**Code for Customer Registration:**

class CustomerRegistrationView(View):

def get(self, request):

form = CustomerRegistrationForm()

return render(request, 'app/customerregistration.html', {'form':form})

def post(self, request):

form = CustomerRegistrationForm(request.POST)

if form.is\_valid():

messages.success(request, "Congratulations!! Registered Successfully")

form.save()

return redirect("login")

return render(request, 'app/customerregistration.html', {'form':form})

**Code for Customer Login:**

class LoginForm(AuthenticationForm):

username = UsernameField(widget=forms.TextInput(attrs={'autofocus':True, 'class':'form-control'}))

password = forms.CharField(label=\_("Password"), strip=False,widget=forms.PasswordInput

(attrs={'autocomplete':'current-password', 'class':'form-control'}))

**Code for Product Detail View:**

class ProductDetailView(View):

def get(self,request,pk):

product = Product.objects.get(pk=pk)

totalitem = 0

item\_already\_in\_cart = False

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

item\_already\_in\_cart = Cart.objects.filter(Q(product=product.id)

& Q(user=request.user)).exists()

return render(request, "app/productdetail.html", {"product":product,

'item\_already\_in\_cart':item\_already\_in\_cart, 'totalitem':totalitem})

**Code for Designer Detail View:**

class DesignerDetailView(View):

def get(self,request,pk):

designer = Designer.objects.get(pk=pk)

already\_exist = DesignerPlaced.objects.filter(

user=request.user, designer=designer)

if already\_exist:

return redirect('designer\_orders')

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

return render(request, "app/designerdetail.html",

{"designer":designer, "totalitem":totalitem})

**Code for Show Cart:**

@login\_required

def show\_cart(request):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

user = request.user

carts = Cart.objects.filter(user=user)

amount = 0.0

shipping\_amount = 70

total\_amount = 0.0

cart\_product = [p for p in Cart.objects.all() if p.user == user]

if cart\_product:

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

amount+= tempamount

total\_amount = amount + shipping\_amount

return render(request, 'app/addtocart.html',

{'carts':carts, 'totalamount':total\_amount,

'amount':amount, 'totalitem':totalitem} )

else:

return render(request,'app/emptycart.html')

**Code for Orders:**

@login\_required

def orders(request):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

op = OrderPlaced.objects.filter(user=request.user)

if op:

return render(request, 'app/orders.html',

{'order\_placed':op, 'totalitem':totalitem})

else:

return render(request,'app/emptyorder.html')

**Code for Plus Cart:**

def plus\_cart(request):

if request.method == 'GET':

prod\_id = request.GET['prod\_id']

user = request.user

c = Cart.objects.get(Q(product=prod\_id)& Q(user=request.user))

c.quantity+=1

c.save()

amount = 0.0

shipping\_amount = 70

cart\_product = [p for p in Cart.objects.all() if p.user == user]

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

amount+= tempamount

data = {

'quantity': c.quantity,

'amount': amount,

'totalamount': amount + shipping\_amount

}

return JsonResponse(data)

**Code for Minus Cart:**

def minus\_cart(request):

if request.method == 'GET':

prod\_id = request.GET['prod\_id']

user = request.user

c = Cart.objects.get(Q(product=prod\_id)& Q(user=request.user))

c.quantity-=1

c.save()

amount = 0.0

shipping\_amount = 70

cart\_product = [p for p in Cart.objects.all() if p.user == user]

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

amount+= tempamount

data = {

'quantity': c.quantity,

'amount': amount,

'totalamount': amount + shipping\_amount

}

return JsonResponse(data)

**Code for Delete Item:**

def delete\_item(request):

if request.method == 'GET':

prod\_id = request.GET['prod\_id']

c = OrderPlaced.objects.get(Q(product=prod\_id)& Q(user=request.user))

c.delete()

return redirect('orders')

**Code for Delete Profile:**

def delete\_profile(request):

if request.method == 'GET':

add\_id = request.GET['add\_id']

c = Customer.objects.get(Q(id=add\_id)& Q(user=request.user))

c.delete()

return redirect('address')

**Code for Remove Cart:**

def remove\_cart(request):

if request.method == 'GET':

prod\_id = request.GET['prod\_id']

user = request.user

c = Cart.objects.get(Q(product=prod\_id)& Q(user=request.user))

c.delete()

amount = 0.0

shipping\_amount = 70

cart\_product = [p for p in Cart.objects.all() if p.user == user]

