A PROJECT REPORT
ON

ECOMMERCE-WEBSITE

By

Jaydev Lakhamanbhai Bambhaniya
(CE008-19CEUOF027)
Kevin Chandreshbhai Bhanderi
(CE-011-19CEUEG077)

B.Tech CE Semester-IV
Subject: Software Project

Guided by:

Prof. Pinkal C. Chauhan,
Assistant Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University,Nadiad.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer Engineering

Dharmsinh Desai University

<u>CERTIFICATE</u>

This is to certify that the practical / term work carried out in the subject of

Software Project and recorded in this journal is the

bonafide work of

Jaydev Lakhamanbhai Bambhaniya
(CE008-19CEUOF027)
Kevin Chandreshbhai Bhanderi
(CE-011-19CEUEG077)

of B.Tech semester **IV** in the branch of **Computer Engineering**

during the academic year
2020-2021.

<u>Prof. Pinkal C. Chauhan,</u>	**<u>Dr. C. K. Bhensdadia</u>**,
Assistant Professor,	Head,
Dept. of Computer Engg. ,	Dept. of Computer Engg.,
Faculty of Technology,	Faculty of Technology,
Dharmsinh Desai University,Nadiad.	Dharmsinh Desai University,Nadiad.

Table Of Content

1. Abstract

- **E-Commerce** is the online buying and selling process that is extremely important in **our** daily life now. The foremost **reason** behind the growth of Internet users besides social media in **e-commerce**. **E-Commerce** is at the heart of the Internet and **e-commerce** is as important as a heart is for a body.

- This is an one type of Ecommerce System where user can buy and sell old products from/to the admin according to their requirements. Admin can sell old products which are available in stock at suited price. Admin can Buy Products from customer and do some modification and put for sell on own price. Customer can buy Products or add to cart for future requirements.

2. Introduction

2.1 Brief Introduction

- This e-commerce website is somehow related with OLX, In this website there are two types of customer.
    1) Buyer
    2) Seller
- Seller can sell their old products with proper details and its feasible cost, then if admin wants to buy that product then admin can buy and make payment by any method.
- Same way, Buyer can buy the products (Second hand) by visiting the website, and they can make payment by any method.
- Buyer can add products to the cart as well, if they want to buy multiple products, and they can also remove the product from the cart, and at last they make payment for all products at a time.

2.2 Tools/Technologies Used

- ❖ **Technologies::**
    - Django Framework
    - Python
    - MySQL
    - HTML
    - CSS
- ❖ **Tools::**
    - Git
    - Visual Studio Code
- ❖ **Platform::**
    - Local Development Server

3. Requirement Specifications

3.1 Product Scope

This System is designed to Provide Service between Seller and Buyer. The aim for designing this product is the '**Sell High-Buy Low'.** So, that seller get best price for his/her old thing and buyer can buy old thing at low cost.

3.2 System Functional Requirements

1. Manage Account

R.1.1 Sign Up

- ➢ Description
  - New user or New admin have to signup first for use this website after signup user can go into home page of this website.
- ➢ Input
  - User has to enter email, username, password, phone no., date of birth
- ➢ Output
  - Asking For OTP given by system via entered mail. If OTP verification success then and then Registration Done.
- ➢ Process
  - After entering the email and password first website check for validation if email id is already registered then ask for other email id, Otherwise user get OTP in given email which is used for verification.

R.1.2 Login

R.1.2.1 Admin Login

- ➢ Description
  - Here admin has to login using email id and after login, he/she can manage the product and whole website's data.
- ➢ Input
  - Admin has to enter email id and password to access this website and modify product's details.
- ➢ Output
  - Admin logged in website.

R.1.2.2 Customer Login

- Description
  - To log in website user has enter to log in using Registered Email id and password then he/she can buy or sell the product according to their requirement with admin permission.
- Input
  - User has to enter the email-id and password.
- Output
  - User logged in.
- Process
  - System check for validation (correct email-id and correct password and verified email-id).
  - If system identified incorrectness then put message to enter correct information

R.1.3 Edit Profile

- Description
  - In this functionality user can modify its profile like user can change his/her username, password, profile picture ,phone no., address etc…
- Input
  - Choose details which user's wants to modify.
- Output
  - Successfully saved all the changes to the profile.

2. Manage Product

R.3.1 Show product

- Description
  - Admin can show the products according to stock and price.
- Input
  - Product's list and their information
- Output
  - Product display on the website.

R.3.2 Update Product

- Description
  - Admin can update details (like picture, price, availability…) of product currently in website.
- Input
  - New information of product.
- Output
  - Product details updated.

R.3.3 Add Product

- Description
  - Admin can add new product into website according to the product category.
- Input
  - Information of new product.
- Output
  - Product added successfully.

R.3.4 Delete Product

- Description
  - If Admin wants to delete Input
- Input
  - New information of product.
- Output
  - Product details updated.

3. Manage Payment

R.4.1 Make payment

4.1.1 Select Option for payment

- Input
  - User can select any type of payment method like online (GPay, PhonePe, PayTm) or offline method (COD).
- Output
  - If Chosen option is COD, then display payment successful Or else ask for amount details and do transaction.

R.4.2 check payment status

- Description
  - If user wants to check weather payment made done or not or if use wants to know about details of the payment.
- Input
  - Select the product id for which user wants to know the details of payment.
- Output
  - Displays the details of payment for that product.

4. Manage Order

R.5.1 Track order

- Description
  - Customer can track their order using order Id.
- Input
  - Order id and date of order placed.
- Output
  - Status of order display.

R.5.2 Cancel Order

- Description
  - Customer can cancel their order using order Id.
- Input
  - Selection
- Process
  - Check for order whether its delivered or not if its delivered then order cannot be cancel.
- Output
  - Cancel order successfully and display information about refund if payment placed at online.

R.5.3 Show order

- Input
  - Select option "View Order"
- Output
  - Displayed all order history placed from current logged in account.

3.3 Other Nonfunctional Requirements

1. Performance

- The system must be interactive and must not involve long delays. Though in case of opening the app components or loading the page the system shows the delays less than 2 seconds.

2. Safety

- The users' data is highly personal. The system has authorization to avoid any un- authorized access to user's private data.

3. Reliability

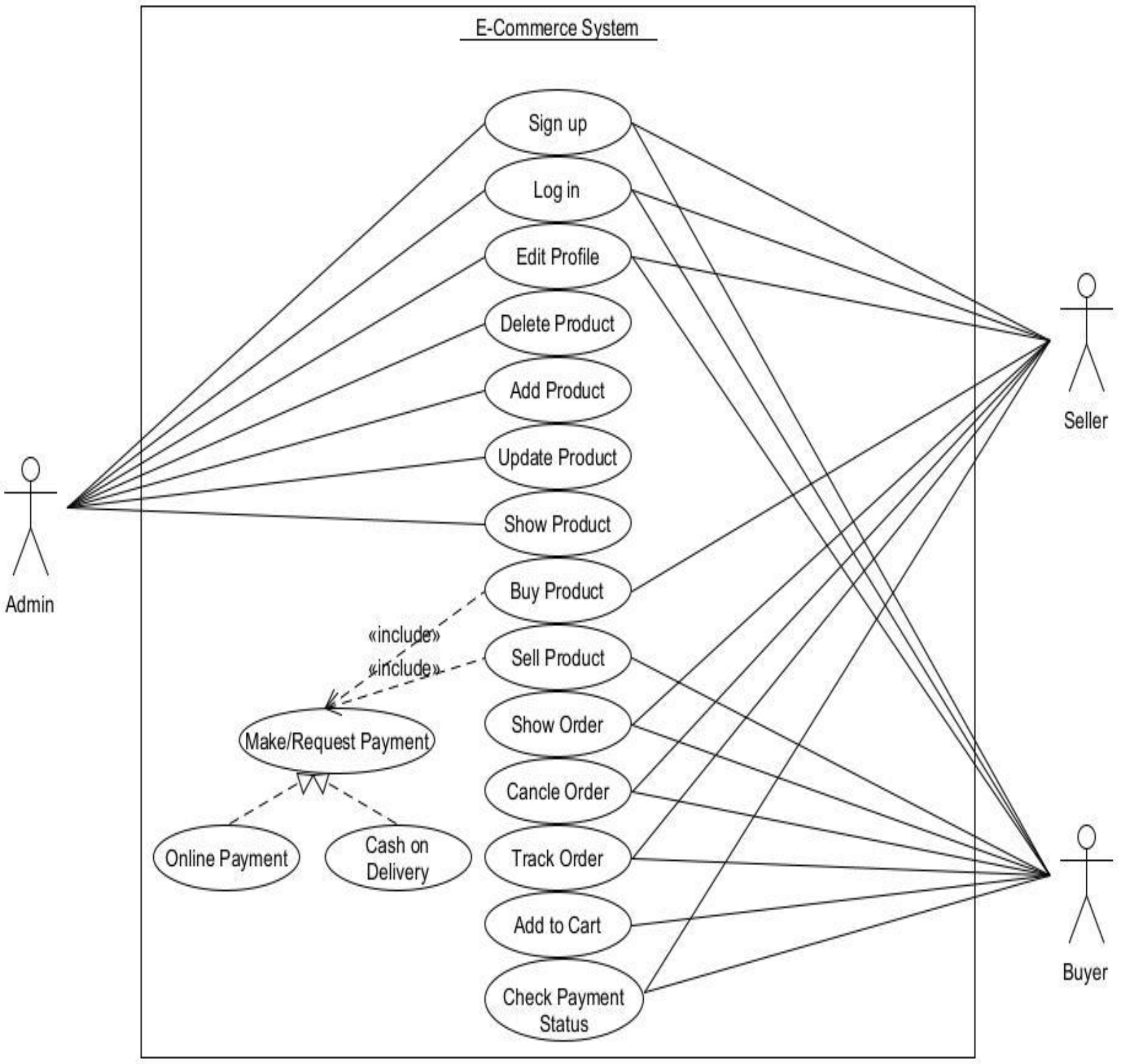
- As the system has personal data, its reliability is the major factor for consideration.

