


E-commerce Project


Shopping store


Homepage


[Sign In](#) [Log In](#)


Shopping Store



Grocery



Mobile



Men/Women Fashion


TV/Electronics


HOME


Appliances


Beauty Toys




JUNIPER


UP TO 75% OFF

[SHOP NOW](#)


Featured Products




DENNISLINGO PREMIUM ATTIRE
Full Sleeve Men Tn Shirt
Size: Large, Medium, Small, X-Large, XX-Large
₹610
[Add To Cart](#)




popes
Crew Neck Cotton Boys T-Shirt - Yellow
Size: Large, Medium, Small, X-Large, XX-Large
₹225
[Add To Cart](#)




WEDANI
Colorblock Bell Sleeve Top
Size: Large, Medium, Small, X-Large
₹499
[Add To Cart](#)




Babying Half Sleeves Top & Skirt Heart Print - Red
A round neck top and elastic waist skirt for girls
₹538
[Add To Cart](#)




Allen Solly
Solid Cotton Blend Super Slim Fit Men Trousers
₹999
[Add To Cart](#)



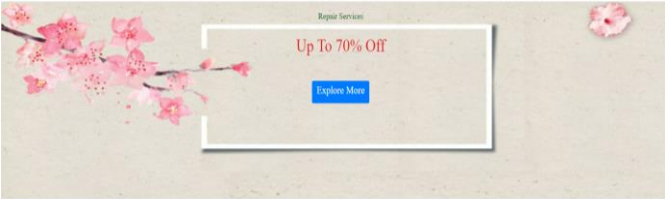
Loyal-Queen
Solid Cotton Round Neck Women Regalia Dress
₹1255
[Add To Cart](#)



Celio Jeans
Men Regular Fit Solid Jacket
₹999
[Add To Cart](#)



Indya
DZ0705 Sage Green Embroidered Georgette Saree
₹1,159
[Add To Cart](#)



Register Services
Up To 70% Off
[Explore More](#)

Customer

- [HELP PAGE](#)
- [TRACK ORDER](#)
- [SIZE GUIDE](#)
- [BUYING GUIDE](#)
- [HOW DO I SHOP?](#)
- [HOW DO I PAY?](#)
- [FIND PLACES WE DELIVER](#)
- [STYLE BLOG](#)

Store and Sites

- [ABOUT US](#)
- [CONTACT US](#)
- [CORPORATE SITE](#)
- [STORE LOCATOR](#)
- [CAREERS](#)
- [SITEMAP](#)


Policies

- [TERMS OF USE](#)
- [PRIVACY](#)
- [DELIVERY POLICY](#)
- [EXCHANGES & RETURNS](#)

