Project Report

On

Online Shopping System

B.Tech (CE) SEM-IV

SEPP & SP-1

Vimal Matholiya - CE066 (19CEUOG120)
Gopal Malaviya - CE063 (19CEUES049)

Under the Guidance of

Prof. Brijesh Bhatt
Prof. Jigar Pandya
Prof. Pinkal Chauhan

Department Of Computer Engineering

Contents

1. Introduction

This project is for online shopping of electronics devices like Mobiles , TVs , Laptop , Headphone s , Tablets and Powerbanks. It supports many features which are provided by other shopping website.

Project is created using Django(Python) and other tools.

In this software user has many features like login , create account , reset password , view profile , search , cart , orders and logout.

User can create account in the software by filling a signup form. After successful registration user can login to the system by providing username and password. If by chance user forget his password he can reset his password , for that software sends email to the user for password reset link. User can view his profile and also can update it.

User can search any product and can add it into the cart. User can update quantity of product in cart and can remove any product from cart.

User can place order and view order history. If the order is not delivered then user can cancel that order.

At the end user can logout from system by logout button.

Admin can add or remove any product in the system and also can update product details. Admin can view all the orders and can update status of any order.

2. Software Requirement Specification

User of the System: Customers , Admins

R.1 Manage Users

R.1.1 Registration

Description: User register himself into the system by entering his required details.

Input: User details like name,bith-date,email-id,password,address and mobile number

Processing: New account is created for the user and unique id is given to the user to login.

Output: User gets id and password

R.1.2 Login

Description: User login to the system by entering valid id and password.

Input: User's id and password

Processing: Id and password are verified,if they are valid then access is given otherwise access is denied.

Output: User logged in to the system and home page opens

R.1.3 Update Profile

Description: User update his profile.

Input: User selection

Output: Display Confirmation message that Profile is updated.

R.2 Manage Items

R.2.1 Add item

Input: Details of item to be added

Output: Item is added

R.2.2 Update item
Input: Item to be updated
Output: Item is updated

R.2.3 Remove item
Input: Name of the item
Output: Item is removed

R.2.4 Make Item Out of Stock
Description: If item is not available then make it out of stock
Input: Item name
Output: Item is made out of Stock

R.2.5 Display Item count
Description: The total number of item in the system is displayed.
Input: User selection
Output: Count of items

R.2.6 Display Total Purchases
Description: The total number of purchases made in given time period is displayed.
Input: Time period
Output: Count of total purchases

R.3 View Item

R.4.1 Search Item
Description: Search the type of items Which the customer wants.
Input: The type of items which the customer wants
Output: The items that are matched with user's entered details are displayed

R.4.2 Filter
Description:filter all the items according to customer's need.
Input: The criteria on which the customer want to filter items
Output: Display a filtered list of items

R.4.4 Show Details of Items
Input: Name of the item
Output: Details of items are displayed

R.4 Purchase Item

R.4.1 Changes to Cart
Description: Customers add item into cart for later purchasing and also they can remove item from cart.

R.4.1.1 Add Item to the Cart
Input: Selected item by customer.
Output: Item is added to the cart.

R.4.1.2 Remove Item from the Cart
Input: Selected item by customer.
Output: Item is removed from the cart.

R.4.2 Buy
Description: The customer buy item from the cart or directly.
Input: User selection
Output: The total amount to be paid and different payment method are diplayed

R.4.3 Payment
Input: Selected payment method

Output: The order is placed and copy of that is send to the email address given by the customer at the time of login

R.4.4 Cancel Order

Description: The customer can cancel order that he has placed.

Input: User selection

Output: Confirmation message.

R.5 Manage Purchased Items

R.5.1 Feedback

Description: The customer gives feedback according to experience after using item in terms of stars.

Input: The number of stars given by the customer.

Output: The average rating is shown to the customer.

R.5.2 Return Item

Description: The customer returns the purchased item within a week if he found any issue.

Input: Selected item and reason behind returning the item

Output: Confirmation message

R.5.3 Replace Item

Description: The customer replaces the purchased item within a week if he got delivered another item.