4. Database

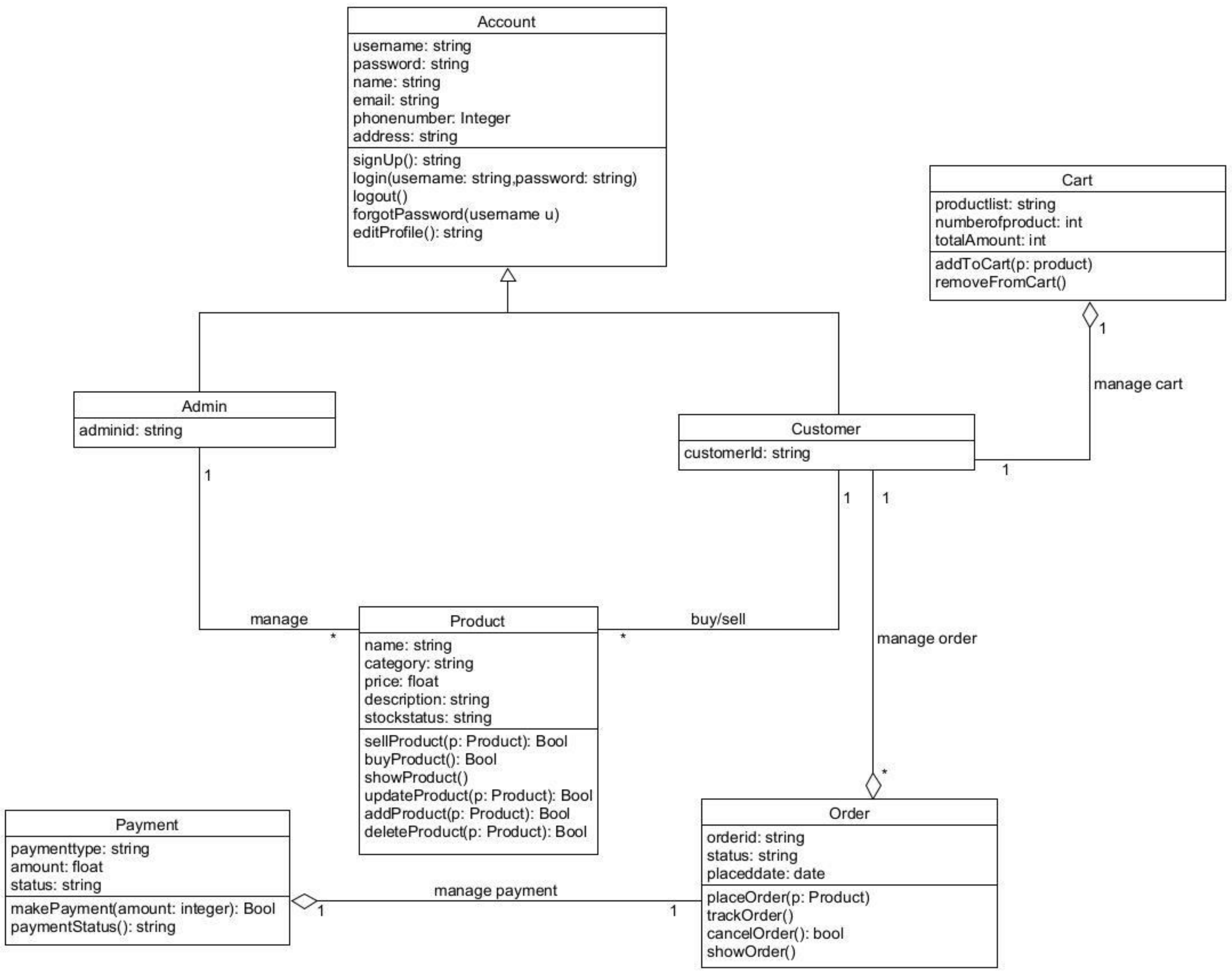
- System requires to access user submissions, versions, reviews and profile data fast to maintain the performance.

