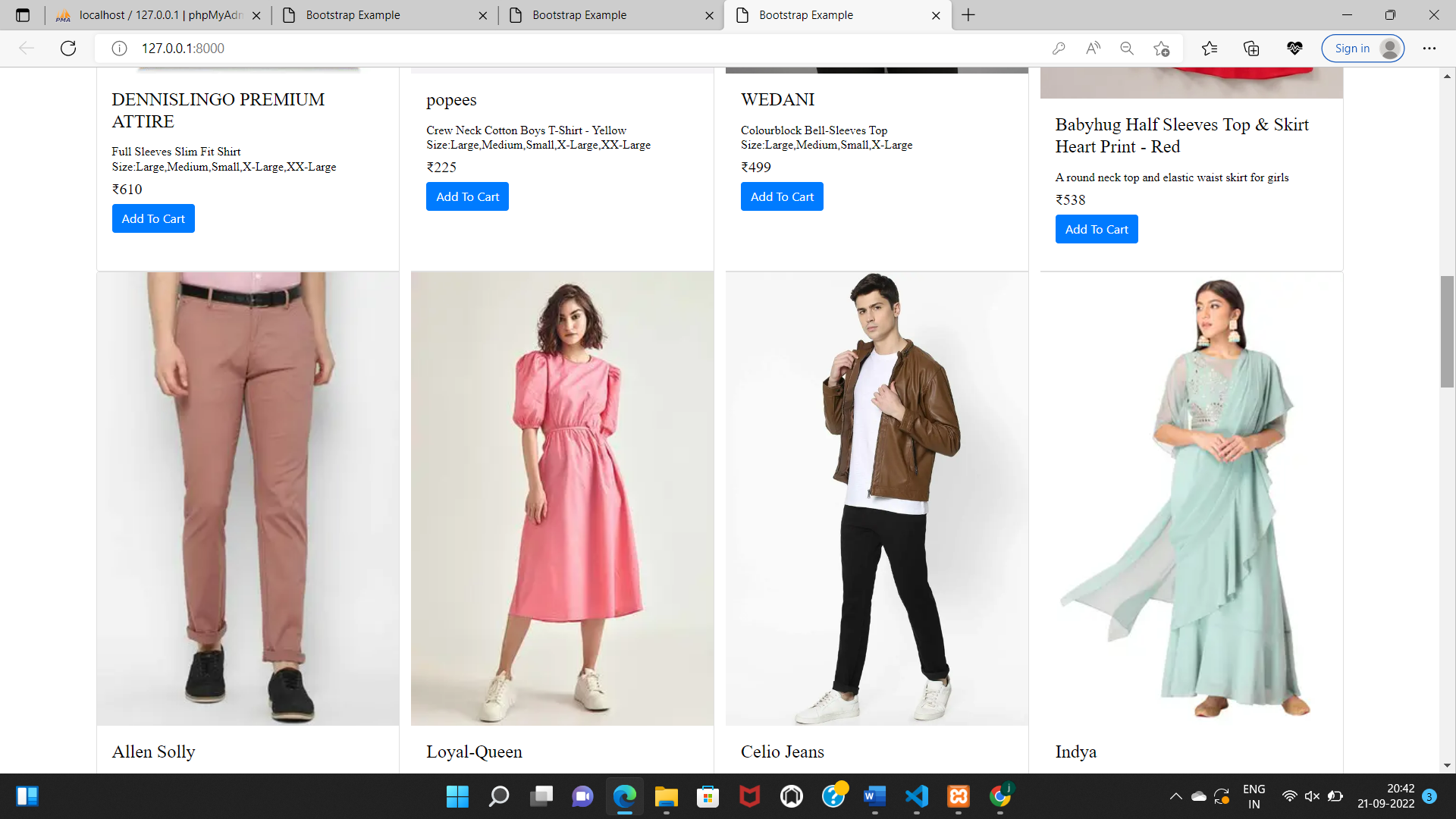
E-commerce Project

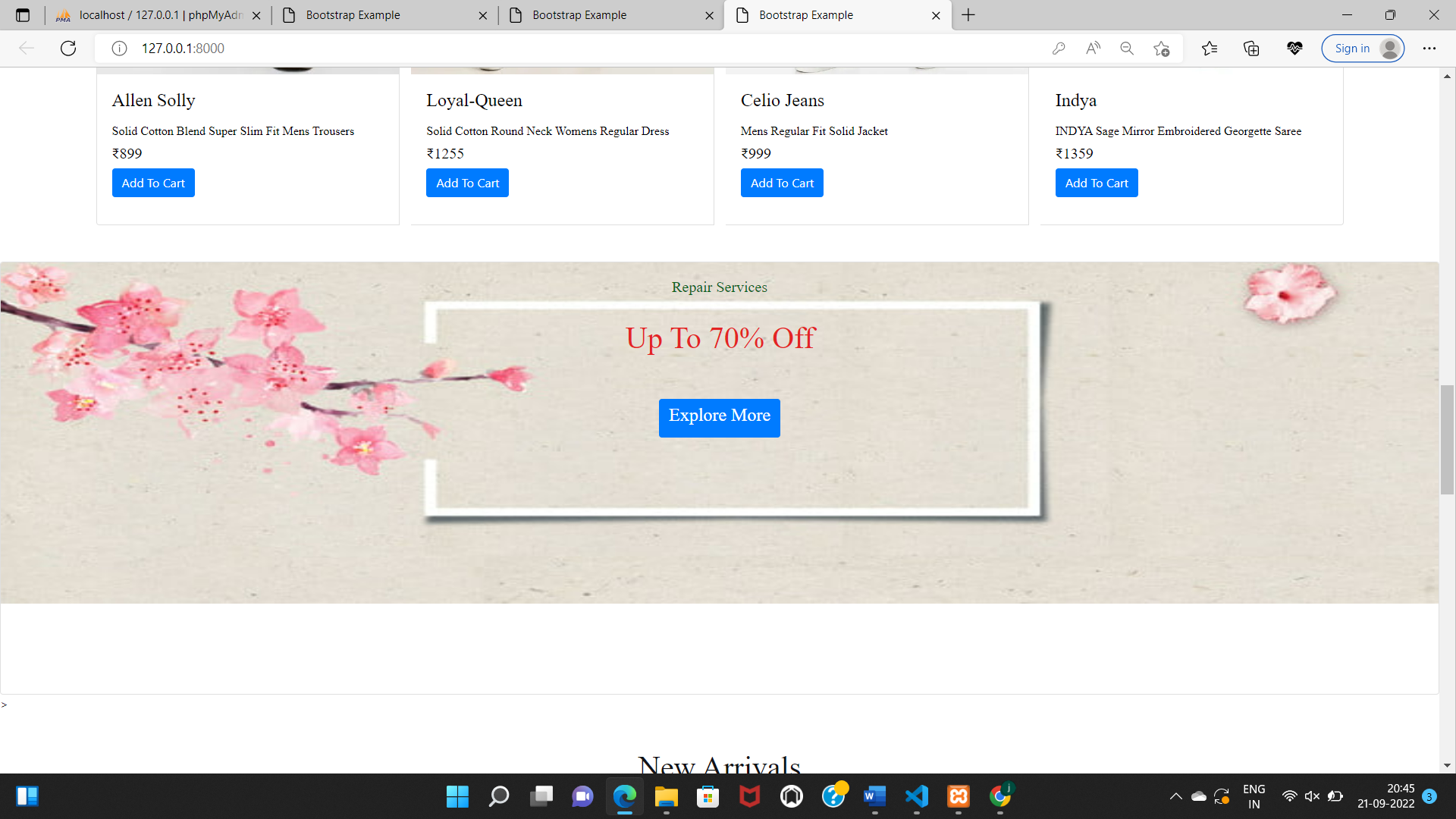
Shopping store

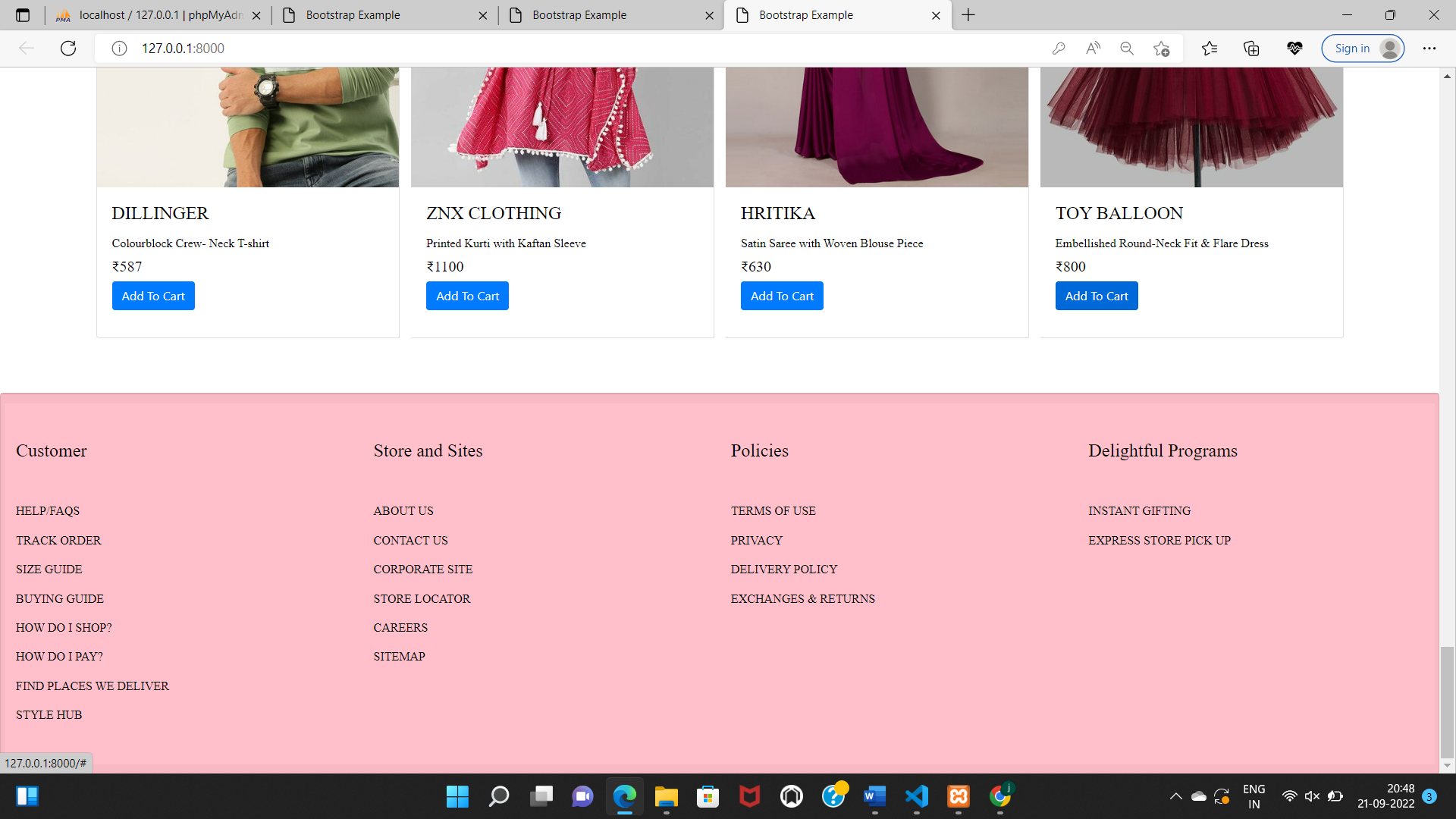
Homepage



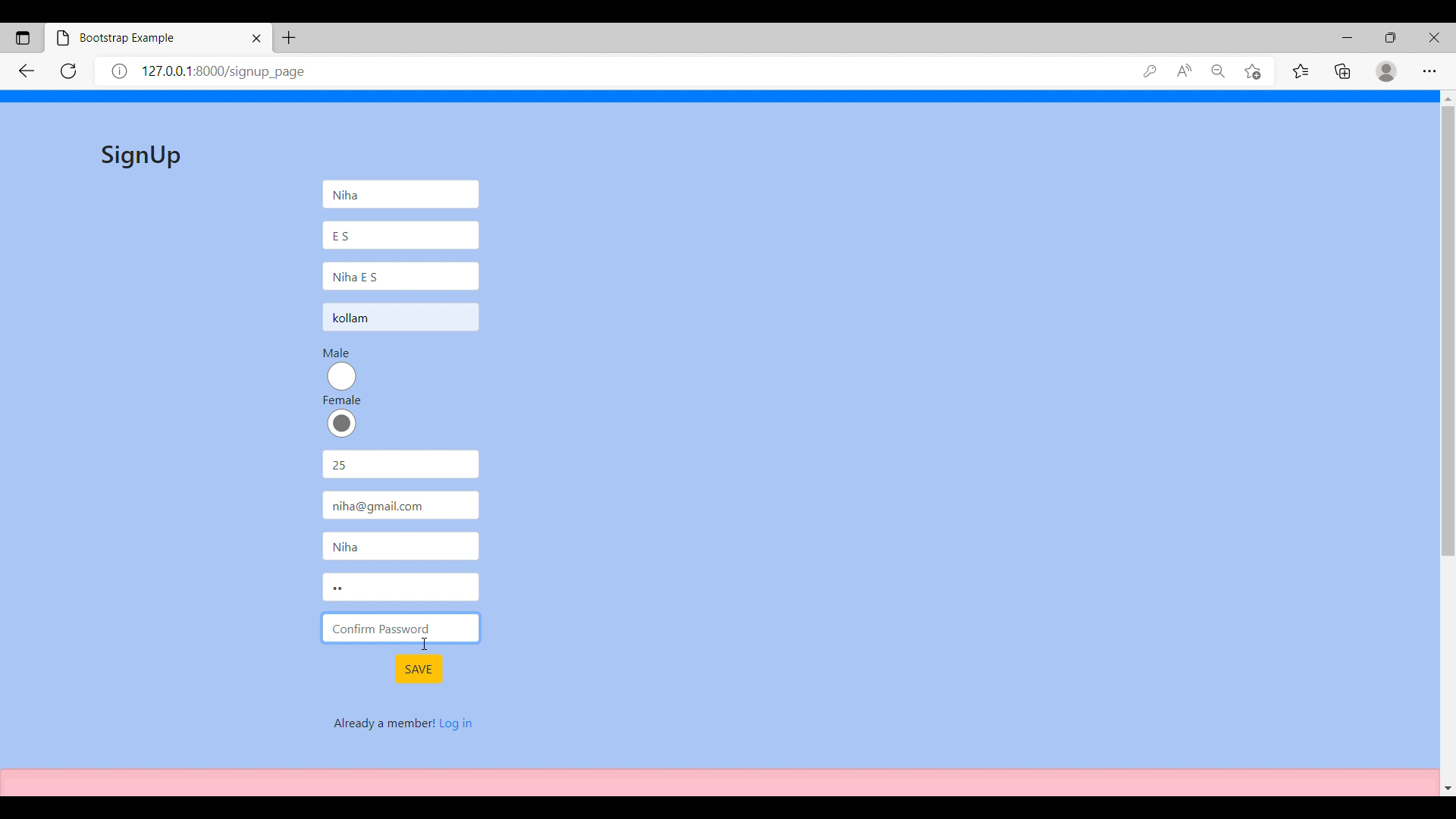




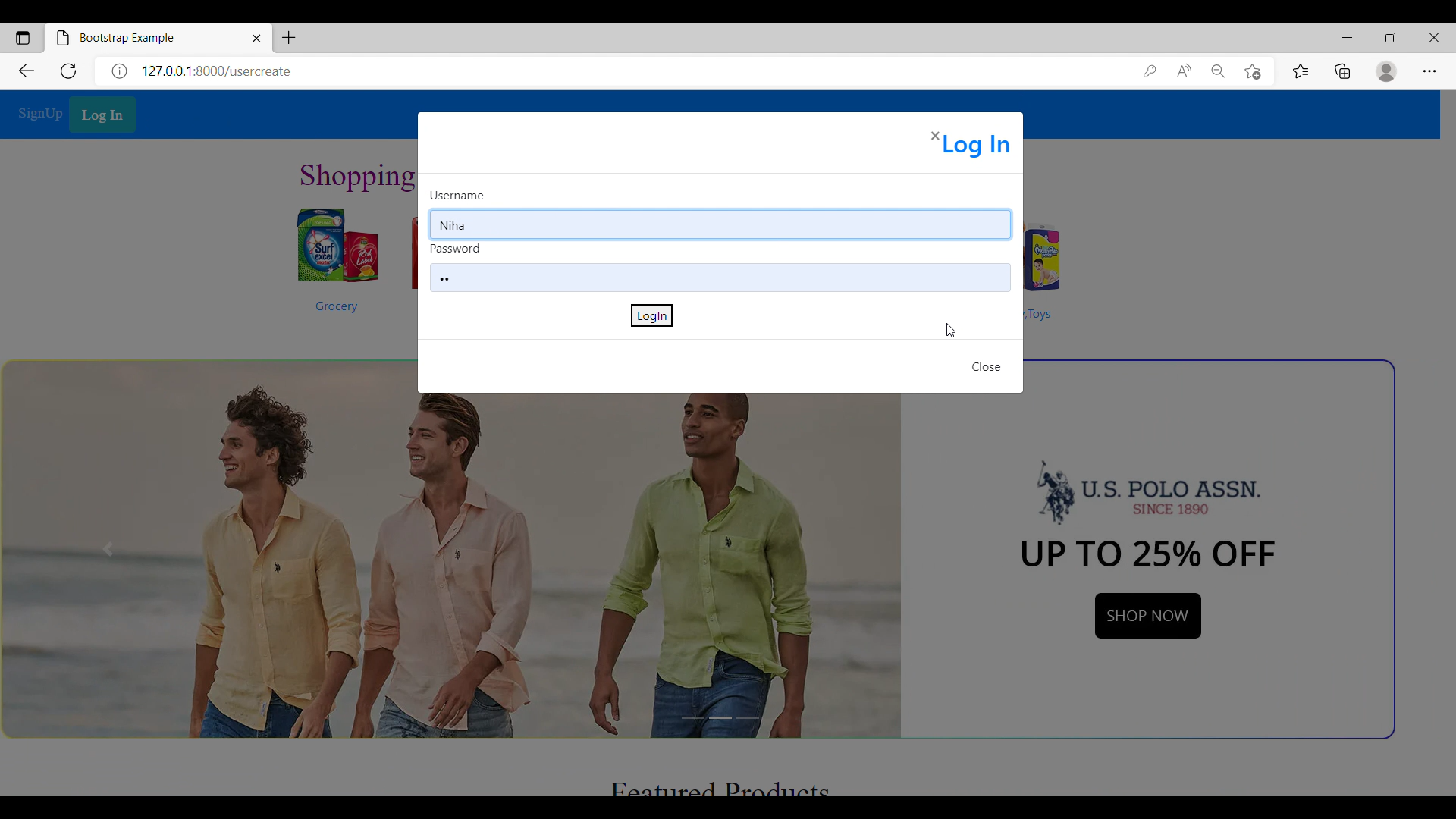




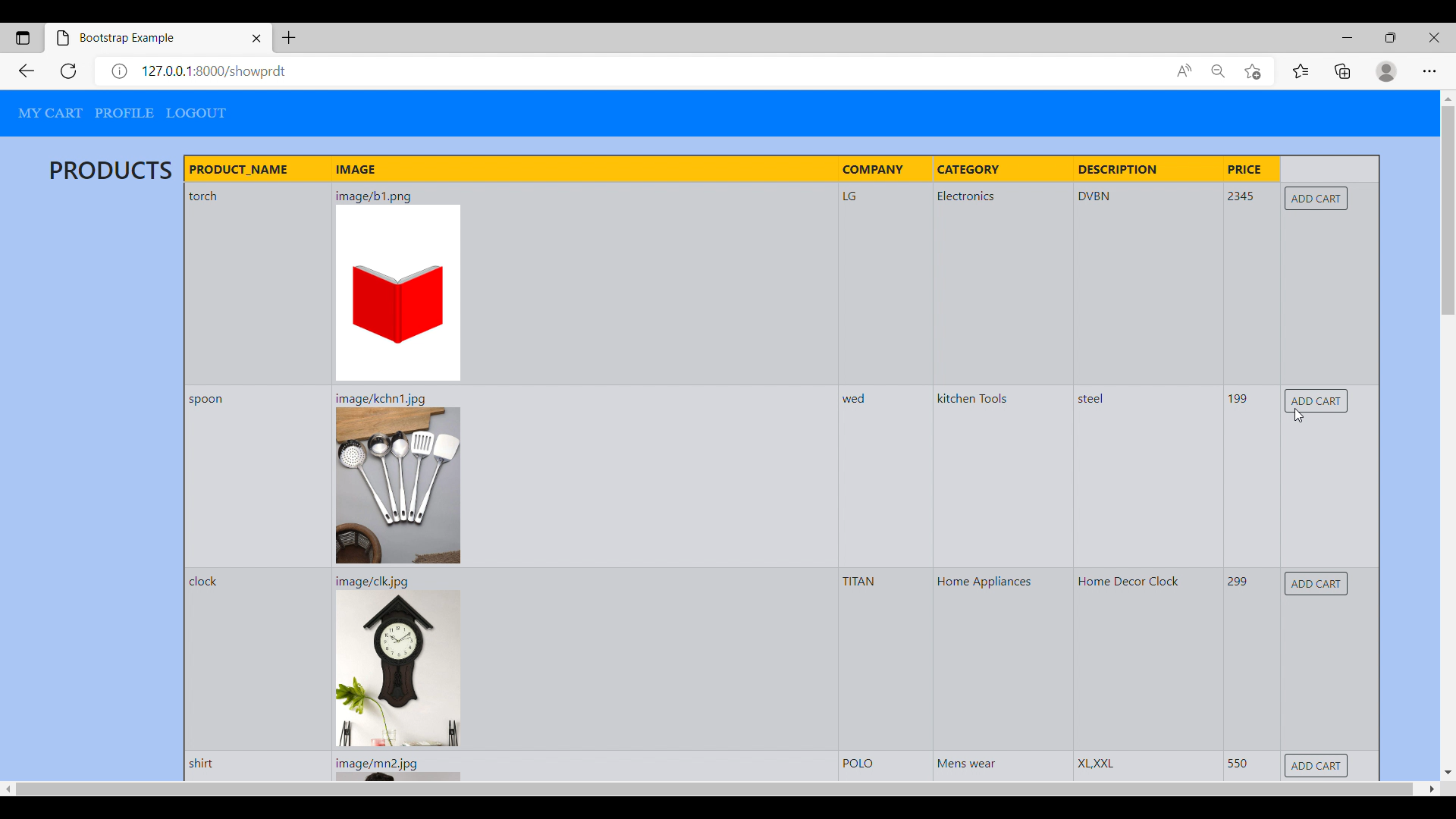
Signup form



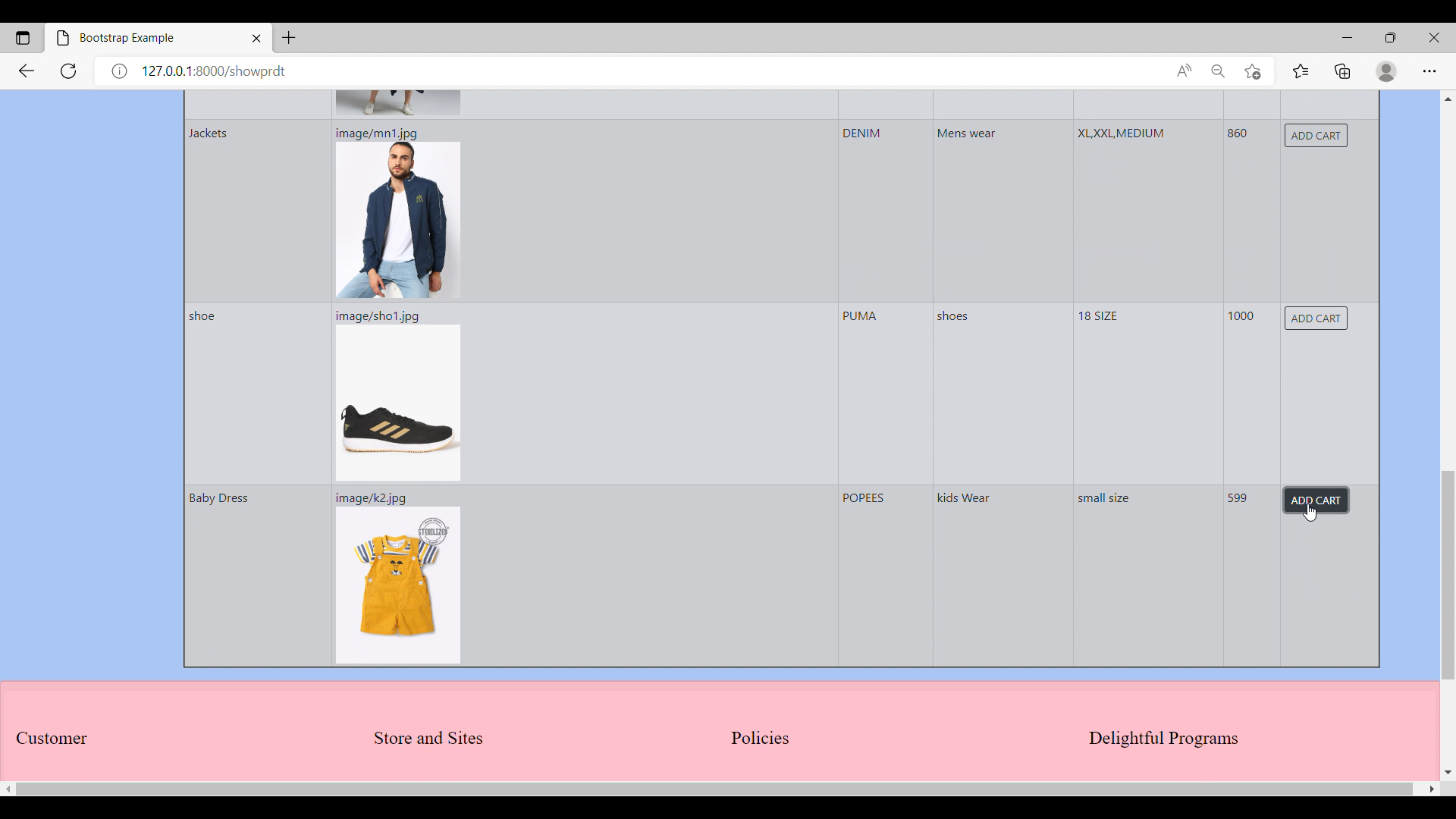
Login Form



Customer Home Page

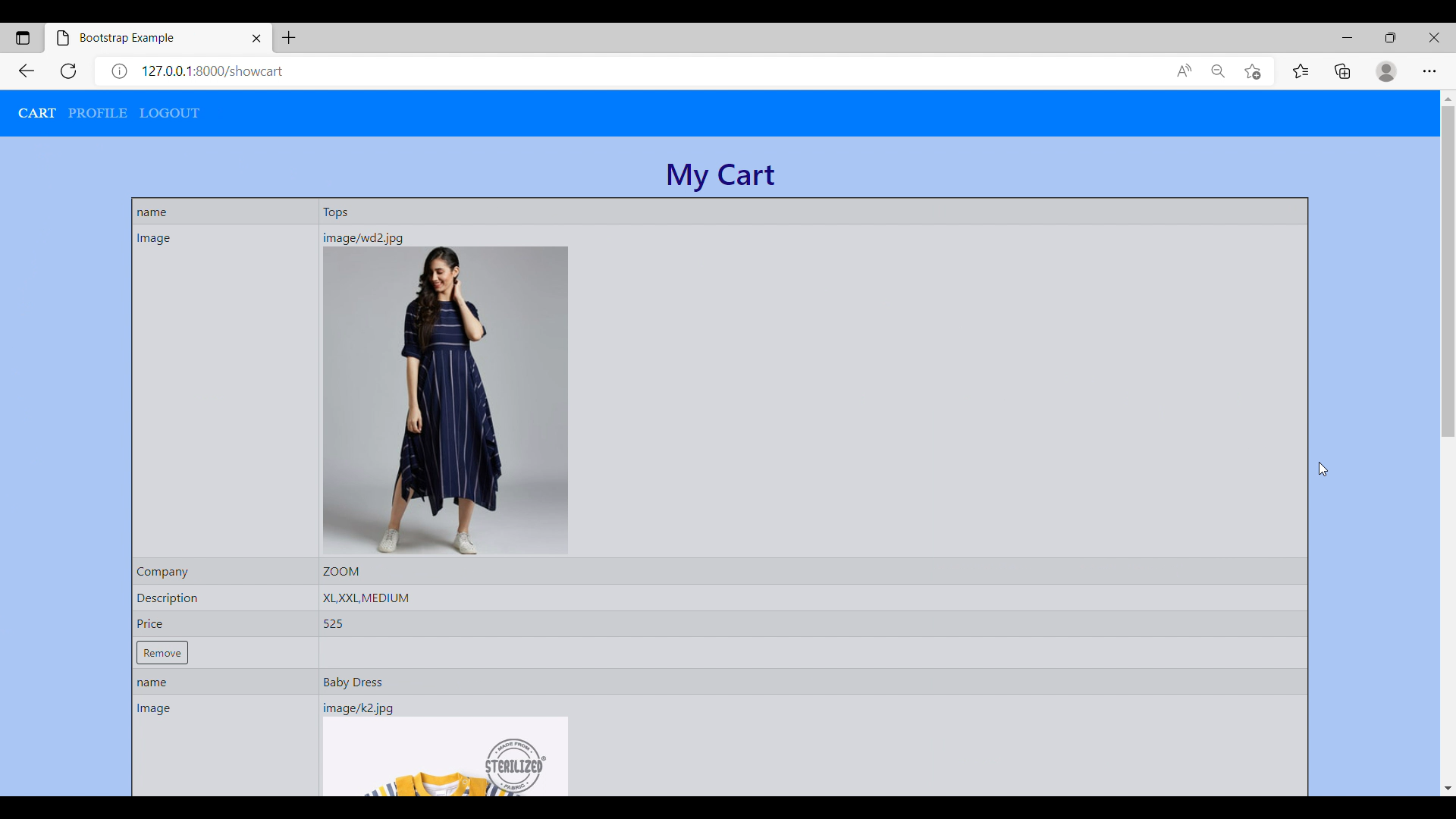


Product adding to cart

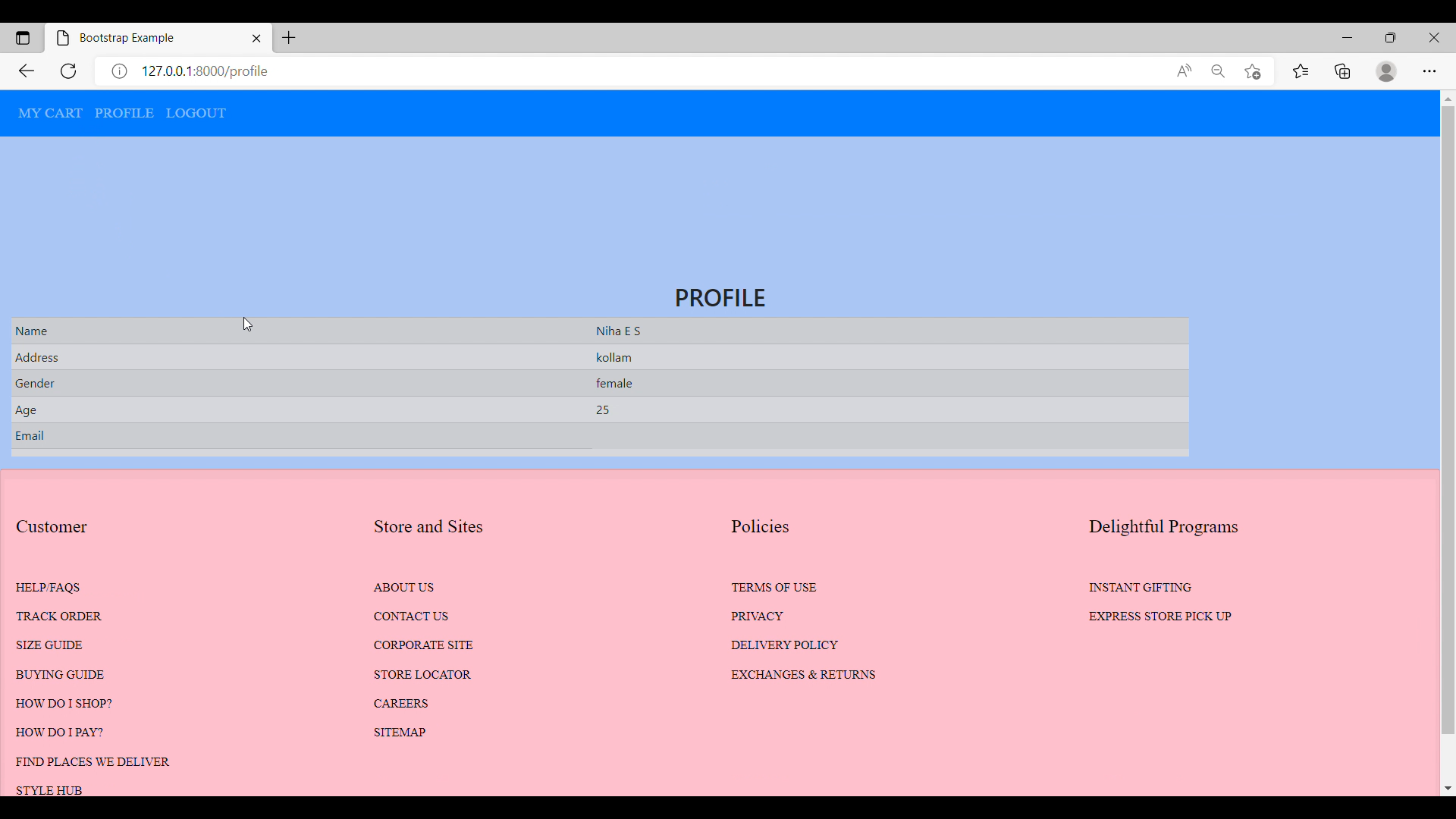


My cart page

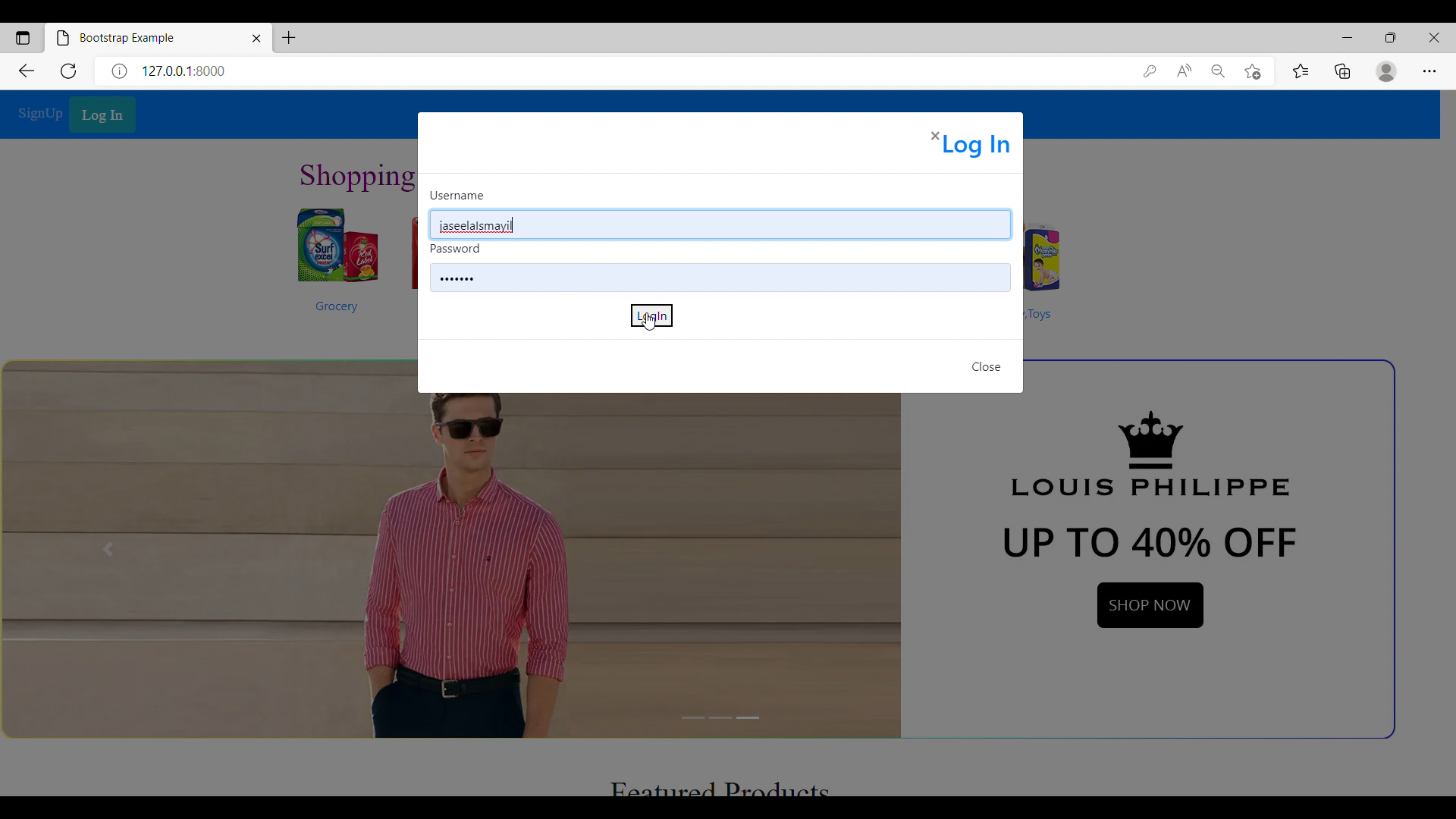
It contains items save by customer



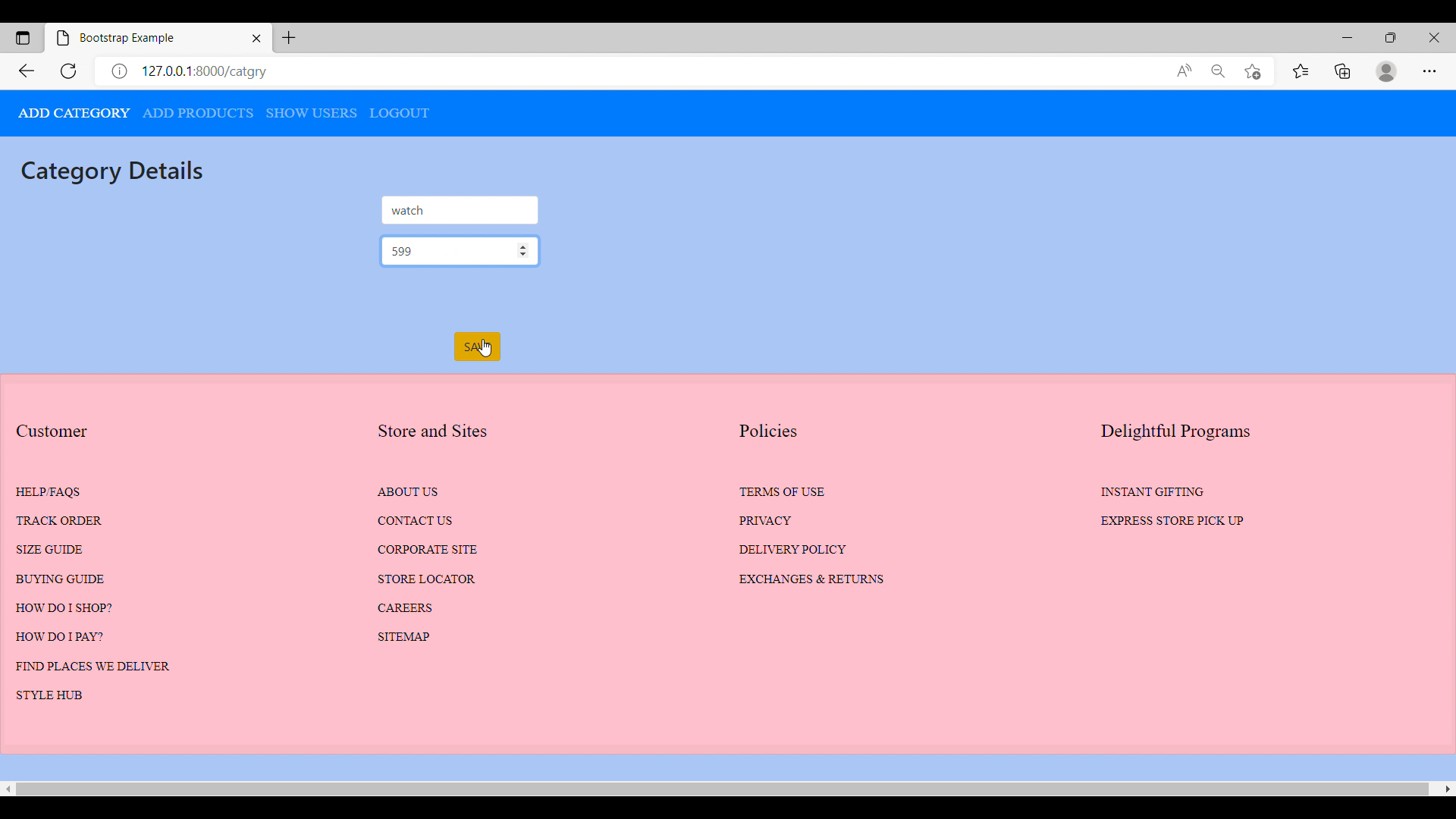
Customer profile viewing



Admin Login



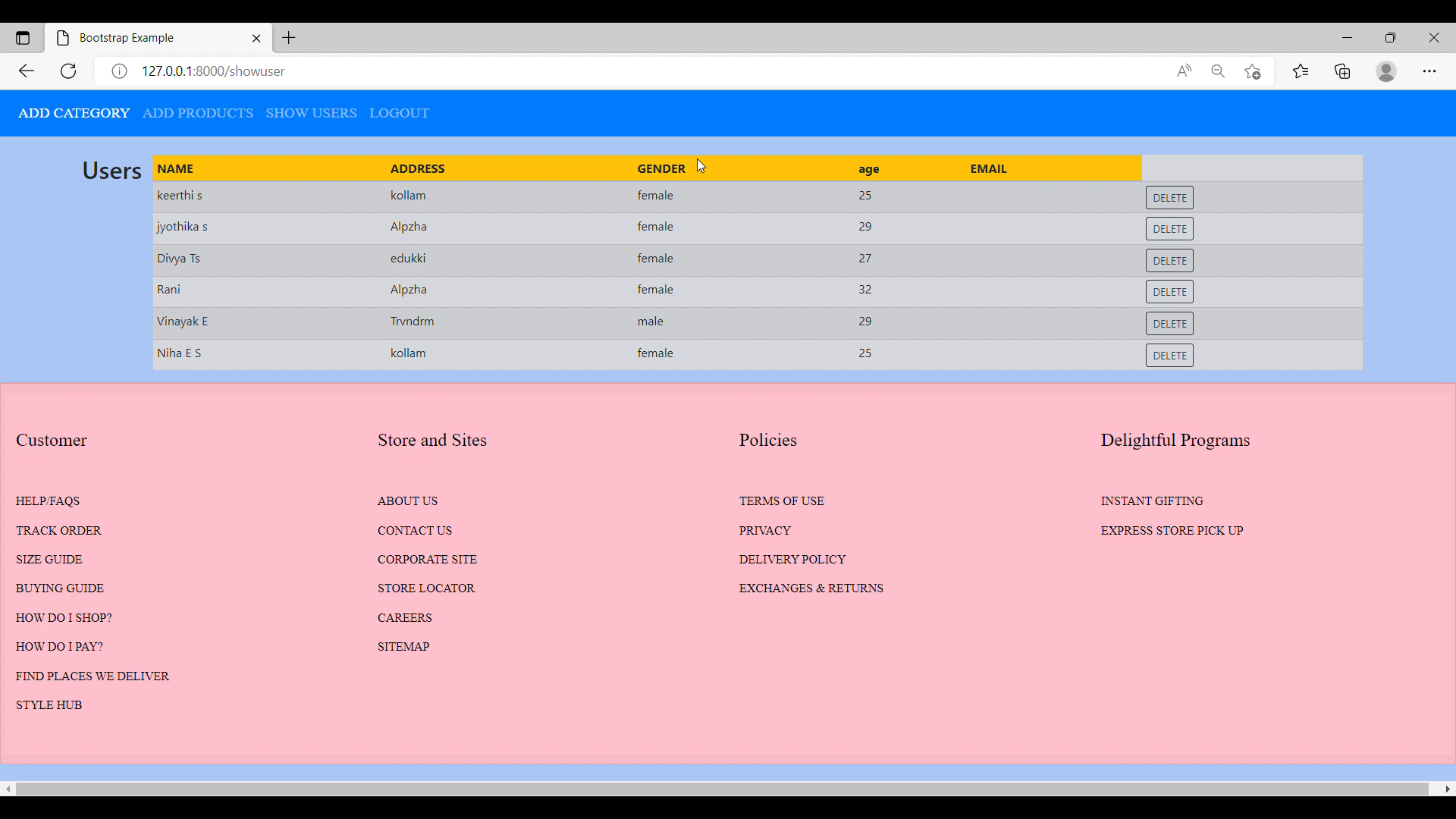
Admin adding Product Categories



Admin adding Product Details



Admin shows Customer Details



Views.py

from multiprocessing import context

from pydoc import describe