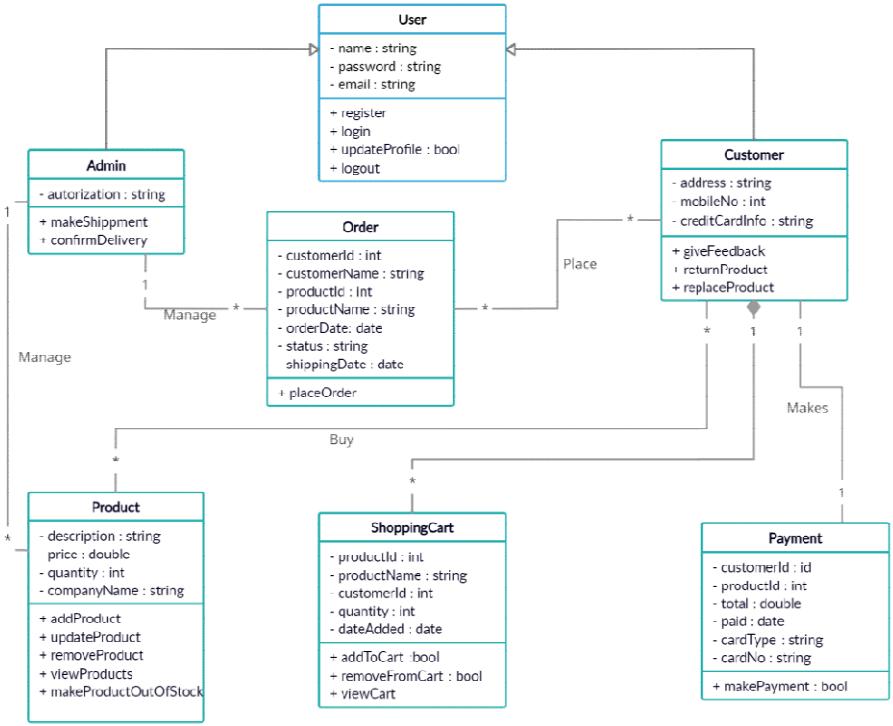
Input: User selection

Output: Confirmation message

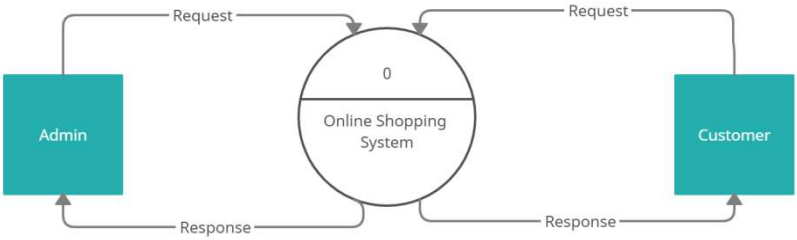
3. Design Documents

Use Case Diagram

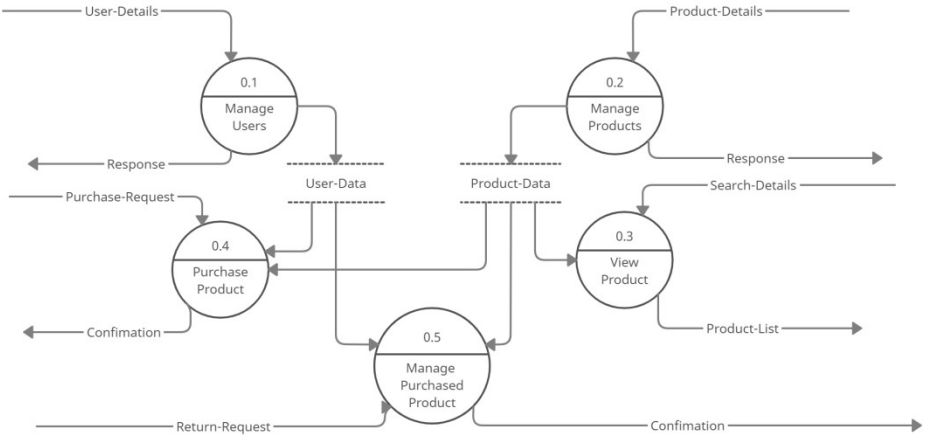
Class Diagram



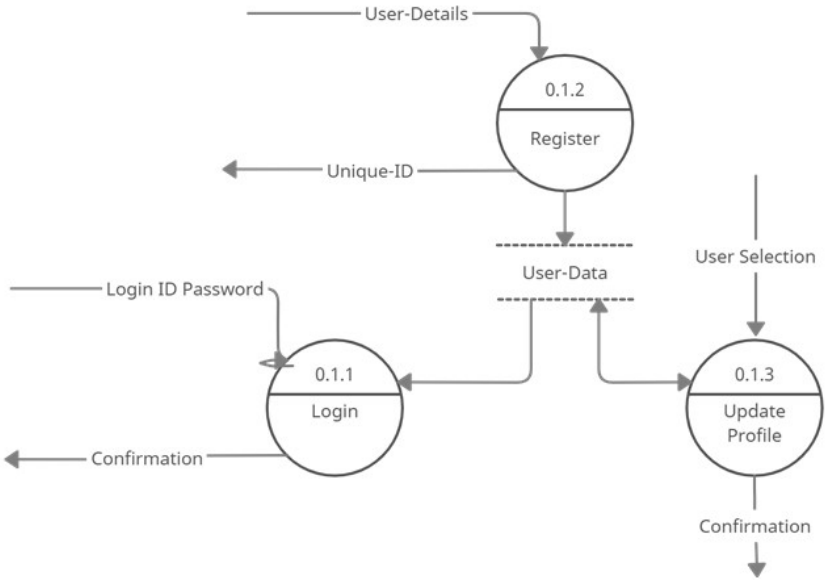
DFD Diagram

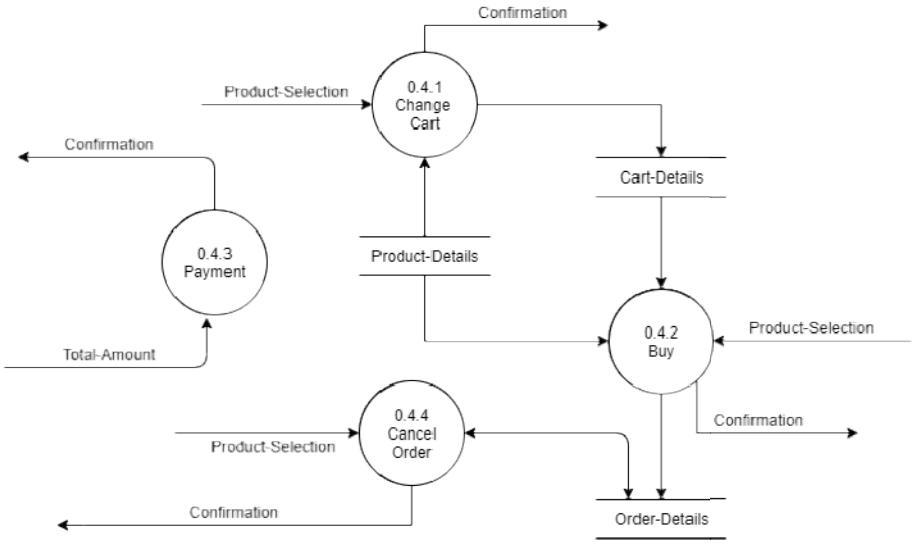


Level 0



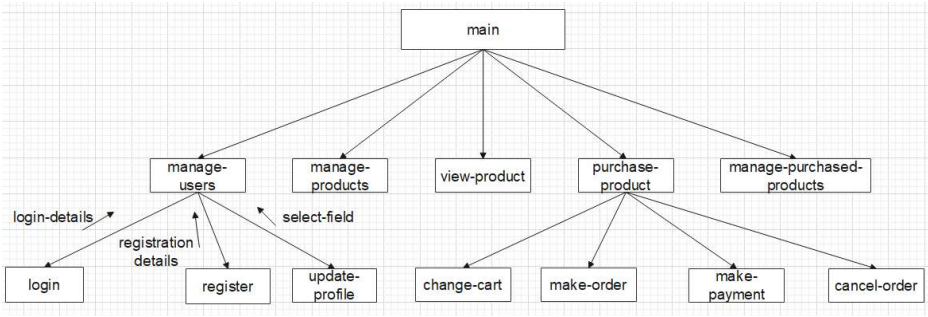
Level 1


User-Details
0.1.2
Register
Unique-ID
User-Data
User Selection
Login ID Password
0.1.1
Login
0.1.3
Update Profile
Confirmation
Confirmation