4. Design

4.1 Use Case Diagram

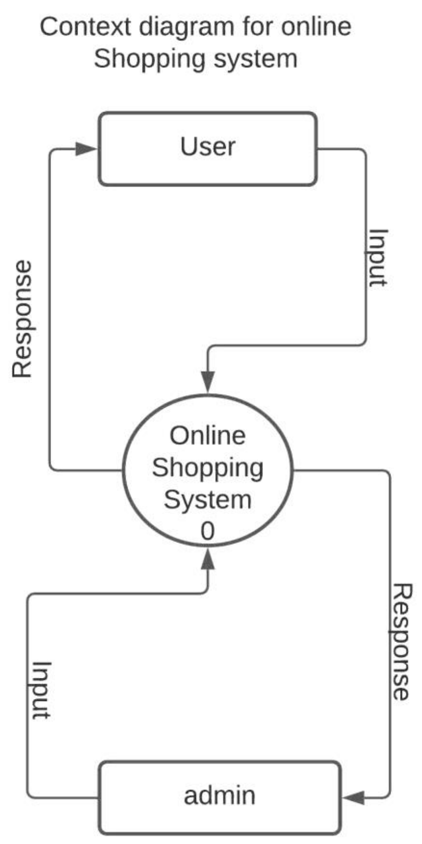


4.2 Class Diagram

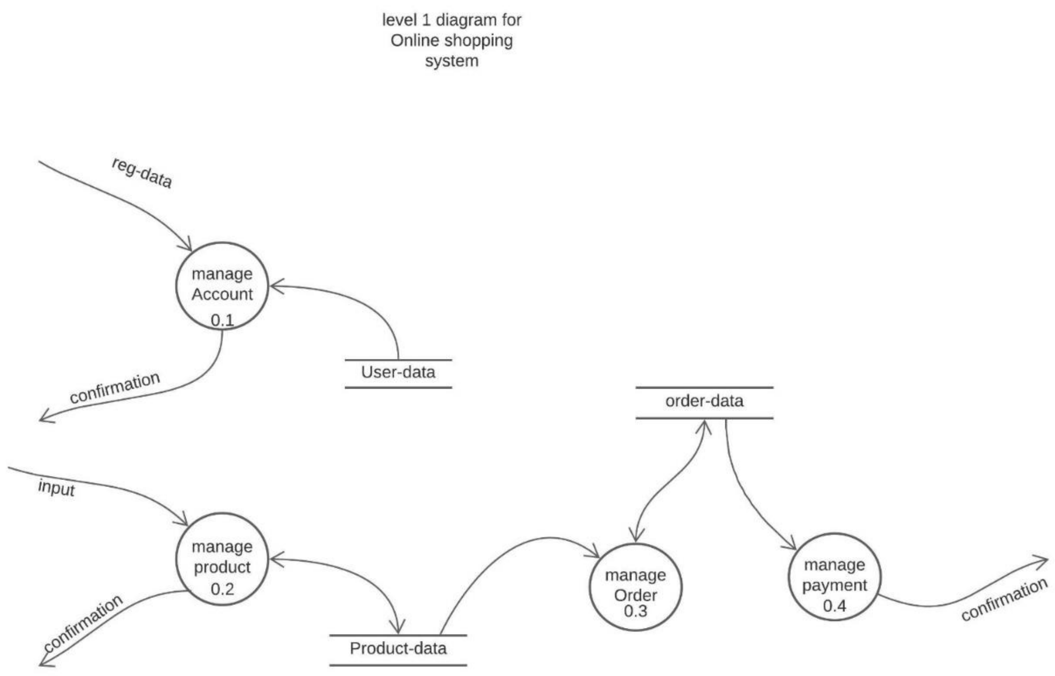


4.3 Data Flow Diagram

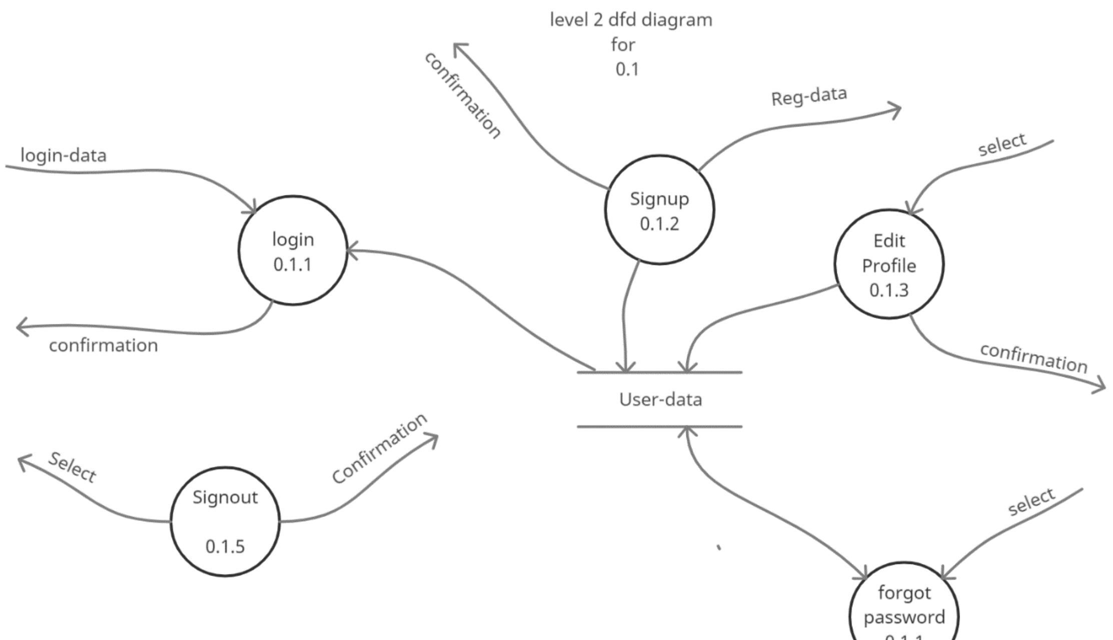
4.3.1 Level 0 DFD Diagram

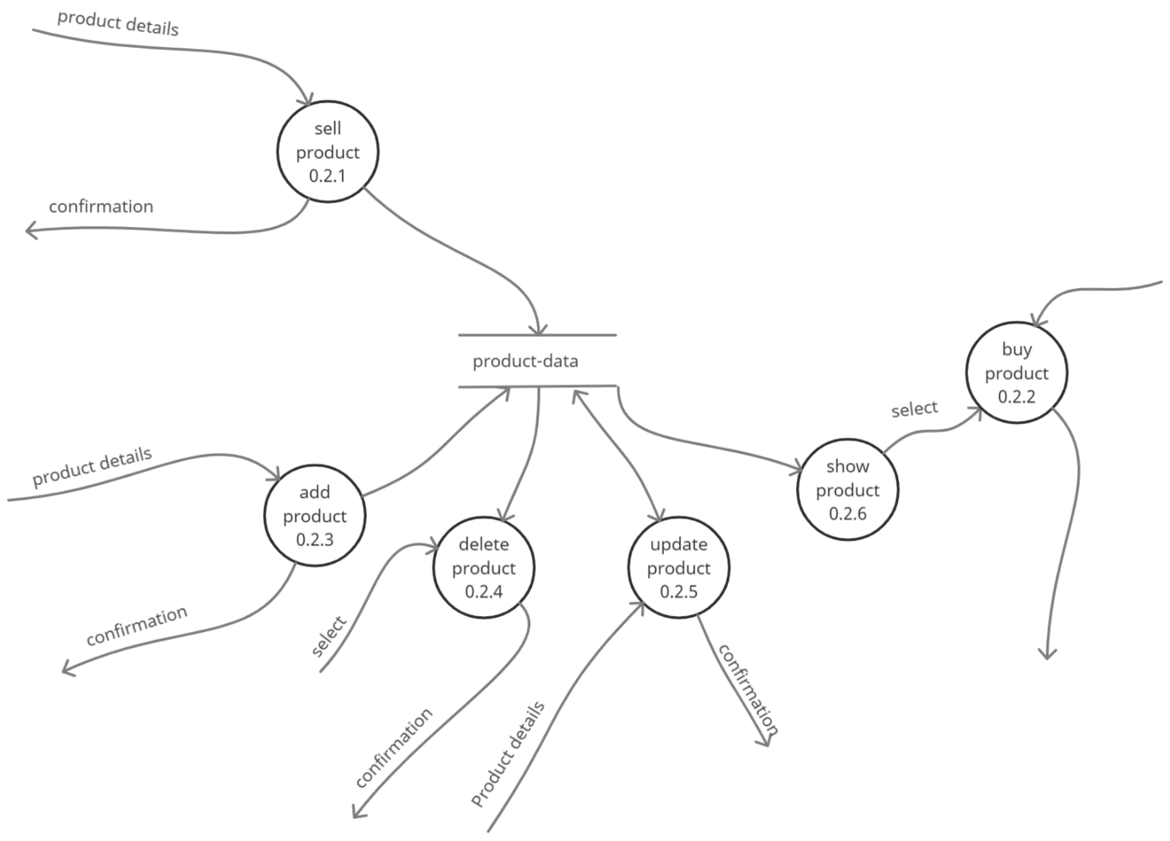


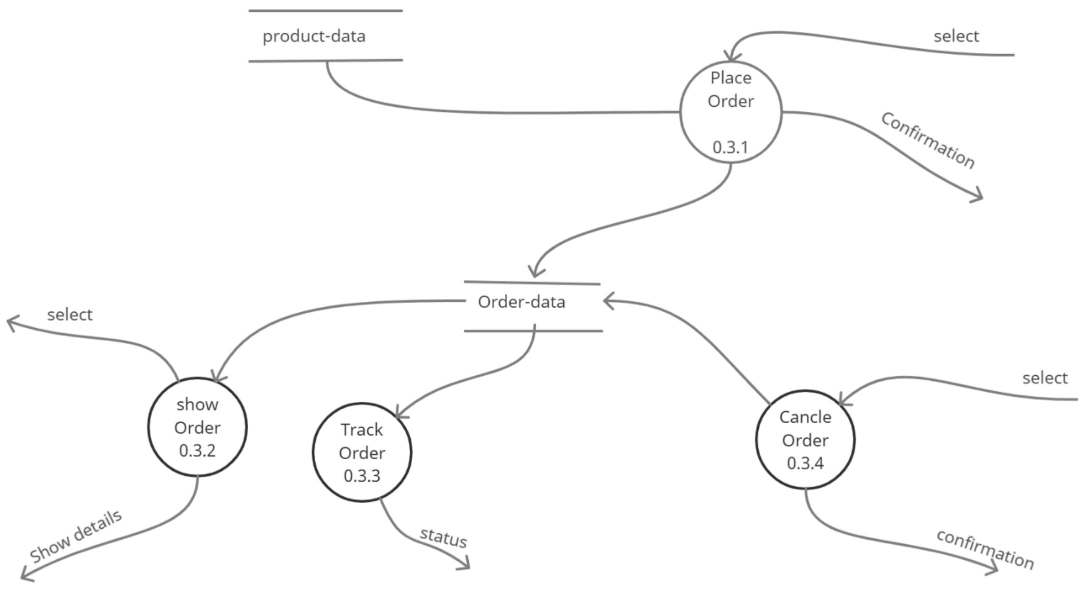
4.3.2 Level 1 DFD Diagram



4.3.3 Level 2 DFD Diagram

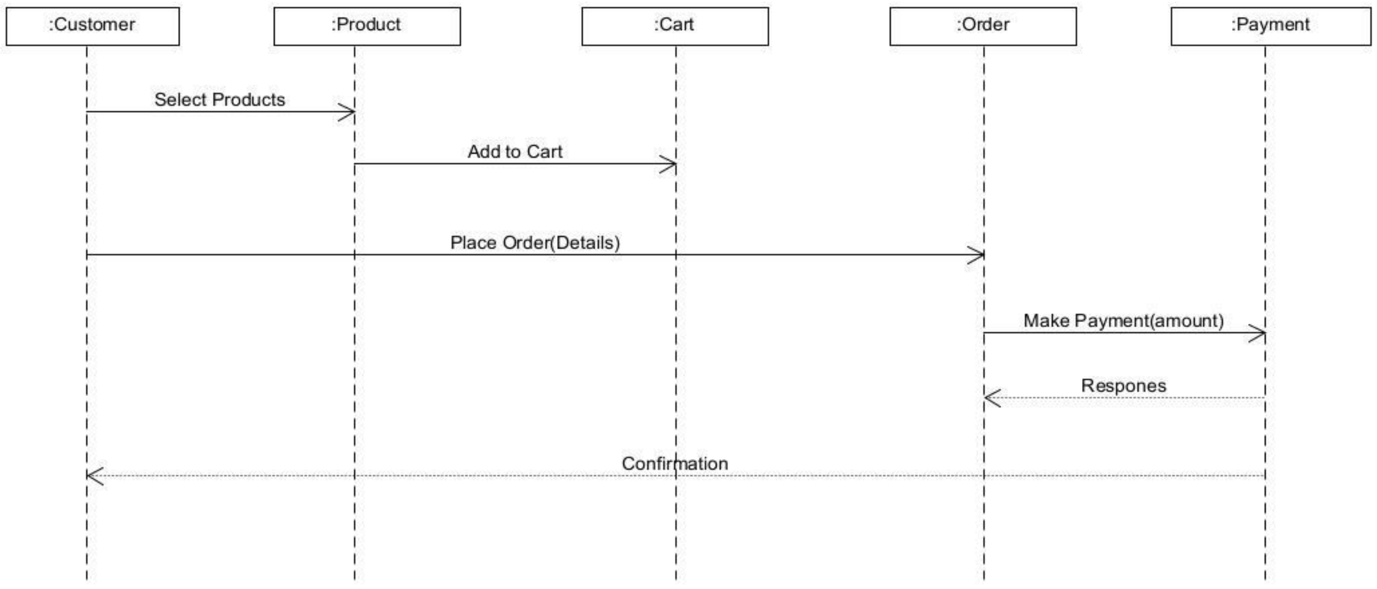



product details