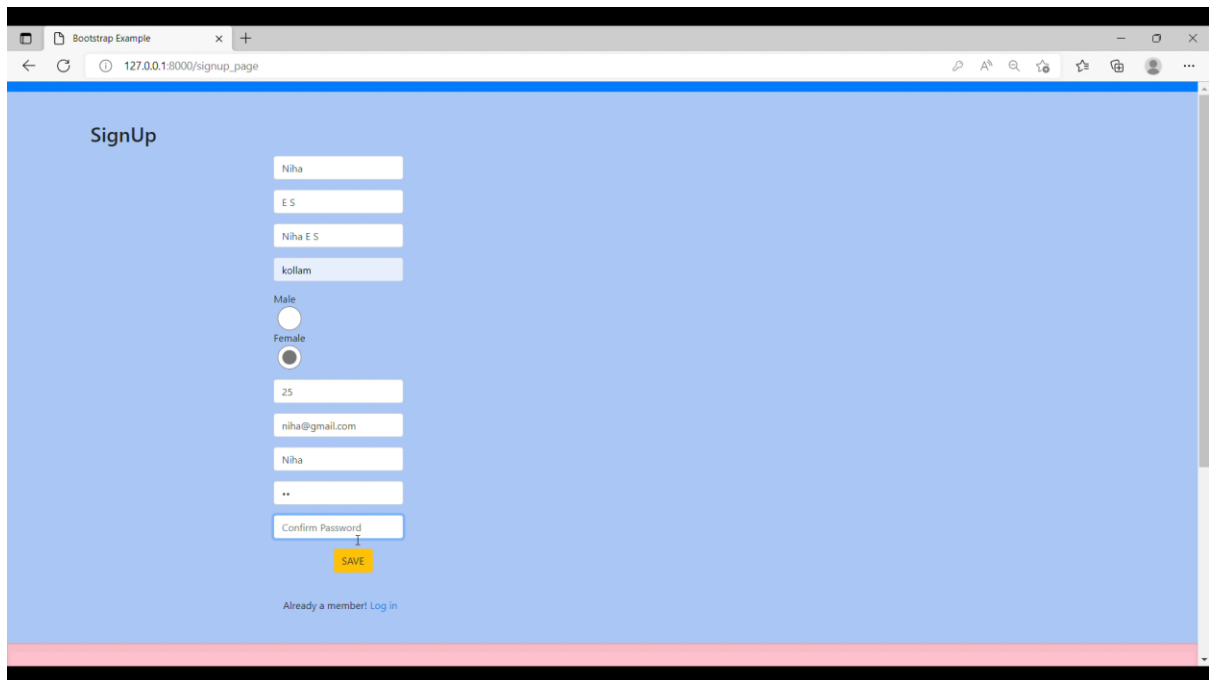
Delightful Programs

- [INSTANT GIFTING](#)
- [EXPRESS STORE PICK UP](#)

USDA FARMVIEW



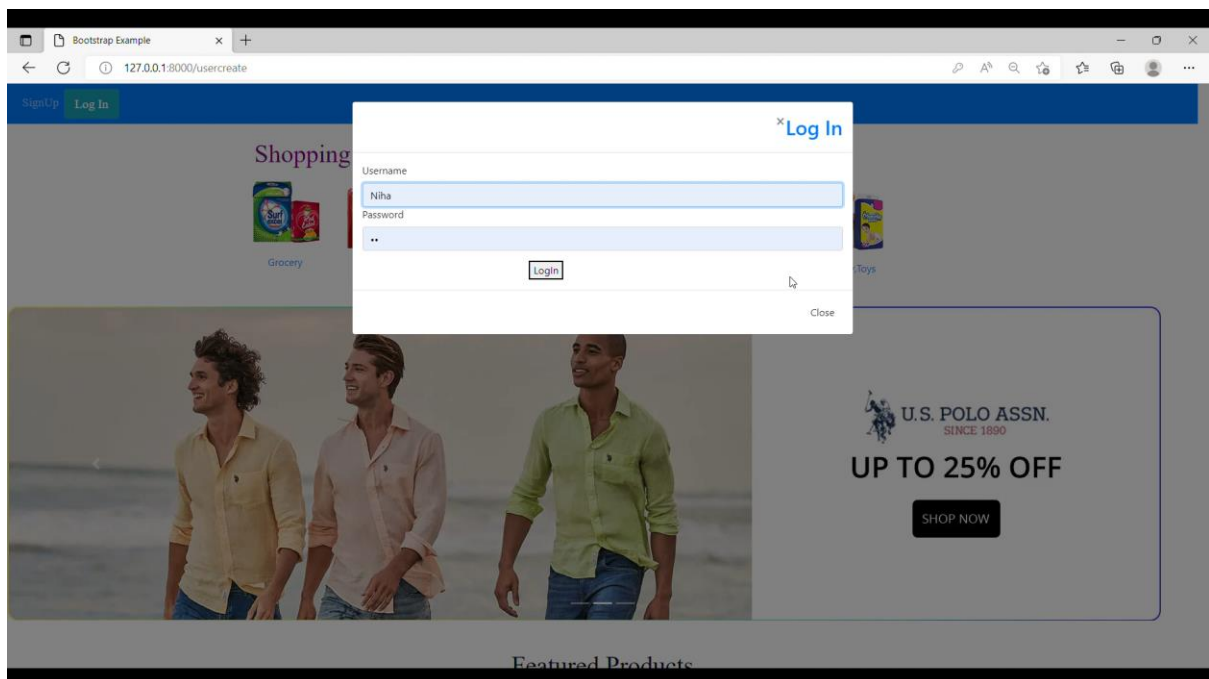
Signup form



A screenshot of a web browser displaying a Bootstrap Signup form. The browser's address bar shows the URL `127.0.0.1:8000/signup_page`. The form is titled "SignUp" and is set against a light blue background. It contains the following fields and elements:

- Text input: "Niha"
- Text input: "E S"
- Text input: "Niha E S"
- Text input: "kollam"
- Gender selection: "Male" (unselected) and "Female" (selected)
- Text input: "25"
- Text input: "niha@gmail.com"
- Text input: "Niha"
- Text input: "**" (password)
- Text input: "Confirm Password"
- A yellow "SAVE" button
- A link: "Already a member? Log in"