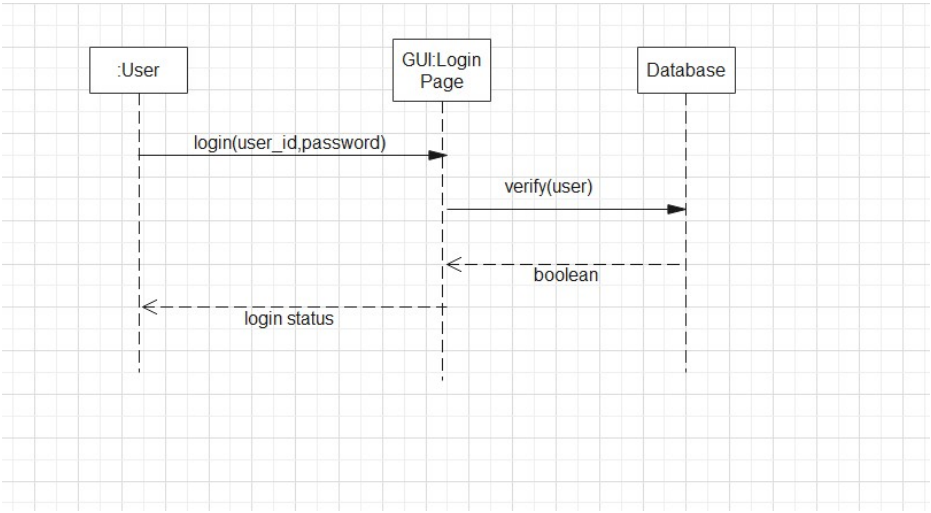

Confirmation
Product-Selection
0.4.1
Change Cart
Confirmation
Cart-Details
0.4.3
Payment
Product-Details
Total-Amount
0.4.2
Buy
Product-Selection
0.4.4
Cancel Order
Confirmation
Product-Selection
Confirmation
Order-Details