sell product 0.2.1
confirmation
product-data
buy product 0.2.2
select
show product 0.2.6
product details
add product 0.2.3
delete product 0.2.4
update product 0.2.5
confirmation
select
confirmation
Product details
confirmation

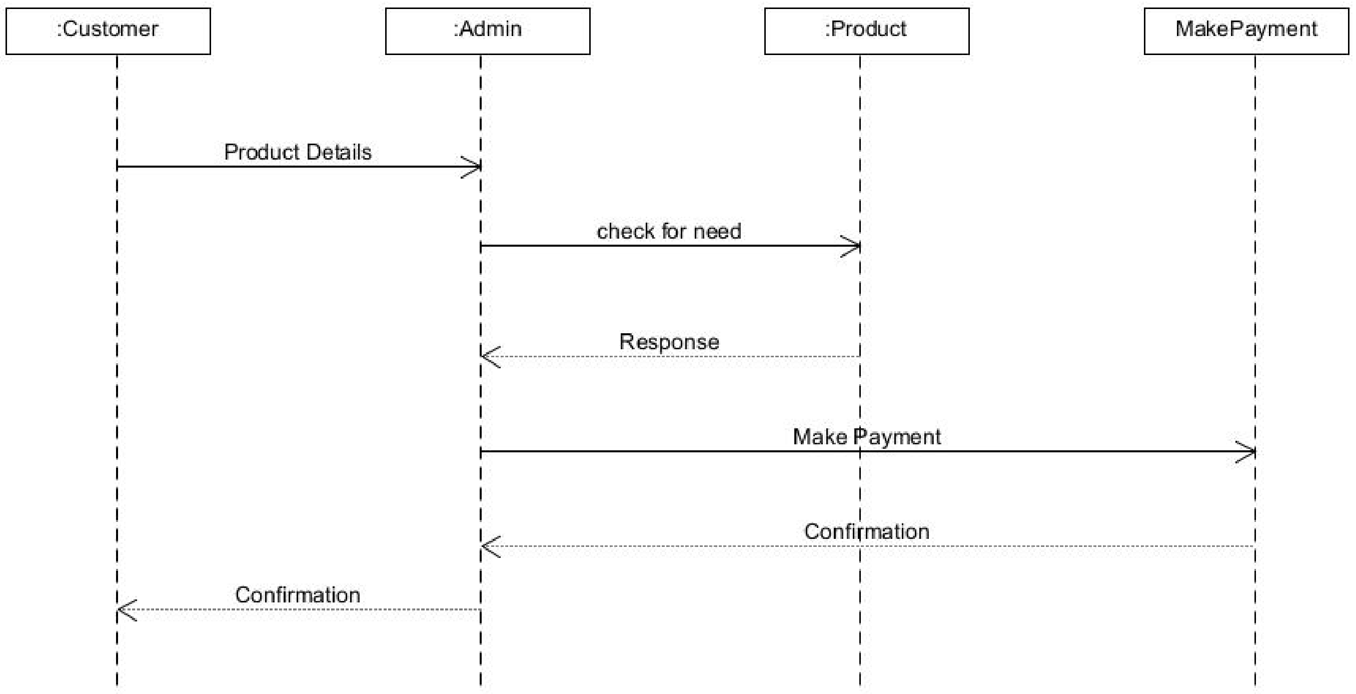

product-data
select
Place Order 0.3.1
Confirmation
Order-data
select
show Order 0.3.2
Track Order 0.3.3
Cancle Order 0.3.4
select
Show details
status
confirmation