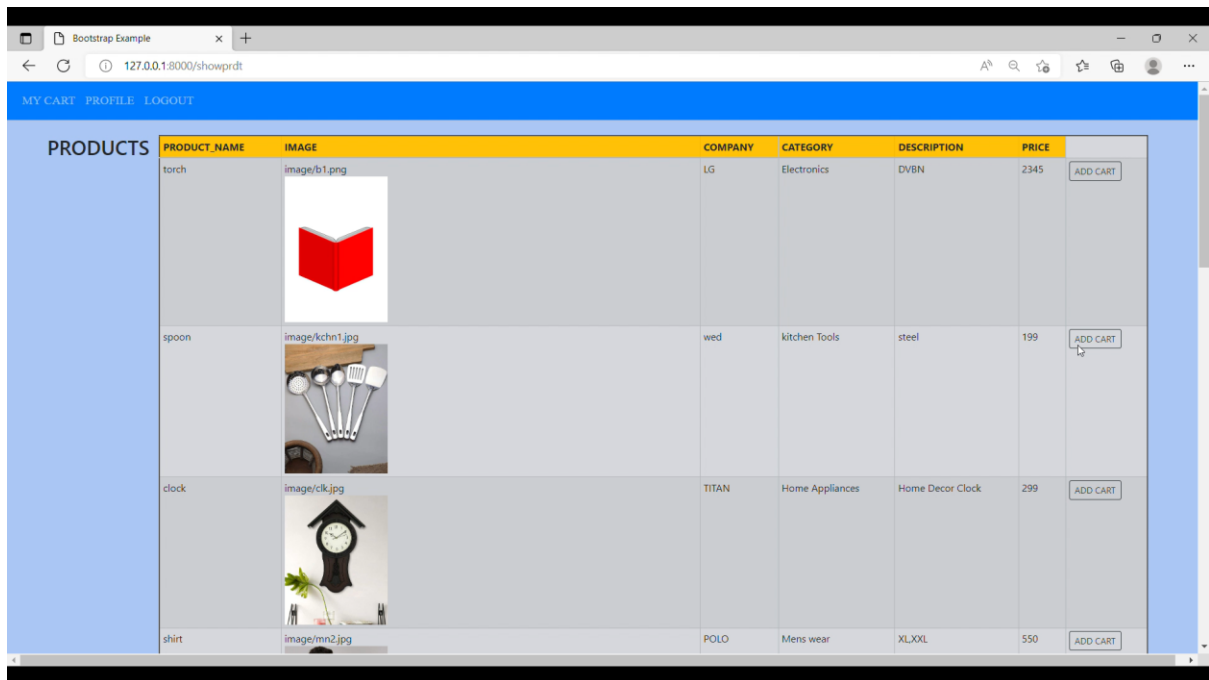
Login Form



A screenshot of a web browser displaying a Bootstrap Login form overlay on a shopping page. The browser's address bar shows the URL `127.0.0.1:8000/usercreate`. The background page features a "Shopping" section with "Grocery" and "Toys" categories, and a "Featured Products" section with a "U.S. POLO ASSN. SINCE 1890" advertisement offering "UP TO 25% OFF" and a "SHOP NOW" button. The Login form overlay, titled "Log In", contains the following fields and elements:

- Text input: "Niha" (Username)
- Text input: "**" (Password)
- A "Login" button
- A "Close" button