Level 2

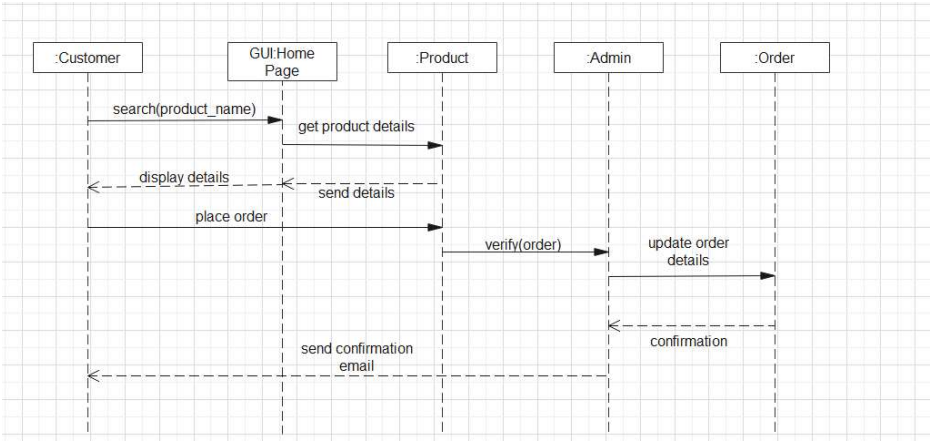
Structure Chart



Sequence Diagram

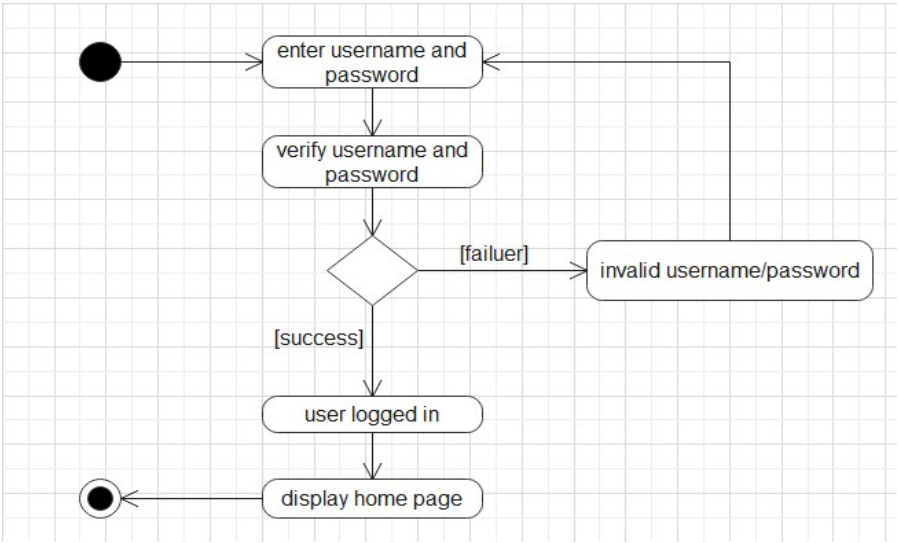


Login

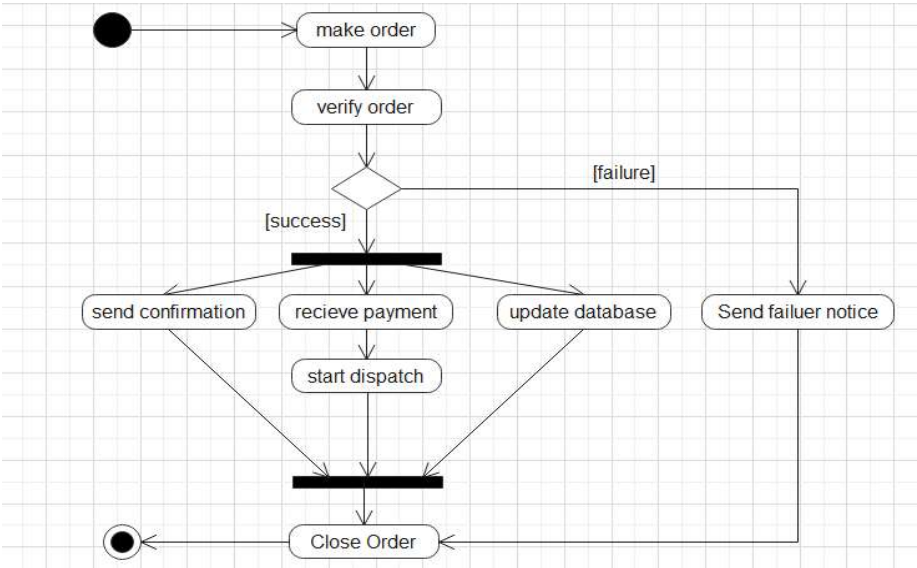


Buy Product

Activity Diagram



Login



Place Order

4. Implementation Details

Models

```
class Customer(models.Model):

    user=models.OneToOneField(User,on_delete=models.CASCADE)
    mobileNo=models.BigIntegerField()
    city=models.CharField(max_length=20,null=True,blank=True)
    state=models.CharField(max_length=20,null=True,blank=True)
    pincode=models.CharField(max_length=6,null=True,blank=True)
    address=models.TextField(max_length=250,null=True,blank=True)
```

```
class Category(models.Model):
    name=models.CharField(max_length=50)
    image=models.ImageField(upload_to="images/categories")

    def __str__(self):
        return self.name
```

```
class Product(models.Model):
    name=models.CharField(max_length=300)
    price=models.IntegerField()
    image=models.ImageField(upload_to="images/products")
    category=models.ForeignKey(Category,on_delete=models.CASCADE)
    companyName=models.CharField(max_length=50)
    quantity=models.IntegerField(default=10)
    ram=models.CharField(max_length=5,choices=RAM_CHOICES,default=None)
    storage=models.CharField(max_length=5,choices=STORAGE_CHOICES,default=None)
    color=models.CharField(max_length=6,choices=COLOR_CHOICES,default='Black')
    desc=models.TextField(max_length=800)
    discount=models.IntegerField(default=0)
    specialOffer=models.BooleanField(default=False)
    specialOfferDesc=models.TextField(max_length=300,default='No Special Offer')
    returnPolicy=models.TextField(max_length=200)
    pubDate=models.DateField()
    deliveryDays=models.IntegerField(default=4)

    def __str__(self):
        return self.name
```

```python
class Cart(models.Model):

    product=models.ForeignKey(Product,on_delete=models.CASCADE)
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    quantity=models.IntegerField()

    def __str__(self):
        return self.product.name
```

```python
class Order(models.Model):

    product=models.ForeignKey(Product,on_delete=models.CASCADE)
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    name=models.CharField(max_length=30)
    mobileNo=models.BigIntegerField()
    quantity=models.IntegerField()
    orderDate=models.DateField(default=datetime.today)
    address=models.TextField(max_length=250)
    city=models.CharField(max_length=20)
    state=models.CharField(max_length=20)
    pincode=models.CharField(max_length=6)
    orderStatus=models.BooleanField(default=0)
    deliverDate=models.DateField(null=True,blank=True)
```

Views

```
class Login(View):
    returnUrl=None

    def post(self,request):

        user=auth.authenticate(username=request.POST['username'],
            password=request.POST['password'])
        if user is not None:
            auth.login(request,user)
            if Login.returnUrl:
                return HttpResponseRedirect(Login.returnUrl)
            else:
                Login.returnUrl=None
                return redirect('/')
        else:
            messages.info(request,'Incorrect Username/Password')
            return redirect('login')
    def get(self,request):

        Login.returnUrl=request.GET.get('returnUrl')
        return render(request,'login.html')
```

```
class ViewProfile(View):

    def post(self,request):

        request.user.first_name=request.POST['firstname']
        request.user.last_name=request.POST['lastname']
        request.user.username=request.POST['username']
        request.user.email=request.POST['email']
        request.user.save()
        customer=Customer.objects.get(user_id=request.user.id)
        customer.mobileNo=request.POST['mobileno']
        customer.save()

        return redirect('/accounts/viewProfile')

    def get(self,request):

        customer=Customer.objects.get(user_id=request.user.id)
        return render(request,'viewProfile.html',{'customer':customer})
```

```
class SignUp(View):
    def post(self,request):
        firstname=request.POST['firstname']
        lastname=request.POST['lastname']
        username=request.POST['username']
        email=request.POST['email']
        password=request.POST['password']
        cpassword=request.POST['confirmpassword']
        mobileno=request.POST['mobileno']
        if password==cpassword:
            if User.objects.filter(username=username).exists():
                messages.info(request,'Username is already taken')
                return redirect('signup')
            elif User.objects.filter(email=email).exists():
                messages.info(request,'Email is already taken')
                return redirect('signup')
            else:
                user=User.objects.create_user(first_name=firstname,
                last_name=lastname,username=username,email=email,password=password)
                user.save()
                customer=Customer(user=user,mobileNo=mobileno)
                customer.save()
                return redirect('login')
        else:
            messages.info(request,'Password doesn\'t match')
            return redirect('signup')

    def get(self,request):
        return render(request,'signup.html')
```

```
def logout(request):

    auth.logout(request)
    return redirect('/')
```

```
def removeFromCart(request,productId):
    if not productId:
        return redirect('/')
    product=Product.objects.get(id=productId)
    cart=Cart.objects.filter(product=product,user=request.user)
    cart[0].delete()
    return redirect('/viewCart')
```

```python
def home(request):

    specialProducts=Product.objects.filter(specialOffer=True)
    activeProduct=specialProducts[0]
    otherSpecialProducts=specialProducts[1:]

    allProducts=[]
    categories=Category.objects.all()
    for category in categories:
        products=Product.objects.filter(category=category)
        allProducts.append(products)

    data={'allProducts':allProducts,
        'specialProducts':otherSpecialProducts,
        'activeProduct':activeProduct}
    return render(request,'index.html',data)
```

```python
def viewCart(request):
    carts=Cart.objects.filter(user=request.user)
    products=[]
    if carts:
        for cart in carts:
            product=Product.objects.get(id=cart.product.id)
            quantity=cart.quantity
            if quantity :
                products.append((product,quantity))
            else:
                cart.delete()
    data={'products':products}
    return render(request,'cart.html',data)
```

```python
def addToCart(request):
    productId=request.GET.get('id')
    if productId==None:
        return redirect('/')
    product=Product.objects.get(id=productId)
    cart=Cart.objects.filter(product=product,user=request.user)
    if not cart:
        cart=Cart(product=product,user=request.user,quantity=1)
        cart.save()
    else:
        return redirect('/')
    return redirect('/viewProducts/productDetails/'+productId)
```

```python
def changeQty(request,productId,op):
    if productId is None or op is None:
        return redirect('/')
    product=Product.objects.get(id=productId)
    cart=Cart.objects.filter(product=product,user=request.user)
    if cart:
        if op=='minus':
            cart[0].quantity-=1
        else:
            if product.quantity==cart[0].quantity:
                messages.info(request,'This product is now out of stock')
                return redirect('/viewCart')
            else:
                cart[0].quantity+=1
        cart[0].save()
    return redirect('/viewCart#'+str(productId))
```

```python
def orders(request):

    if request.method=='GET':

        orders=Order.objects.filter(user=request.user).order_by('-orderDate')

        data={'orders':orders}
        return render(request,'orders.html',data)
```

```python
def cancelOrder(request):

    orderId=request.GET.get('id')

    if orderId:
        order=Order.objects.get(id=orderId)
        if not order.orderStatus:
            order.product.quantity+=order.quantity
            order.product.save()
            order.delete()
            return redirect('/orders')
        else:
            return redirect('/')
    else:
        return redirect('/')
```