from django.contrib import messages

from django.shortcuts import render,redirect

from django.contrib.auth.models import User,auth

from shoppingApp.models import SCategory,SPrdts,SUserMember,SCart

from django.contrib.auth import authenticate,login,logout

from django.contrib.auth.decorators import login\_required

# Create your views here.

def  home(request):

    return render(request,'home.html')

def adminhome(request):

    return render(request,'adminhome.html')

def userhome(request):

    return render(request,'userhome.html')

def signup\_page(request):

    return render(request,'signup.html')

def usercreate(request):

    if request.method =='POST':

        firstname=request.POST['firstname']

        lastname=request.POST['lastname']

        fullname=request.POST['fullname']

        address=request.POST['address']

        gender=request.POST['gender']

        age=request.POST['age']

        email=request.POST['email']

        username=request.POST['username']

        password=request.POST['password']

        cpassword=request.POST['cpassword']

        if password==cpassword:

            if User.objects.filter(username=username).exists():

                messages.info(request,'This Username Already Exists!')

                return redirect('usercreate')

            else:

                user=User.objects.create\_user(

                    first\_name=firstname,

                    last\_name=lastname,

                    email=email,

                    username=username,

                    password=password)

                user.save()

                ds=User.objects.get(id= user.id)

                udt=SUserMember(

                    fullname=fullname,

                    address=address,

                    gender=gender,