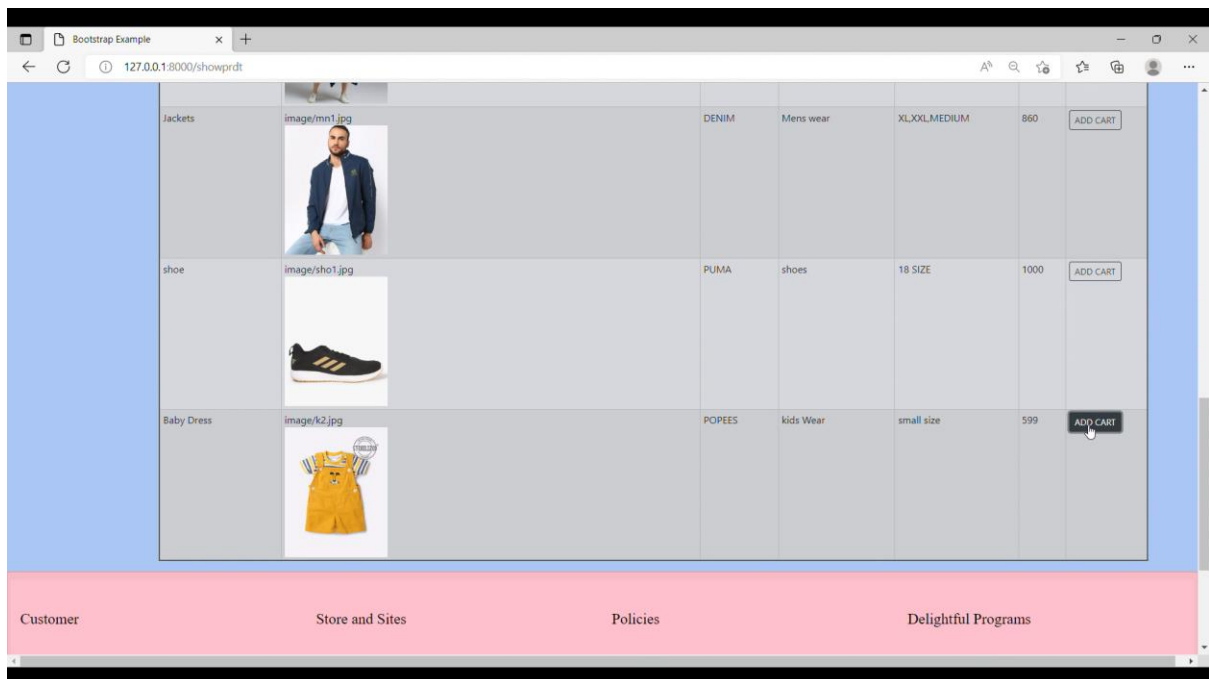
Customer Home Page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/showprdt". The page has a blue header with links for "MY CART", "PROFILE", and "LOGOUT". Below the header is a table titled "PRODUCTS" with the following columns: PRODUCT_NAME, IMAGE, COMPANY, CATEGORY, DESCRIPTION, PRICE, and an "ADD CART" button. The table contains four rows of product data.

PRODUCT_NAME	IMAGE	COMPANY	CATEGORY	DESCRIPTION	PRICE	ADD CART
torch	image/b1.png	LG	Electronics	DVBN	2345	ADD CART
spoon	image/kchn1.jpg	wed	kitchen Tools	steel	199	ADD CART
clock	image/clk.jpg	TITAN	Home Appliances	Home Decor Clock	299	ADD CART
shirt	image/mn2.jpg	POLO	Mens wear	XLXXL	550	ADD CART

Product adding to cart



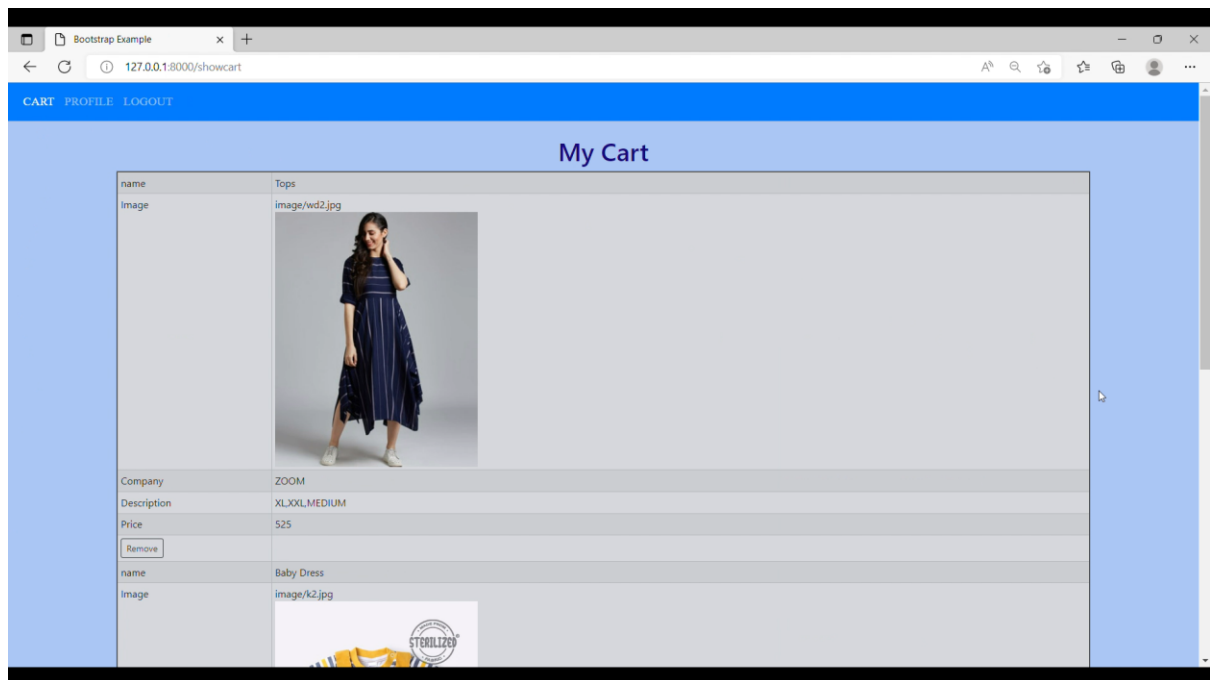
The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/showprdt". The page has a blue header with links for "MY CART", "PROFILE", and "LOGOUT". Below the header is a table titled "PRODUCTS" with the following columns: PRODUCT_NAME, IMAGE, COMPANY, CATEGORY, DESCRIPTION, PRICE, and an "ADD CART" button. The table contains three rows of product data. The "ADD CART" button for the "Baby Dress" row is highlighted with a mouse cursor.