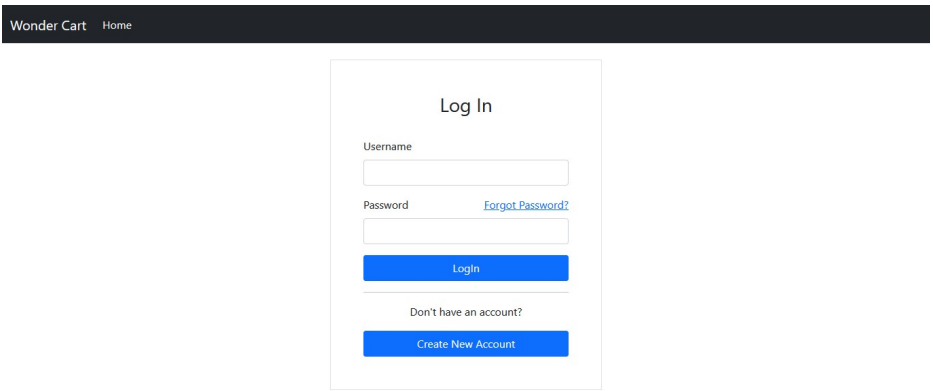
```
class Checkout(View):
    def get(self,request):
        productId=request.GET.get('id')
        products=[]
        customer=Customer.objects.get(user_id=request.user.id)
        if productId:
            product=Product.objects.get(id=productId)
            if not product.quantity:
                return redirect('/')
            cart=Cart.objects.filter(product=product,user=request.user)
            if not cart:
                cart=Cart(product=product,user=request.user,quantity=1)
                cart.save()
            carts=Cart.objects.filter(product=product,user=request.user)
        else:
            carts=Cart.objects.filter(user=request.user)
        if not carts:
            return redirect('/')
        for cart in carts:
            product=Product.objects.get(id=cart.product.id)
            products.append((product,cart.quantity))
        data={'products':products,'id':productId,'customer':customer}
        return render(request,'checkout.html',data)
    def post(self,request):
        firstname=request.POST.get('firstname')
        lastname=request.POST.get('lastname')
        mobileNo=request.POST.get('mobileno')
        pincode=request.POST.get('pincode')
        city=request.POST.get('city')
        state=request.POST.get('state')
        address=request.POST.get('address')
        if request.POST.get('save')=='True':
            customer=Customer.objects.get(user_id=request.user.id)
            customer.pincode=pincode
            customer.mobileNo=mobileNo
            customer.city=city
            customer.state=state
            customer.address=address
            customer.save()
            request.user.first_name=firstname
            request.user.last_name=lastname
            request.user.save()
        productId=request.POST.get('id')
        if productId:
            product=Product.objects.get(id=productId)
            carts=Cart.objects.filter(product=product,user=request.user)
        else:
            carts=Cart.objects.filter(user=request.user)
        for cart in carts:
            cart.product.quantity-=cart.quantity
            cart.product.save()
            order=Order(product=cart.product,user=cart.user,
                quantity=cart.quantity,
                name=firstname+' '+lastname,mobileNo=mobileNo,
                pincode=pincode,city=city,state=state,address=address)
            order.save()
            cart.delete()
        messages.success(request,'Your Order is placed successfully')
        return redirect('/orders')
```

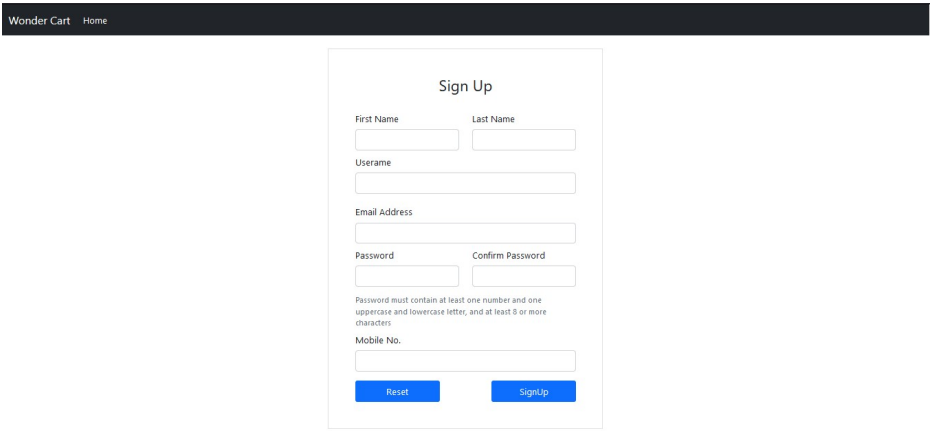
```
def productDetails(request,id):
    if id==None:
        return redirect('/')
    product=Product.objects.get(id=id)
    products=Product.objects.filter(name=product.name)
    sProducts=set()
    for prod in products:
        sProducts.add((prod.id,prod.get_color_display(),prod.get_ram_display(),
            prod.get_storage_display()))
    data={'product':product,'sProducts':sProducts}
    return render(request,'productDetails.html', data)
```

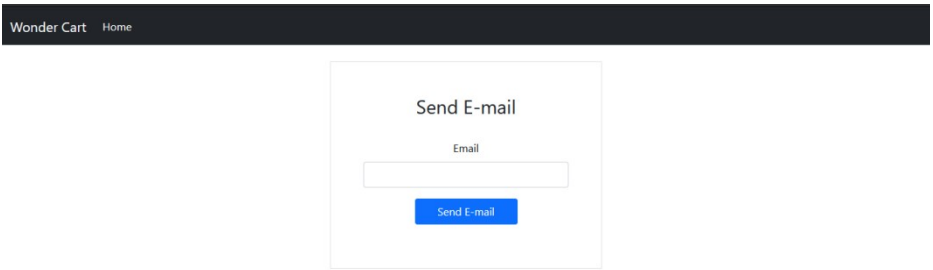
```
def afilter(request,selectedCategory):
    price=request.GET.get('price')
    company=request.GET.get('company')
    if not selectedCategory:
        return redirect('/')
    allProducts=[]
    category=Category.objects.get(name=selectedCategory)
    products=Product.objects.filter(category=category)
    companies=set()
    for product in products:
        companies.add(product.companyName)
    if price and company:
        if price=='lth':
            products=Product.objects.order_by('price').filter(category=category,
                companyName=company)
        else:
            products=Product.objects.order_by('-price').filter(category=category,
                companyName=company)
    elif not price and company:
        products=Product.objects.filter(category=category,companyName=company)
    else:
        if price == 'lth':
            products=Product.objects.order_by('price').filter(category=category)
        else:
            products=Product.objects.order_by('-price').filter(category=category)
    allProducts.append(products)
    data={'allProducts':allProducts,'category':selectedCategory,
        'companies':companies}
    return render(request,'afilter.html',data)
```