                    age=age,

                    user=ds)

                udt.save()

        else:

            print('Password is not Matching..')

            return redirect('usercreate')

    return render(request,'home.html')

def SLogin(request):

    if request.method=='POST':

        username=request.POST['username']

        password=request.POST['password']

        user = auth.authenticate(username=username,password=password)

        if user is not None:

            if user.is\_staff:

                auth.login(request,user)

                return redirect('adminhome')#adminhome

            else:

                auth.login(request,user)

                return redirect('showprdt')#index

        else:

            return redirect('SLogin')

    else:

        return redirect('home')#admin home

@login\_required(login\_url='SLogin')

def SLogout(request):

    if request.user.is\_authenticated:

        auth.logout(request)

    return redirect('index')#

def catgry(request):

    return render(request,'addCatgry.html')

def addcatgry(request):

    if request.method== 'POST':

        catg=request.POST['category']

        sprice=request.POST['st\_price']

        cgr=SCategory(category=catg,st\_price=sprice)

        cgr.save()

        return redirect('adminhome')

def showcatg(request):

    if not request.user.is\_staff:

        return redirect('usercreate')

    cgrs=SCategory.objects.all()

    return render(request,'addCatgry.html',{'cgr':cgrs})

def productpage(request):

    categs=SCategory.objects.all()

    return render(request,'addPrdts.html',{'categs':categs})

def add\_prd(request):

    if request.method=='POST':

        pname=request.POST['pname']

        select=request.POST['category']

        category=SCategory.objects.get(id=select)