PRODUCT_NAME	IMAGE	COMPANY	CATEGORY	DESCRIPTION	PRICE	ADD CART
Jackets	image/mn1.jpg	DENIM	Mens wear	XLXXL,MEDIUM	860	ADD CART
shoe	image/sho1.jpg	PUMA	shoes	18 SIZE	1000	ADD CART
Baby Dress	image/k2.jpg	POPEES	kids Wear	small size	599	ADD CART

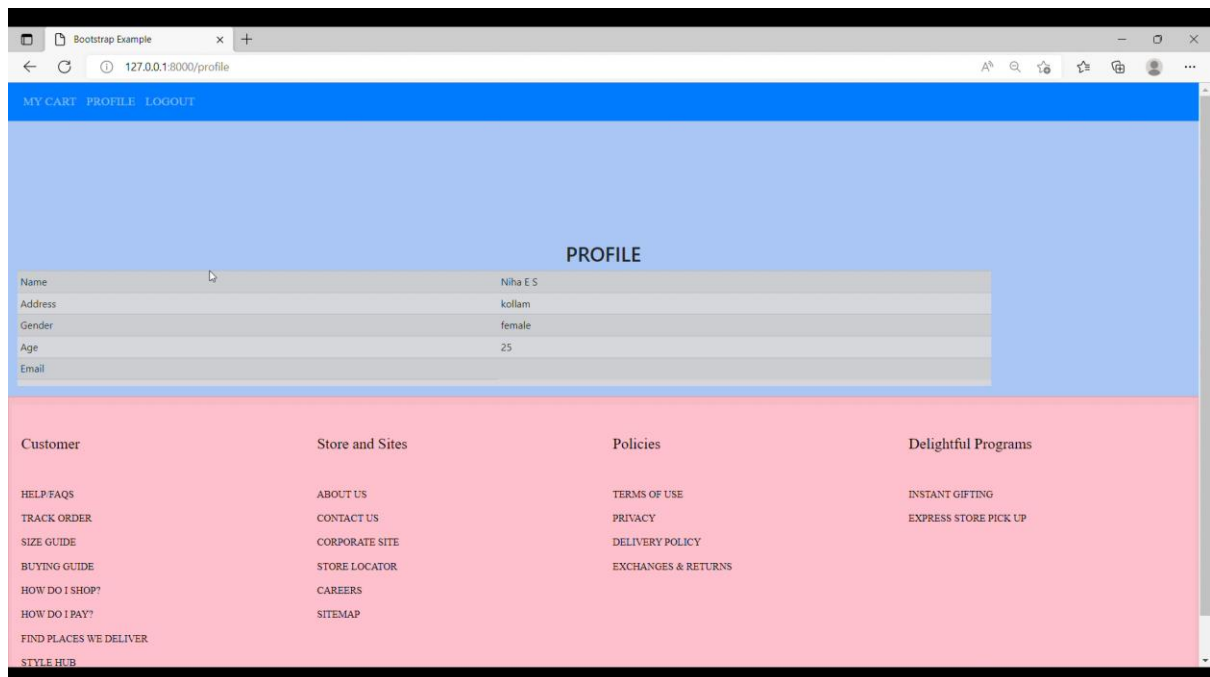
Below the table, there is a pink footer bar with the following text: "Customer", "Store and Sites", "Policies", and "Delightful Programs".