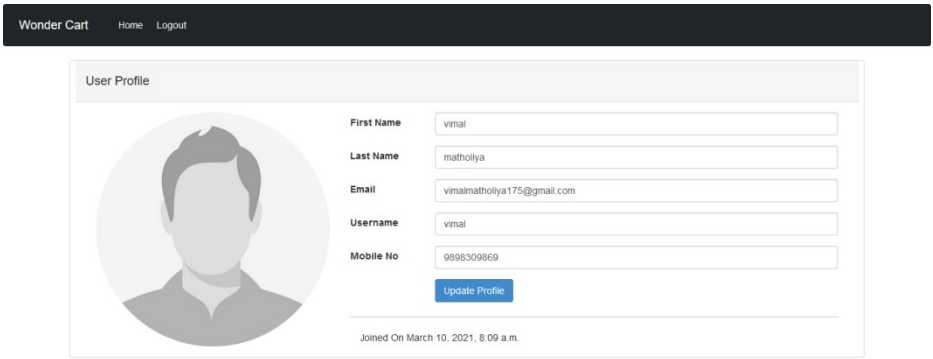
```
def search(request):
    keyword=request.GET.get('keyword',None)
    if keyword==None:
        return redirect('/')
    oKeyword=keyword
    if keyword[len(keyword)-1]=='s':
        keyword=keyword[:len(keyword)-1]
    keywords=keyword.split(' ')
    matchedCategories=[]
    categories=Category.objects.all()
    for cat in categories:
        products=Product.objects.filter(category=cat)
        matchedProducts=[]
        for product in products:
            if keyword==product.name.lower() or
                    keyword == product.companyName.lower() or
                    keyword == product.category.name.lower():
                matchedProducts.append(product)
            elif len(keywords) == 2 and
                    keywords[0] == product.companyName.lower() and
                    keywords[1]== product.category.name.lower():
                matchedProducts.append(product)
            elif len(keywords) >= 1:
                count=0
                for word in keywords:
                    if word in product.name.lower():
                        count+=1
                if count>=5:
                    matchedProducts.append(product)
                elif count>=4:
                    matchedProducts.append(product)
                elif count>=3:
                    matchedProducts.append(product)
                elif count>=2:
                    matchedProducts.append(product)
        if len(matchedProducts)==0:
            continue
        matchedCategories.append(matchedProducts)
    data={'matchedCategories':matchedCategories,'keyword':oKeyword}
    return render(request,'search.html',data)
```