4.4 Sequence Diagram

4.4.1 Sequence Diagram For Buy Product Use Case

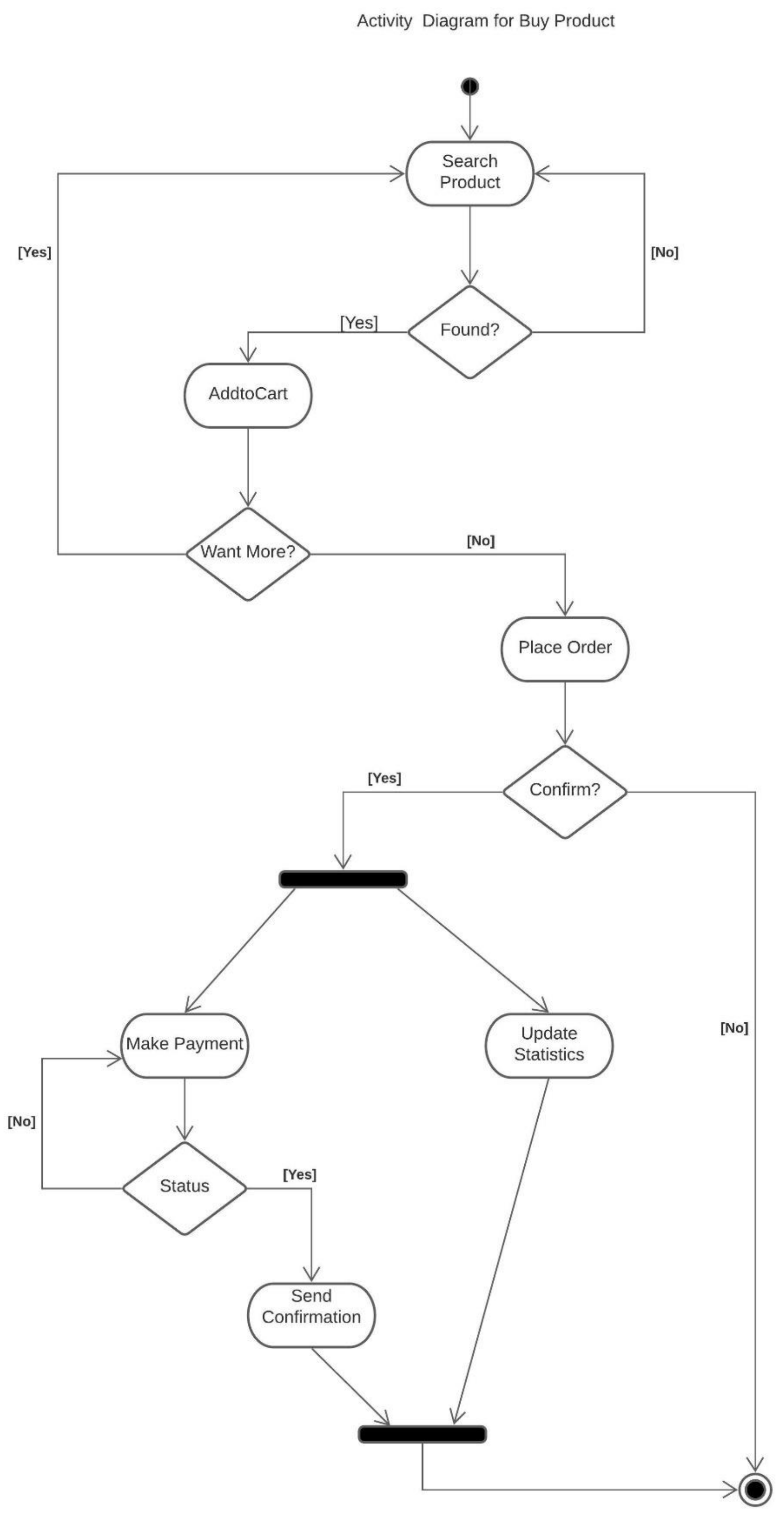


4.4.2 Sequence Diagram For Sell Product Use Case

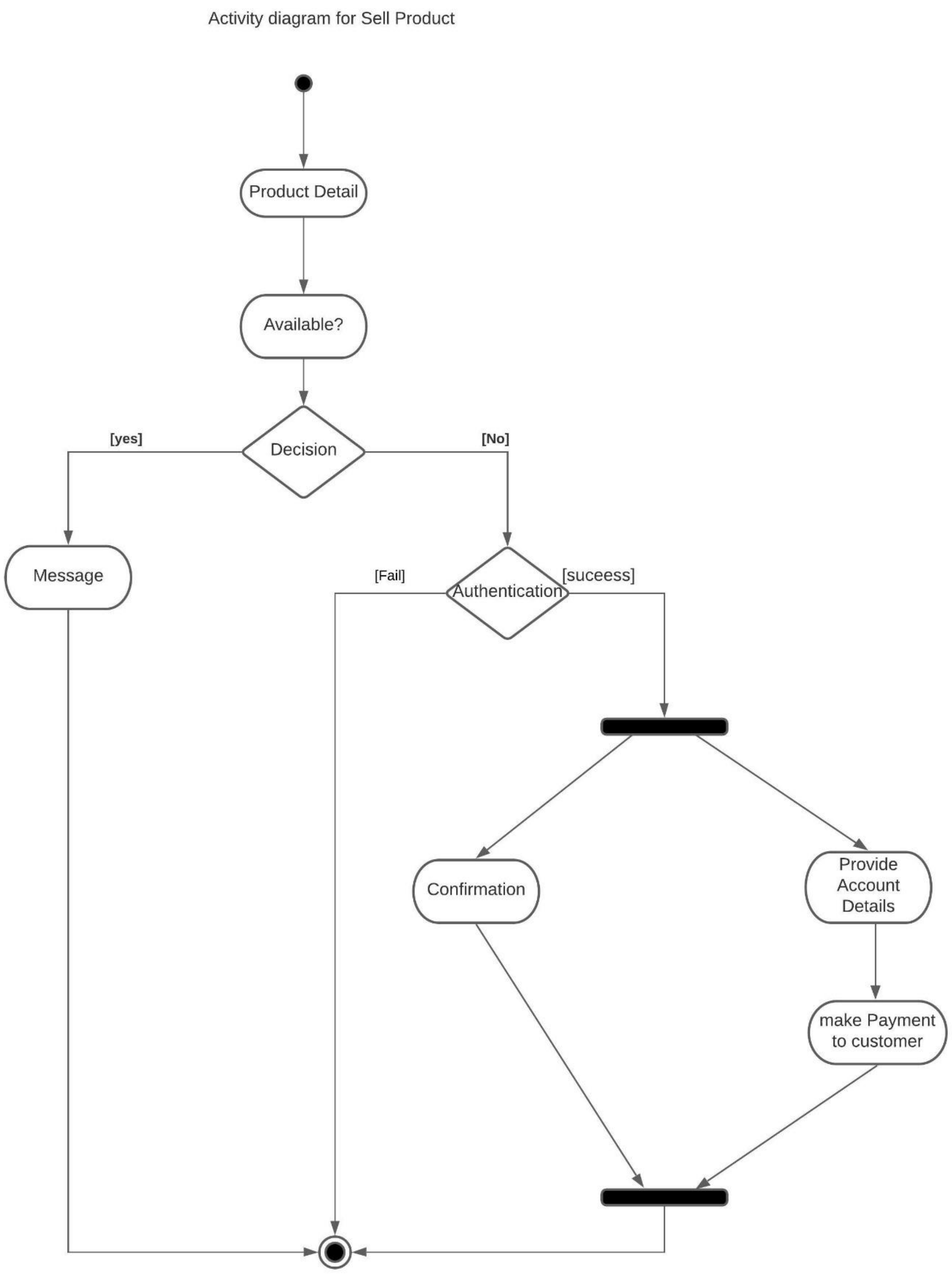


4.5 Activity Diagram

4.5.1 Activity Diagram For Buy Product Use Case



4.5.2 Activity Diagram For Sell Product Use Case



5. Implementation Details

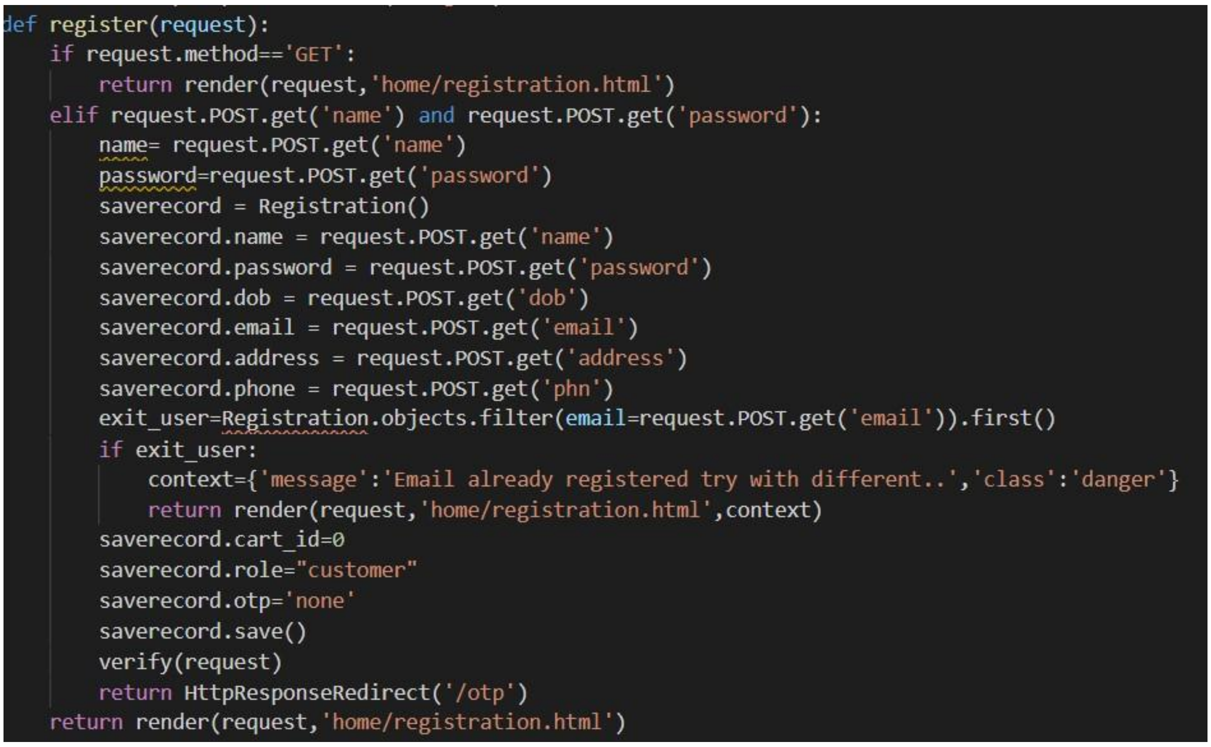
5.1 Modules
The System consist of 4 various modules namely

1) Manage Account
2) Manage Product
3) Manage Order
4) Manage Payment

These modules are implements several methods to perform major functionalities. Implementation is done using Django and MySQL.