My cart page

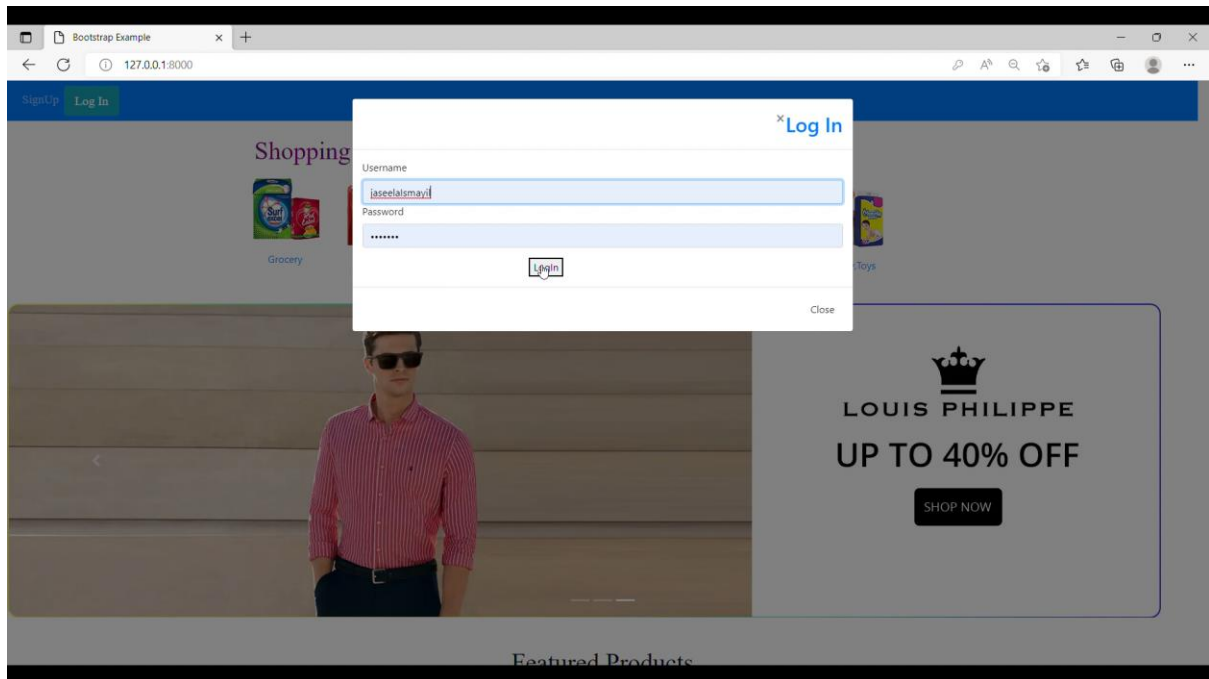
It contains items save by customer



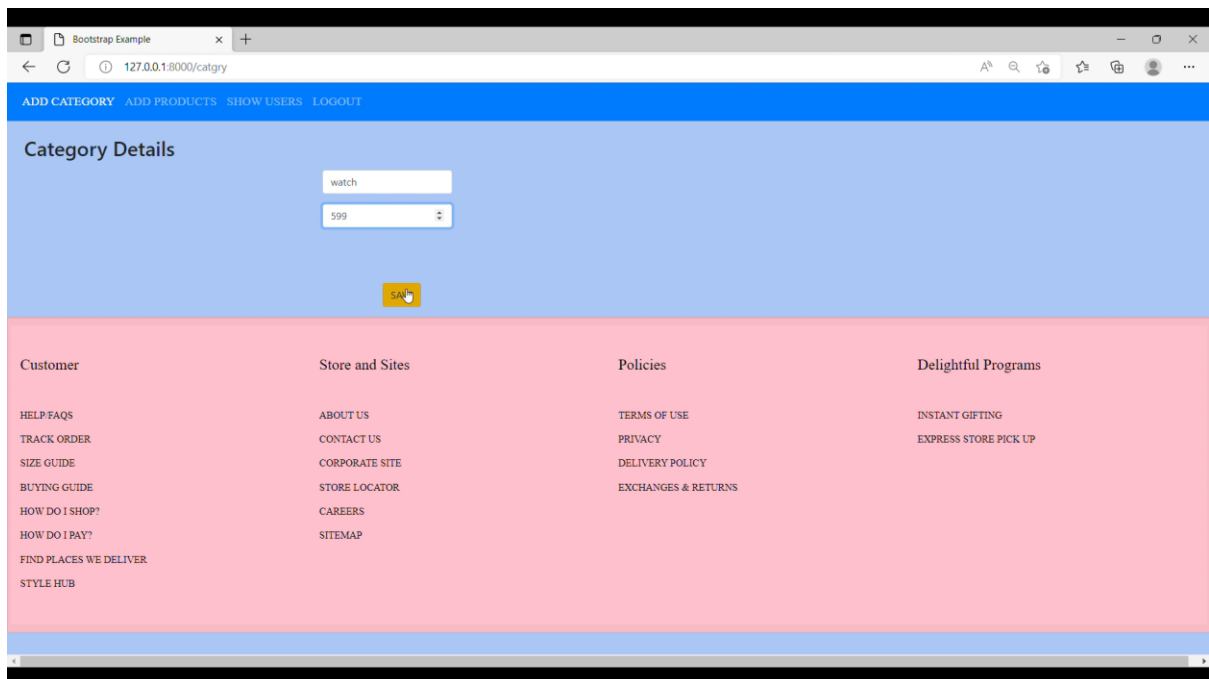
Customer profile viewing



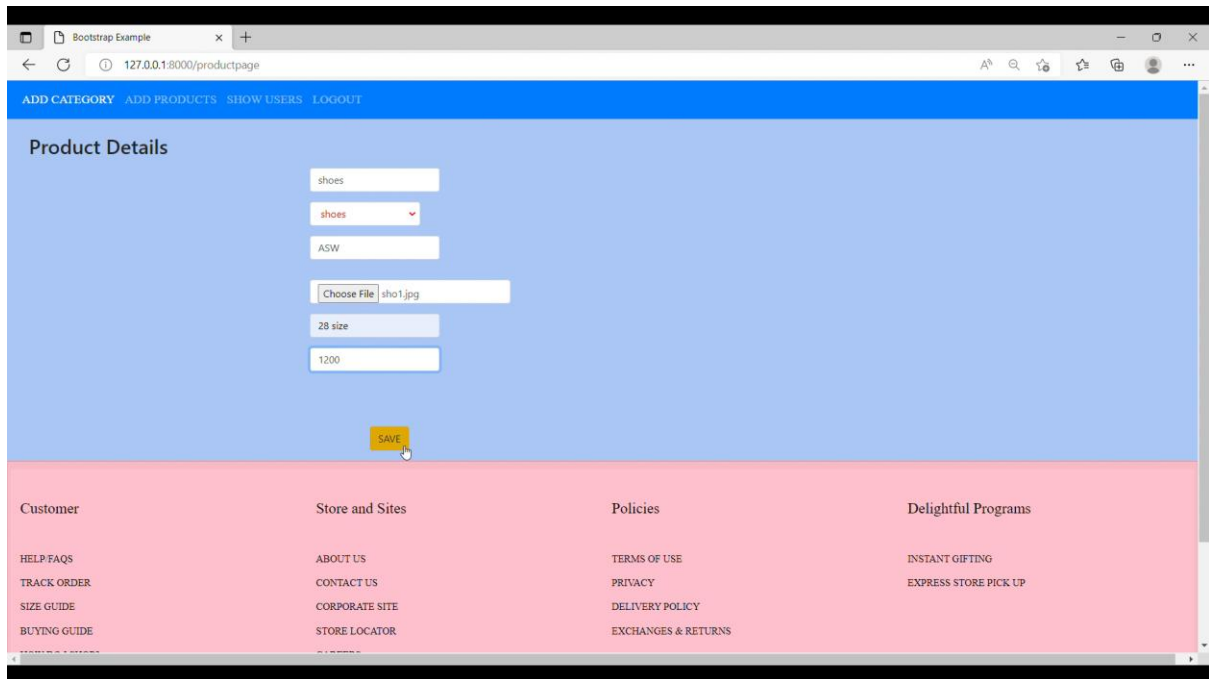
Admin Login



Admin adding Product Categories

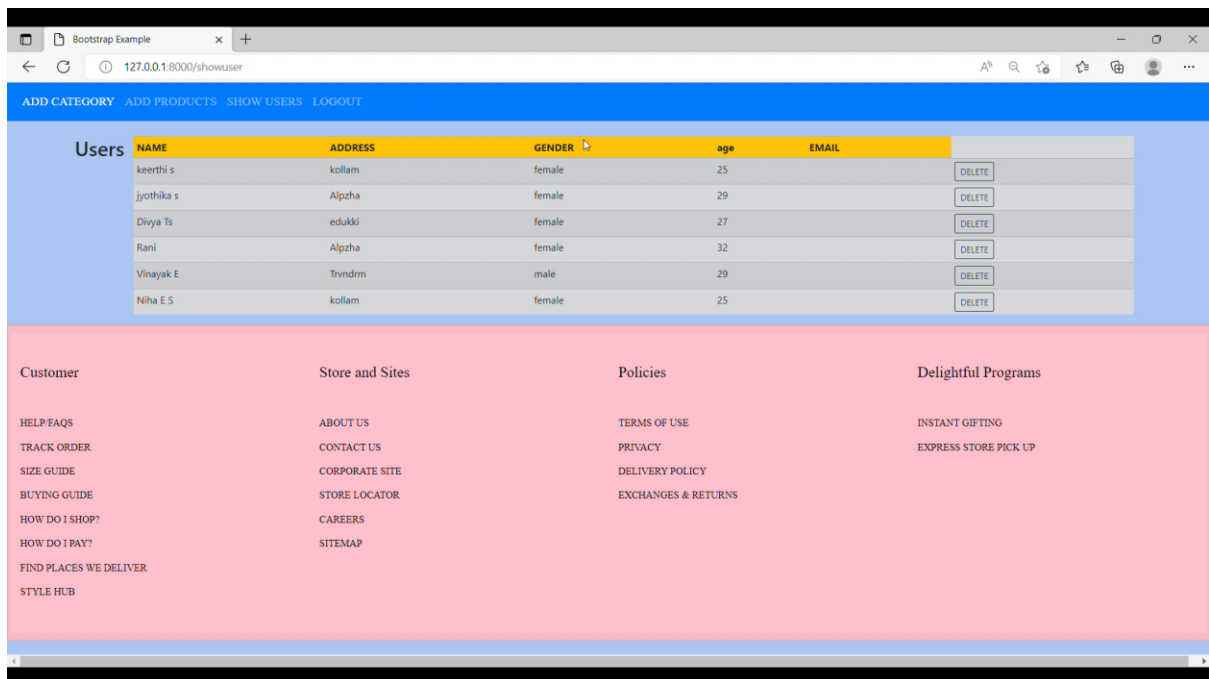


Admin adding Product Details



The screenshot shows a web browser window with the URL `127.0.0.1:8000/productpage`. The page has a blue header with navigation links: [ADD CATEGORY](#), [ADD PRODUCTS](#), [SHOW USERS](#), and [LOGOUT](#). The main content area is titled "Product Details" and contains a form with the following fields: a text input with "shoes", a dropdown menu with "shoes", a text input with "ASW", a file upload button "Choose File" next to "sho1.jpg", a text input with "28 size", and a text