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

amount+= tempamount

data = {

'amount': amount,

'totalamount': amount + shipping\_amount

}

return JsonResponse(data)

**Code for Profile View:**

@method\_decorator(login\_required, name='dispatch')

class ProfileView(View):

totalitem = 0

def get(self, request):

form = CustomerProfileForm()

totalitem = len(Cart.objects.filter(user=request.user))

return render(request, 'app/profile.html', {'form':form,

'active':'btn-primary', 'totalitem':totalitem})

def post(self, request):

form = CustomerProfileForm(request.POST)

if form.is\_valid():

usr = request.user

name = form.cleaned\_data['name']

phone = form.cleaned\_data['phone']

locality = form.cleaned\_data['locality']

city = form.cleaned\_data['city']

state = form.cleaned\_data['state']

zipcode = form.cleaned\_data['zipcode']

reg = Customer(user=usr, name=name, phone=phone,

locality=locality, city=city, state=state, zipcode=zipcode)

reg.save()

messages.success(request, 'Congratulations!!! Profile updated successfully')

form = CustomerProfileForm()

return render(request, 'app/profile.html', {'form':form, 'active':'btn-primary'})

**Code for Address:**

@login\_required

def address(request):

totalitem = 0

add = Customer.objects.filter(user=request.user)

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

return render(request, 'app/address.html', {'add':add, 'active':'btn-primary',

'totalitem':totalitem})

**Code for Checkout:**

@login\_required

def checkout(request):

totalitem = 0

user = request.user

add = Customer.objects.filter(user=user)

cart\_items = Cart.objects.filter(user=user)

amount = 0.0

shipping\_amount =70

totalamount = 0.0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

cart\_product = [p for p in Cart.objects.all() if p.user == user]

if add:

if cart\_product:

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

amount+= tempamount

totalamount = amount + shipping\_amount

dlr = int(totalamount/82.35)

return render(request, 'app/checkout.html', {'add':add, 'totalamount':totalamount,

'cart\_items':cart\_items, 'dlr':dlr, 'totalitem':totalitem})

else:

return redirect('profile')

**Code for Add to Cart:**

@login\_required

def add\_to\_cart(request):

user = request.user

product\_id = request.GET.get('prod\_id')

product = Product.objects.get(id=product\_id)

Cart(user=user, product=product).save()

return redirect('/cart')

**Code for Buynow Checkout:**

@login\_required

def buynow\_checkout(request):

user = request.user

product\_id = request.GET.get('prod\_id')

pro\_duct = Product.objects.get(id=product\_id)

BCart(user=user, product=pro\_duct).save()

add = Customer.objects.filter(user=user)

amount = 0.0

shipping\_amount =70

totalamount = 0.0

# cart\_product = [p for p in Product.objects.all() if p.user == user]

if add:

if pro\_duct:

amount = pro\_duct.discounted\_price

image = pro\_duct.product\_image

totalamount = amount + shipping\_amount

dlr = int(totalamount/82.35)

return render(request, 'app/buynow\_checkout.html', {'add':add,"amount":amount,

"image":image, 'totalamount':totalamount,

'pro\_duct':pro\_duct, 'dlr':dlr})

else:

return redirect('profile')

**Code for Payment Done:**

@login\_required

def payment\_done(request):

user = request.user

custid = request.GET.get('custid')

customer = Customer.objects.get(id=custid)

cart = Cart.objects.filter(user=user)

for c in cart:

OrderPlaced(user=user, customer=customer,

product=c.product, quantity=c.quantity).save()

c.delete()

return redirect("orders")

**Code for About, Contact and Help:**

@login\_required

def about(request):

return render(request, "app/about.html")

@login\_required

def contact(request):

return render(request, "app/contact\_page.html")

@login\_required

def help(request):

return render(request, "app/chatbot.html")

**Code for Adding Product:**

@login\_required

def mshirts(request, data=None):

if data == None:

mshirts = Product.objects.filter(category="MST")

elif data == "Nap\_Chief" or data == "Ben\_Martin" or data == "Park\_Avenue":

mshirts = Product.objects.filter(category="MST").filter(brand=data)

return render(request, 'app/mshirts.html', {"mshirts":mshirts})

**Code for Search:**

def search(request):

query = request.GET.get('query')

results = []

if query:

query1 = query.title()

results = Product.objects.filter(title=query1) or Product.objects.filter(brand=query1)

context = {

'query1': query1,

'results': results,

}

return render(request, 'app/search.html', context)