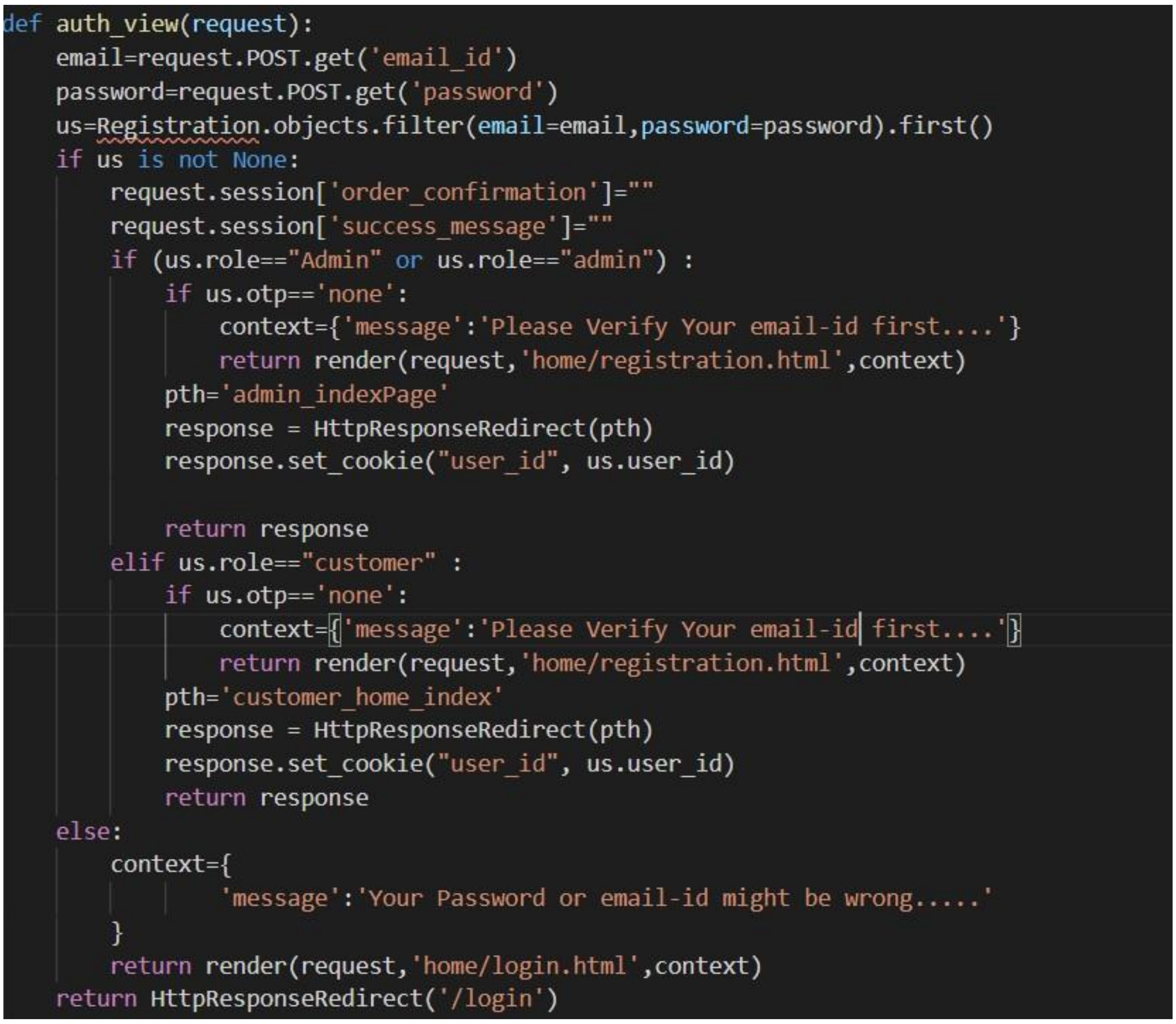
5.2 Implementation Of modules
Manage Account

This module contain all methods related to account. It contains Signup, Authorization, Authentication ,Login,Update Profile, Reset Password .



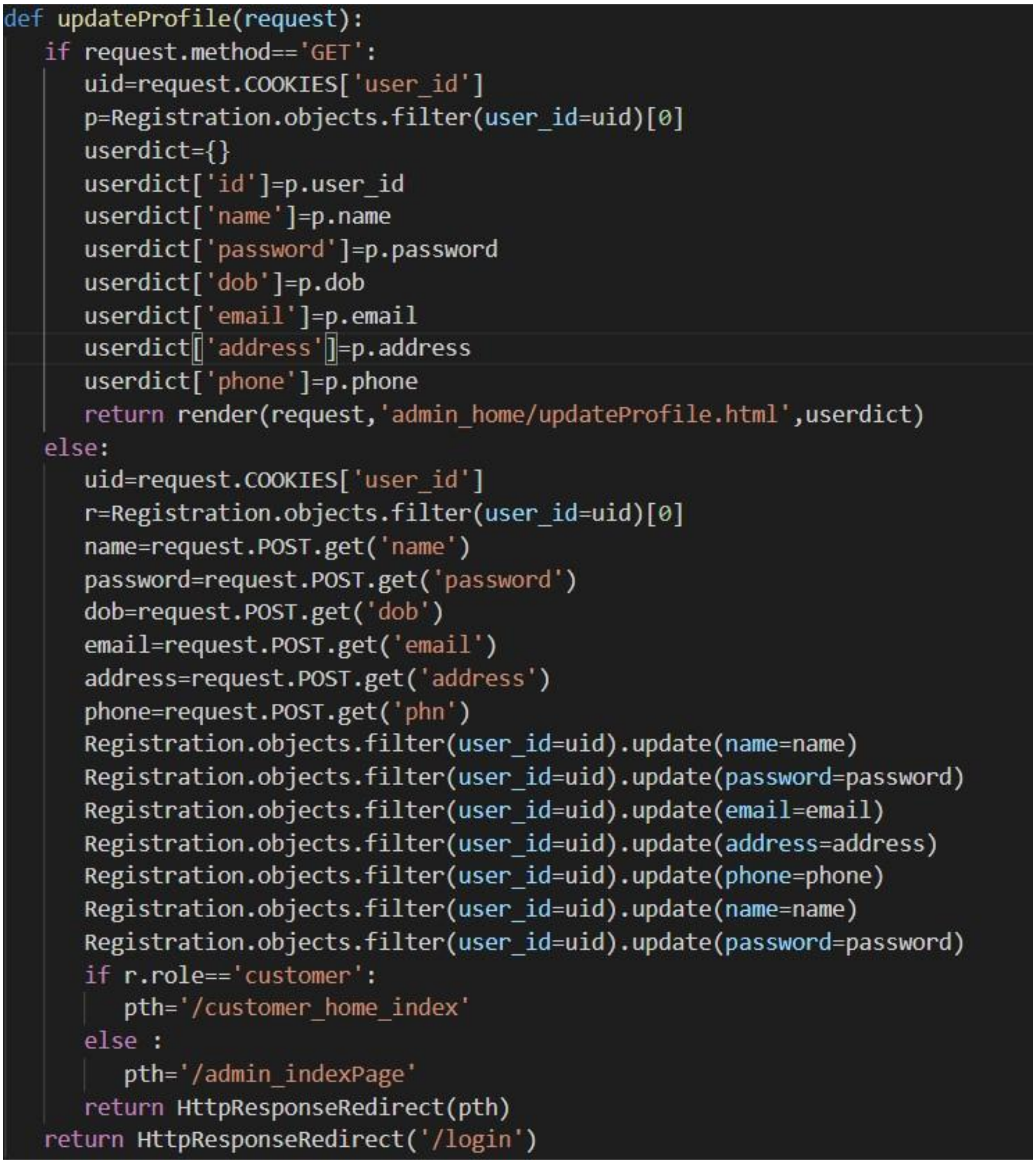
```
def register(request):
    if request.method=='GET':
        return render(request,'home/registration.html')
    elif request.POST.get('name') and request.POST.get('password'):
        name= request.POST.get('name')
        password=request.POST.get('password')
        saverecord = Registration()
        saverecord.name = request.POST.get('name')
        saverecord.password = request.POST.get('password')
        saverecord.dob = request.POST.get('dob')
        saverecord.email = request.POST.get('email')
        saverecord.address = request.POST.get('address')
        saverecord.phone = request.POST.get('phn')
        exit_user=Registration.objects.filter(email=request.POST.get('email')).first()
        if exit_user:
            context={'message':'Email already registered try with different..','class':'danger'}
            return render(request,'home/registration.html',context)
        saverecord.cart_id=0
        saverecord.role="customer"
        saverecord.otp='none'
        saverecord.save()
        verify(request)
        return HttpResponseRedirect('/otp')
    return render(request,'home/registration.html')
```