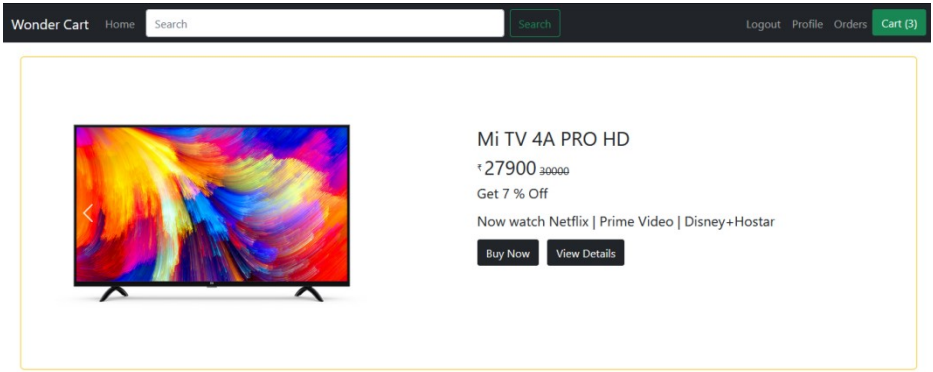
5. Workflow / Layouts

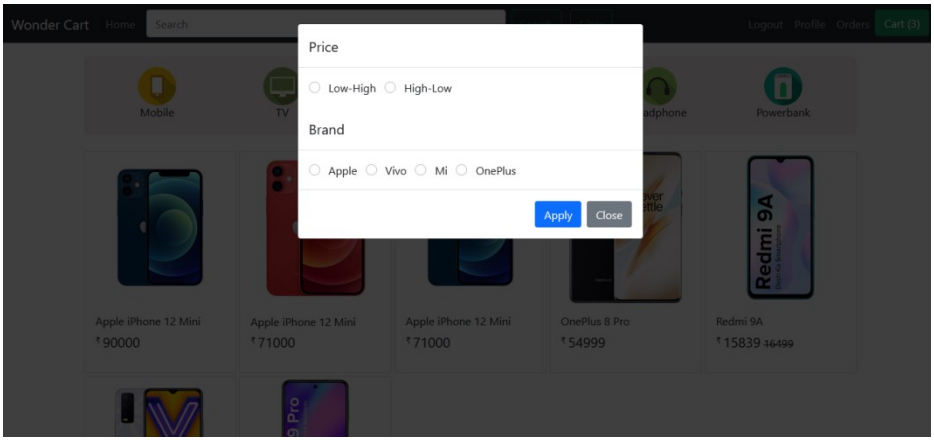


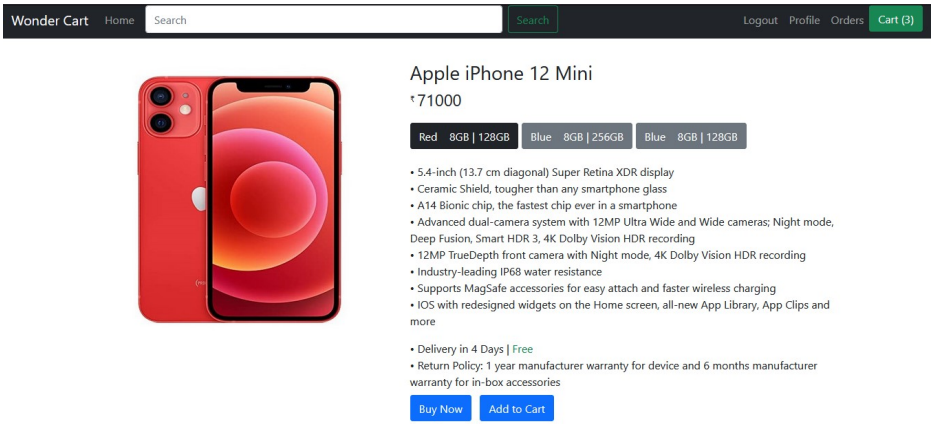


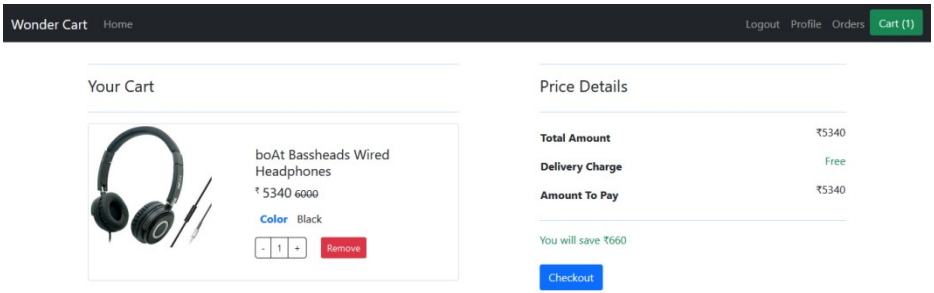


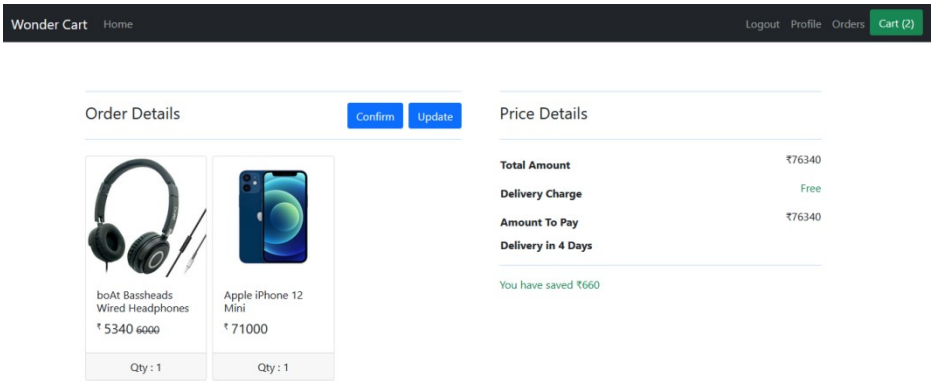

Wonder Cart
Home
Logout
User Profile
First Name
vimal
Last Name
matholiya
Email
vimalmatholiya175@gmail.com
Username
vimal
Mobile No
9898309869
Update Profile
Joined On March 10, 2021, 8:09 a.m.

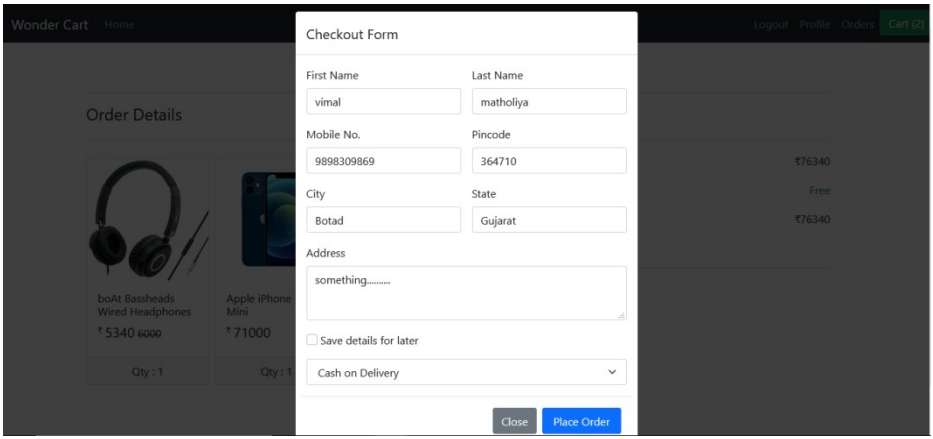

Wonder Cart
Home
Search
Search
Logout
Profile
Orders
Cart (3)
Mi TV 4A PRO HD
₹27900 30000
Get 7 % Off
Now watch Netflix | Prime Video | Disney+Hostar
Buy Now
View Details
Mobile
TV
Laptop
Tablet
Headphone
Powerbank
Mobile
Apple iPhone 12 Mini
₹71000
Redmi 9A
₹15839 16499
OnePlus 8 Pro
₹54999
Vivo Y20
₹12999
Redmi Note 9 Pro
₹12999

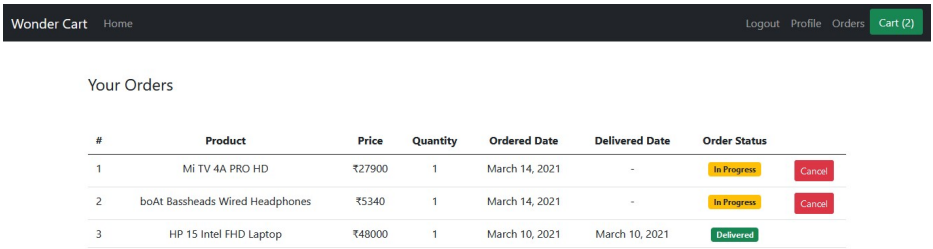

Wonder Cart
Home
Search
Logout
Profile
Orders
Cart (3)
Price
Low-High
High-Low
Brand
Apple
Vivo
Mi
OnePlus
Apply
Close
Mobile
TV
Powerbank
Apple iPhone 12 Mini
₹90000
Apple iPhone 12 Mini
₹71000
Apple iPhone 12 Mini
₹71000
OnePlus 8 Pro
₹54999
Redmi 9A
₹15839 ₹6499

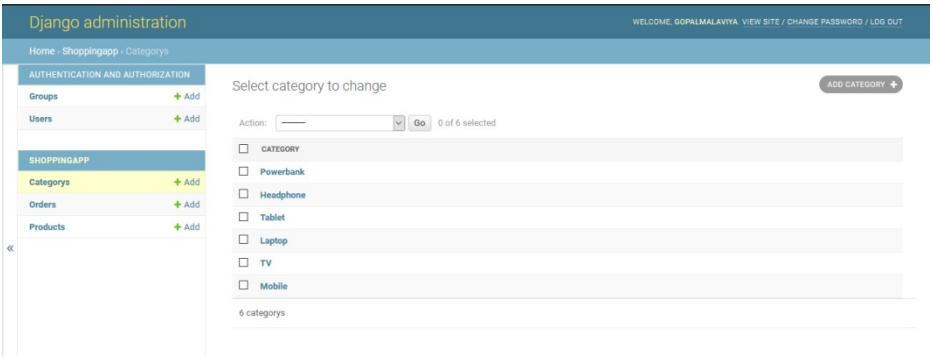

Wonder Cart
Home
Search
Search
Logout
Profile
Orders
Cart (3)
Apple iPhone 12 Mini
₹71000
Red 8GB | 128GB
Blue 8GB | 256GB
Blue 8GB | 128GB
• 5.4-inch (13.7 cm diagonal) Super Retina XDR display
• Ceramic Shield, tougher than any smartphone glass
• A14 Bionic chip, the fastest chip ever in a smartphone
• Advanced dual-camera system with 12MP Ultra Wide and Wide cameras; Night mode, Deep Fusion, Smart HDR 3, 4K Dolby Vision HDR recording
• 12MP TrueDepth front camera with Night mode, 4K Dolby Vision HDR recording
• Industry-leading IP68 water resistance
• Supports MagSafe accessories for easy attach and faster wireless charging
• IOS with redesigned widgets on the Home screen, all-new App Library, App Clips and more
• Delivery in 4 Days | Free
• Return Policy: 1 year manufacturer warranty for device and 6 months manufacturer warranty for in-box accessories
Buy Now
Add to Cart