        company=request.POST['company']

        image=request.FILES.get('image')

        desc=request.POST['desc']

        price=request.POST['price']

        prd=SPrdts(pname=pname,category=category,company=company,image=image,desc=desc,price=price)

        prd.save()

        return redirect('adminhome')

def showuser(request):

    if not request.user.is\_staff:

        return redirect('usercreate')

    suser=SUserMember.objects.all()

    return render(request,'showUser.html',{'suser':suser})

def delUser(request,id):

    if not request.user.is\_staff:

        return redirect('usercreate')

    suser=SUserMember.objects.get(id=id)

    suser.delete()

    return redirect('showuser')

def profile(request):

    s=SUserMember.objects.get(user=request.user)

    return render (request,'profile.html',{'s':s})

def cat(request):

    return render(request,'viewCart.html')

@login\_required(login\_url='SLogin')

def add\_cart(request,id):

    pnames=SPrdts.objects.get(id=id)

    dt=SCart.objects.create(pname=pnames,user=request.user)

    dt.save()

    return redirect('showprdt')

@login\_required(login\_url='SLogin')

def showcart(request):

    pnames=SCart.objects.filter(user=request.user)

    return render(request,'viewCart.html',{'pnames':pnames})

def removeItem(request,id):

    if  request.user.is\_staff:

        return redirect('usercreate')

    scart=SCart.objects.get(id=id)

    scart.delete()

    return redirect('showcart')

def showprdt(request):

     pname=SPrdts.objects.all()

     return render(request,'viewPrdts.html',{'pname':pname})

Models.py

from unicodedata import category

from django.db import models

from django.contrib.auth.models import User

class SCategory(models.Model):

    category=models.TextField()

    st\_price=models.IntegerField()

    def \_\_str\_\_(self):

        return self.category

class SPrdts(models.Model):

    category=models.ForeignKey(SCategory,on\_delete=models.CASCADE,null=True)

    pname=models.TextField()

    company=models.TextField()

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

    desc=models.TextField()

    price=models.IntegerField()

    def \_\_str\_\_(self):

        return self.category.category

class SUserMember(models.Model):

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

    fullname=models.TextField()

    address=models.TextField()

    gender=models.TextField()

    age=models.IntegerField()

    email=models.TextField()

    def \_\_str\_\_(self):

        return self.user.username

class SCart(models.Model):

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

    pname=models.ForeignKey(SPrdts,on\_delete=models.CASCADE,null=True)