Implementation of Sign Up functionality



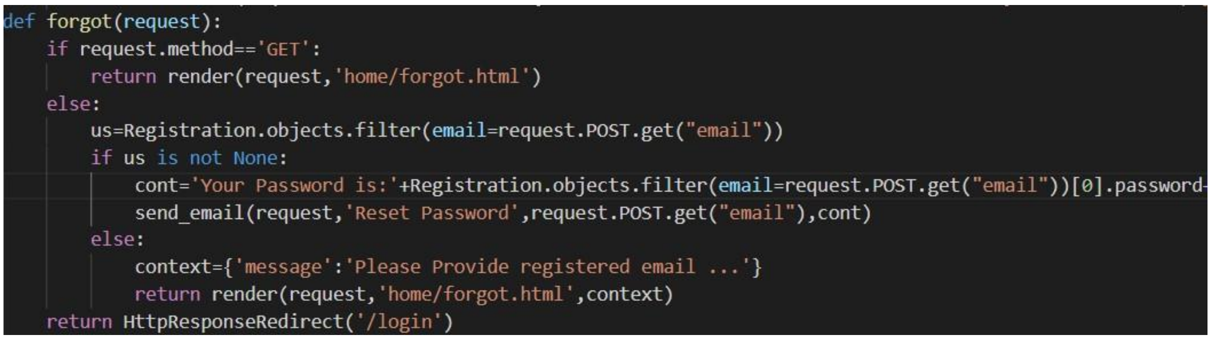
```python
def auth_view(request):
    email=request.POST.get('email_id')
    password=request.POST.get('password')
    us=Registration.objects.filter(email=email,password=password).first()
    if us is not None:
        request.session['order_confirmation']=""
        request.session['success_message']=""
        if (us.role=="Admin" or us.role=="admin") :
            if us.otp=='none':
                context={'message':'Please Verify Your email-id first....'}
                return render(request,'home/registration.html',context)
            pth='admin_indexPage'
            response = HttpResponseRedirect(pth)
            response.set_cookie("user_id", us.user_id)

            return response
        elif us.role=="customer" :
            if us.otp=='none':
                context={'message':'Please Verify Your email-id first....'}
                return render(request,'home/registration.html',context)
            pth='customer_home_index'
            response = HttpResponseRedirect(pth)
            response.set_cookie("user_id", us.user_id)
            return response
    else:
        context={
                'message':'Your Password or email-id might be wrong.....'
        }
        return render(request,'home/login.html',context)
    return HttpResponseRedirect('/login')
```

Implementation of Authentication and Login



```python
def updateProfile(request):
    if request.method=='GET':
        uid=request.COOKIES['user_id']
        p=Registration.objects.filter(user_id=uid)[0]
        userdict={}
        userdict['id']=p.user_id
        userdict['name']=p.name
        userdict['password']=p.password
        userdict['dob']=p.dob
        userdict['email']=p.email
        userdict['address']=p.address
        userdict['phone']=p.phone
        return render(request,'admin_home/updateProfile.html',userdict)
    else:
        uid=request.COOKIES['user_id']
        r=Registration.objects.filter(user_id=uid)[0]
        name=request.POST.get('name')
        password=request.POST.get('password')
        dob=request.POST.get('dob')
        email=request.POST.get('email')
        address=request.POST.get('address')
        phone=request.POST.get('phn')
        Registration.objects.filter(user_id=uid).update(name=name)
        Registration.objects.filter(user_id=uid).update(password=password)
        Registration.objects.filter(user_id=uid).update(email=email)
        Registration.objects.filter(user_id=uid).update(address=address)
        Registration.objects.filter(user_id=uid).update(phone=phone)
        Registration.objects.filter(user_id=uid).update(name=name)
        Registration.objects.filter(user_id=uid).update(password=password)
        if r.role=='customer':
            pth='/customer_home_index'
        else :
            pth='/admin_indexPage'
        return HttpResponseRedirect(pth)
    return HttpResponseRedirect('/login')
```

Implementation of Update Profile

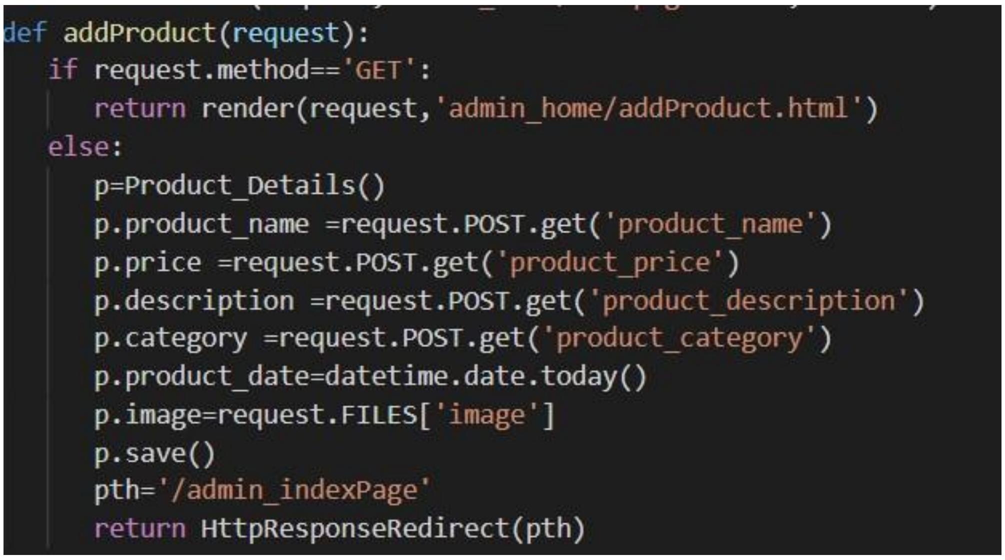


```python
def forgot(request):
    if request.method=='GET':
        return render(request,'home/forgot.html')
    else:
        us=Registration.objects.filter(email=request.POST.get("email"))
        if us is not None:
            cont='Your Password is:'+Registration.objects.filter(email=request.POST.get("email"))[0].password-
            send_email(request,'Reset Password',request.POST.get("email"),cont)
        else:
            context={'message':'Please Provide registered email ...'}
            return render(request,'home/forgot.html',context)
    return HttpResponseRedirect('/login')
```

Implementation of Forgot Password

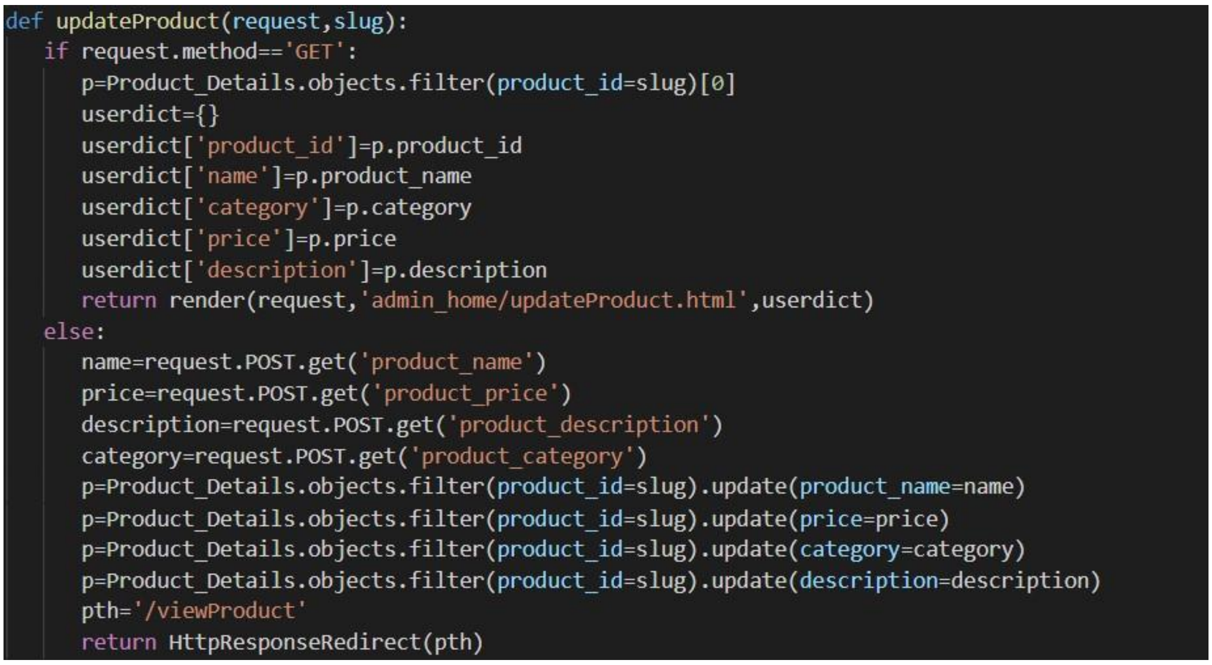
Manage Product

This module contain every method which is deals with Product data like Add Product, Update Product, Sell Product, Delete Product , View Product etc…



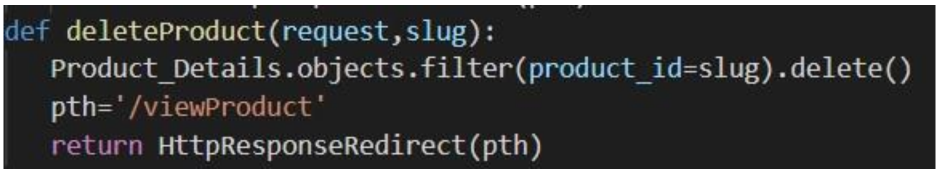
```
def addProduct(request):
    if request.method=='GET':
        return render(request,'admin_home/addProduct.html')
    else:
        p=Product_Details()
        p.product_name =request.POST.get('product_name')
        p.price =request.POST.get('product_price')
        p.description =request.POST.get('product_description')
        p.category =request.POST.get('product_category')
        p.product_date=datetime.date.today()
        p.image=request.FILES['image']
        p.save()
        pth='/admin_indexPage'
        return HttpResponseRedirect(pth)
```