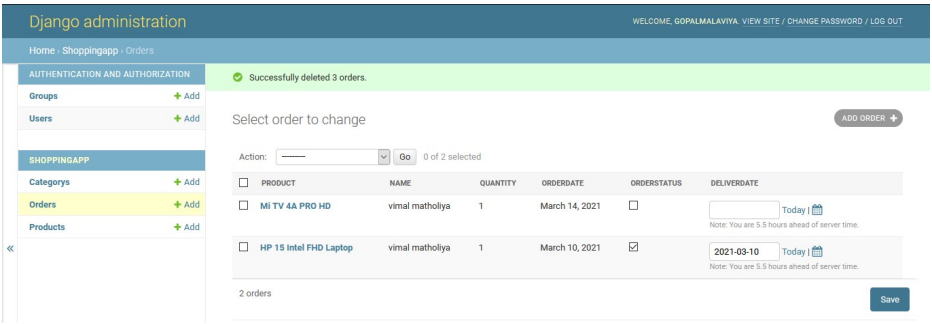

Wonder Cart
Home
Logout
Profile
Orders
Cart (1)
Your Cart
boAt Bassheads Wired Headphones
₹5340 6000
Color Black
1
Remove
Price Details
Total Amount
₹5340
Delivery Charge
Free
Amount To Pay
₹5340
You will save ₹660
Checkout

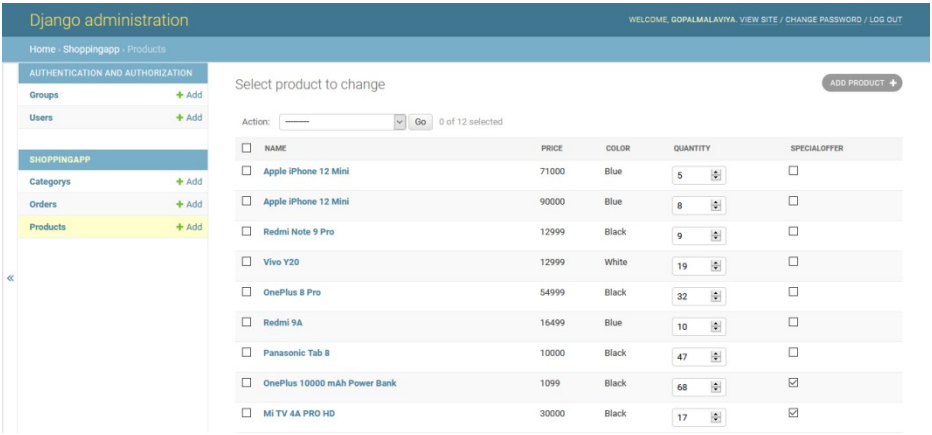

Wonder Cart
Home
Logout
Profile
Orders
Cart (2)
Order Details
Confirm
Update
boAt Bassheads Wired Headphones
₹ 5340 6000
Qty : 1
Apple iPhone 12 Mini
₹ 71000
Qty : 1
Price Details
Total Amount
₹76340
Delivery Charge
Free
Amount To Pay
₹76340
Delivery in 4 Days
You have saved ₹660


Checkout Form
First Name
vimal
Last Name
matholiya
Mobile No.
9898309869
Pincode
364710
City
Botad
State
Gujarat
Address
something.....
Save details for later
Cash on Delivery
Close
Place Order


Wonder Cart
Home
Logout
Profile
Orders
Cart (2)
Your Orders
Product Price Quantity Ordered Date Delivered Date Order Status
1 Mi TV 4A PRO HD ₹27900 1 March 14, 2021 - In Progress Cancel
2 boAt Bassheads Wired Headphones ₹5340 1 March 14, 2021 - In Progress Cancel
3 HP 15 Intel FHD Laptop ₹48000 1 March 10, 2021 March 10, 2021 Delivered


Django administration
WELCOME, GOPALMALAVIYA. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Shoppingapp › Categorys
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
SHOPPINGAPP
Categorys + Add
Orders + Add
Products + Add
Select category to change
ADD CATEGORY +
Action: ——— Go 0 of 6 selected
CATEGORY
Powerbank
Headphone
Tablet
Laptop
TV
Mobile
6 categorys


Django administration
WELCOME, GOPALMALAVIYA. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Shoppingapp › Orders
Successfully deleted 3 orders.
Select order to change
ADD ORDER +
Action: ——— Go 0 of 2 selected
PRODUCT
NAME
QUANTITY
ORDERDATE
ORDERSTATUS
DELIVERDATE
Mi TV 4A PRO HD
vimal matholiya
1
March 14, 2021
Today |
Note: You are 5.5 hours ahead of server time.
HP 15 Intel FHD Laptop
vimal matholiya
1
March 10, 2021
2021-03-10
Today |
Note: You are 5.5 hours ahead of server time.
2 orders
Save


Django administration
WELCOME, GOPALMALAVIYA. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Shoppingapp › Products
Select product to change
ADD PRODUCT +
Action: ——— Go 0 of 12 selected
NAME
PRICE
COLOR
QUANTITY
SPECIALOFFER
Apple iPhone 12 Mini 71000 Blue 5
Apple iPhone 12 Mini 90000 Blue 8
Redmi Note 9 Pro 12999 Black 9
Vivo Y20 12999 White 19
OnePlus 8 Pro 54999 Black 32
Redmi 9A 16499 Blue 10
Panasonic Tab 8 10000 Black 47
OnePlus 10000 mAh Power Bank 1099 Black 68
Mi TV 4A PRO HD 30000 Black 17

6. Conclusion

All the functionality are implemented after understanding all models and diagrams of the system.

Functionality successfully implemented are as below :

- Login
- Registration
- Reset password
- View / Update profile
- Add to cart
- Remove from cart
- Change quantity in cart
- Search and Filter
- Place order

After the implementation all functionalities were successfully tested and working properly.

7. Limitations and Future Extension

Limitations :

- Search and Filter functionalities are not user friendly , they work for some of the cases not for all.
- In current System user can not return or replace product.

Future Extension :

- We are going to implement make payment functionality in future.
- We will improve user interface for better interaction between system and users.
- We will improve search and filter functionalities in future.

8. Bibliography

References :

- **www.docs.djangoproject.com**
- **www.stackoverflow.com**
- **www.codewithharry.com**
- **www.github.com**