**Code for Product View:**

class ProductView(View):

def get(self, request):

totalitem = 0

blazers = Product.objects.filter(category="MB")

topwear = Product.objects.filter(category="TW")

boy = Product.objects.filter(category="B")

mshirt = Product.objects.filter(category="MST")

mshoes = Product.objects.filter(category="MS")

mhats = Product.objects.filter(category="MH")

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

return render(request,"app/home.html",{"blazers":blazers,

"topwear":topwear, "boy":boy, "mshirt":mshirt,

"mshoes":mshoes, "mhats":mhats, 'totalitem':totalitem })

**Code for Designer Login:**

def designer\_login(request):

if request.method == "POST":

username = request.POST.get("username")

password = request.POST.get("password")

designer\_code = request.POST.get("designer\_code")

user = authenticate(request, username = username, password = password)

temp = Designer.objects.filter(designer\_code=designer\_code)

if user and temp:

login(request, user)

designer = Designer.objects.filter(username=request.user,

designer\_code=designer\_code)

if designer:

return redirect( 'designer\_home')

else:

messages.error(request,"Invalid Username or Password or Designer code")

else:

messages.error(request,"Invalid Username or Password or Designer code")

return render(request, 'app/designer\_login.html')

**Code for Designer Homepage:**

def designer\_home(request):

designer = Designer.objects.filter(username=request.user)

return render(request, 'app/designer\_home.html', {'designer':designer})

**Code for Designer Checkout:**

@login\_required

def designer\_checkout(request):

user = request.user

designer\_id = request.GET.get('designer\_id')

designer = Designer.objects.get(id=designer\_id)

add = Customer.objects.filter(user=user)

TempDesignerPlaced(user=user, designer=designer).save()

already\_exist = DesignerPlaced.objects.filter(user=user, designer=designer)

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

if add:

if designer:

return render(request, 'app/designer\_checkout.html',

{'add':add,'designer':designer, "totalitem":totalitem})

else:

return redirect('profile')

**Code for Payment done Designer:**

@login\_required

def payment\_done\_designer(request):

user = request.user

custid = request.GET.get('custid')

customer = Customer.objects.get(id=custid)

tdp = TempDesignerPlaced.objects.filter(user=user)

loop\_executed = False

for c in tdp:

if not loop\_executed:

DesignerPlaced(user=user, customer=customer,designer=c.designer).save()

TempDesignerPlaced.objects.all().delete()

loop\_executed = True

return redirect("designer\_orders")

**Code for Buynow Payment:**

def payment\_done\_buynow(request):

user = request.user

custid = request.GET.get('custid')

customer = Customer.objects.get(id=custid)

bcart = BCart.objects.filter(user=user)

loop\_executed = False

for c in bcart:

if not loop\_executed:

OrderPlaced(user=user, customer=customer,

product=c.product, quantity=c.quantity).save()

BCart.objects.all().delete()

loop\_executed = True

return redirect("orders")

**Code for Designer Orders:**

@login\_required

def designer\_orders(request):

op = DesignerPlaced.objects.filter(user=request.user)

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

if op:

return render(request, 'app/designer\_orders.html',

{'order\_placed':op, "totalitem":totalitem})

else:

return render(request,'app/emptyorder.html')

**Code for Delete Designer:**

def delete\_designer(request):

if request.method == 'GET':

des\_id = request.GET['des\_id']

c = DesignerPlaced.objects.get(Q(designer=des\_id)& Q(user=request.user))

c.delete()

return redirect('designer\_orders')

**Code for Designer Order View:**

@login\_required

def view\_orders\_designer(request):

designer = Designer.objects.get(username=request.user)

op = DesignerPlaced.objects.filter(designer=designer)

if op:

return render(request, 'app/view\_orders\_designer.html',

{'order\_placed':op})

else:

return render(request,'app/designer\_emptyorder.html')

**Code for Designer Status Update:**

def update\_status(request, designer\_placed\_id):

designer\_placed = get\_object\_or\_404(DesignerPlaced, id=designer\_placed\_id)

if request.method == 'POST':

form = StatusUpdateForm(request.POST)

if form.is\_valid():

designer\_placed.status = form.cleaned\_data['status']

designer\_placed.save()

return redirect('view\_orders\_designer')

else:

form = StatusUpdateForm(initial={'status': designer\_placed.status})

return render(request, 'app/update\_status.html', {'form': form})