input with "1200". A yellow "SAVE" button is at the bottom of the form. Below the form is a pink footer with four columns of links: "Customer" (HELP-FAQS, TRACK ORDER, SIZE GUIDE, BUYING GUIDE), "Store and Sites" (ABOUT US, CONTACT US, CORPORATE SITE, STORE LOCATOR), "Policies" (TERMS OF USE, PRIVACY, DELIVERY POLICY, EXCHANGES & RETURNS), and "Delightful Programs" (INSTANT GIFTING, EXPRESS STORE PICK UP).

Admin shows Customer Details



The screenshot shows a web browser window with the URL `127.0.0.1:8000/showuser`. The page has a blue header with navigation links: [ADD CATEGORY](#), [ADD PRODUCTS](#), [SHOW USERS](#), and [LOGOUT](#). The main content area is titled "Users" and contains a table with the following data:

NAME	ADDRESS	GENDER	age	EMAIL	
keerthi s	kollam	female	25		<button>DELETE</button>
jyothika s	Alpza	female	29		<button>DELETE</button>
Divya Ts	edukki	female	27		<button>DELETE</button>
Rani	Alpza	female	32		<button>DELETE</button>
Vinayak E	Trvndrm	male	29		<button>DELETE</button>
Niha E S	kollam	female	25		<button>DELETE</button>

Below the table is a pink footer with four columns of links: "Customer" (HELP-FAQS, TRACK ORDER, SIZE GUIDE, BUYING GUIDE, HOW DO I SHOP?, HOW DO I PAY?, FIND PLACES WE DELIVER, STYLE HUB), "Store and Sites" (ABOUT US, CONTACT US, CORPORATE SITE, STORE LOCATOR, CAREERS, SITEMAP), "Policies" (TERMS OF USE, PRIVACY, DELIVERY POLICY, EXCHANGES & RETURNS), and "Delightful Programs" (INSTANT GIFTING, EXPRESS STORE PICK UP).

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,d
esc=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)
```