Implementation of Add Product



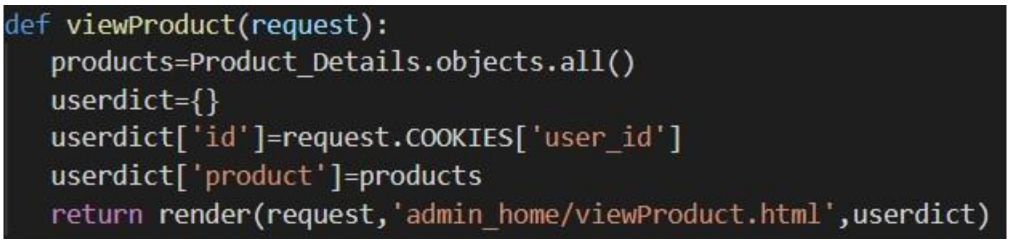
```
def updateProduct(request,slug):
    if request.method=='GET':
        p=Product_Details.objects.filter(product_id=slug)[0]
        userdict={}
        userdict['product_id']=p.product_id
        userdict['name']=p.product_name
        userdict['category']=p.category
        userdict['price']=p.price
        userdict['description']=p.description
        return render(request,'admin_home/updateProduct.html',userdict)
    else:
        name=request.POST.get('product_name')
        price=request.POST.get('product_price')
        description=request.POST.get('product_description')
        category=request.POST.get('product_category')
        p=Product_Details.objects.filter(product_id=slug).update(product_name=name)
        p=Product_Details.objects.filter(product_id=slug).update(price=price)
        p=Product_Details.objects.filter(product_id=slug).update(category=category)
        p=Product_Details.objects.filter(product_id=slug).update(description=description)
        pth='/viewProduct'
        return HttpResponseRedirect(pth)
```

Implementation of Update Product



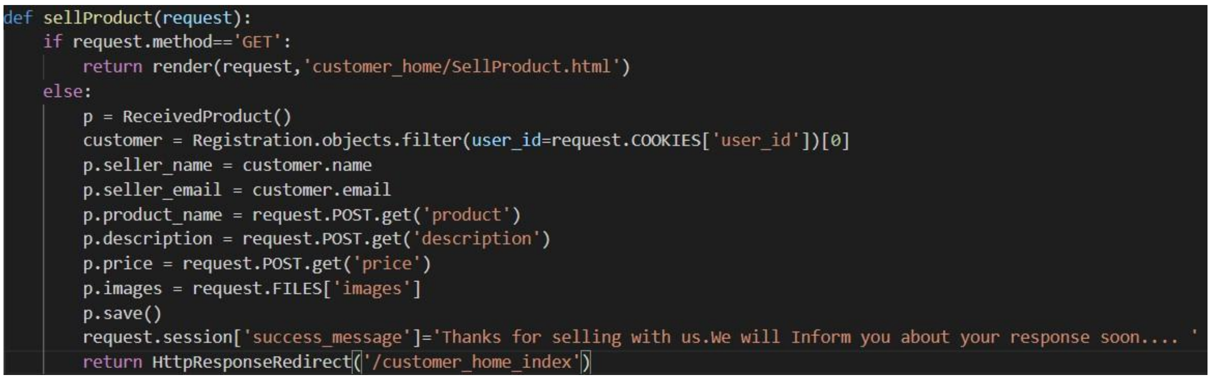
```python
def deleteProduct(request,slug):
    Product_Details.objects.filter(product_id=slug).delete()
    pth='/viewProduct'
    return HttpResponseRedirect(pth)
```

Implementation of Delete Product



```python
def viewProduct(request):
    products=Product_Details.objects.all()
    userdict={}
    userdict['id']=request.COOKIES['user_id']
    userdict['product']=products
    return render(request,'admin_home/viewProduct.html',userdict)
```

Implementation of View Product

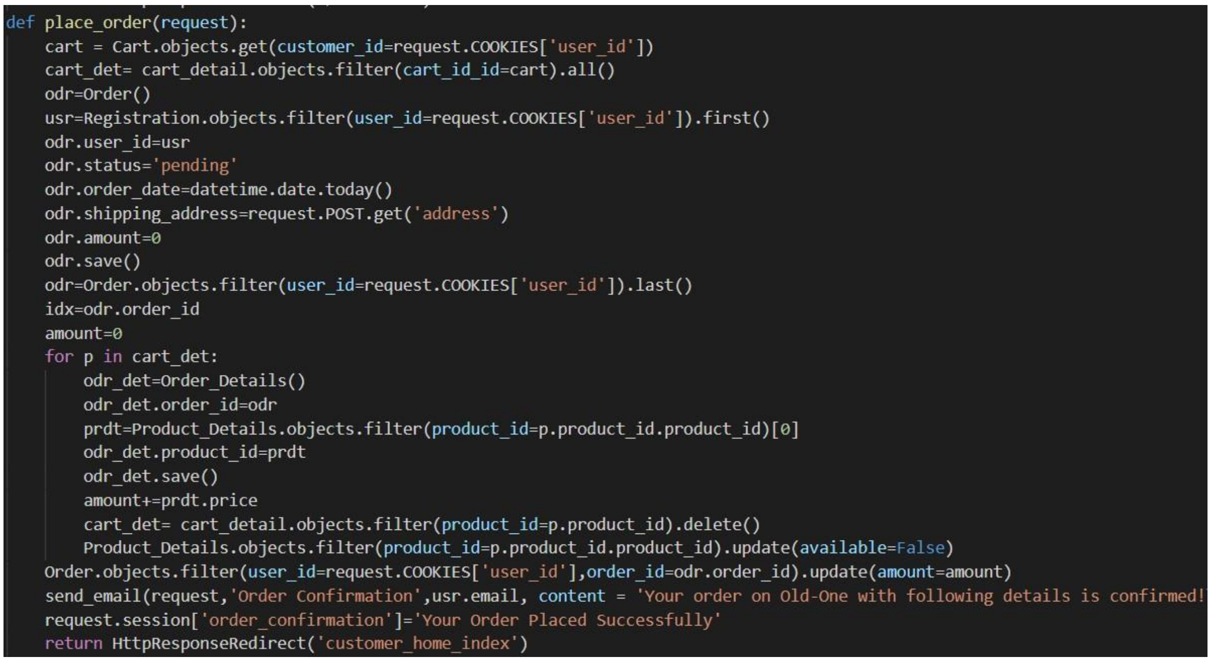


```python
def sellProduct(request):
    if request.method=='GET':
        return render(request,'customer_home/SellProduct.html')
    else:
        p = ReceivedProduct()
        customer = Registration.objects.filter(user_id=request.COOKIES['user_id'])[0]
        p.seller_name = customer.name
        p.seller_email = customer.email
        p.product_name = request.POST.get('product')
        p.description = request.POST.get('description')
        p.price = request.POST.get('price')
        p.images = request.FILES['images']
        p.save()
        request.session['success_message']='Thanks for selling with us.We will Inform you about your response soon.... '
        return HttpResponseRedirect('/customer_home_index')
```

Implementation of Sell Product(customer)

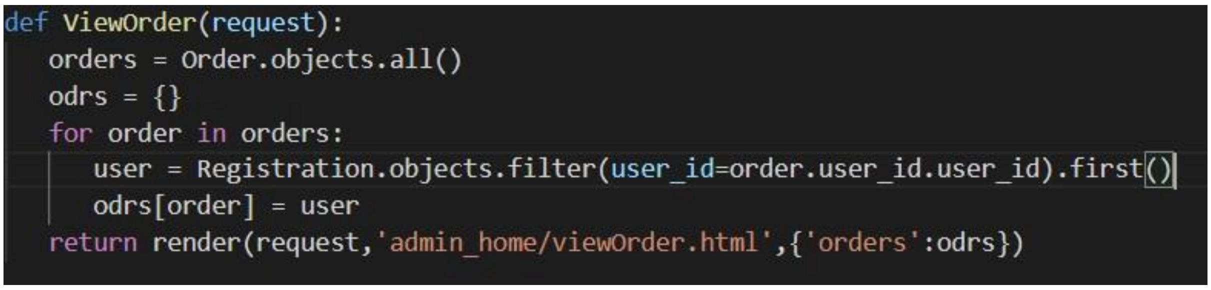
Manage Order

This module handle everything regarding Orders Using Order data store like For Admin View Order(Abstract), View Order Details . admin can modify Order status like from pending to done/success. Customer can Place Order, view own order history.



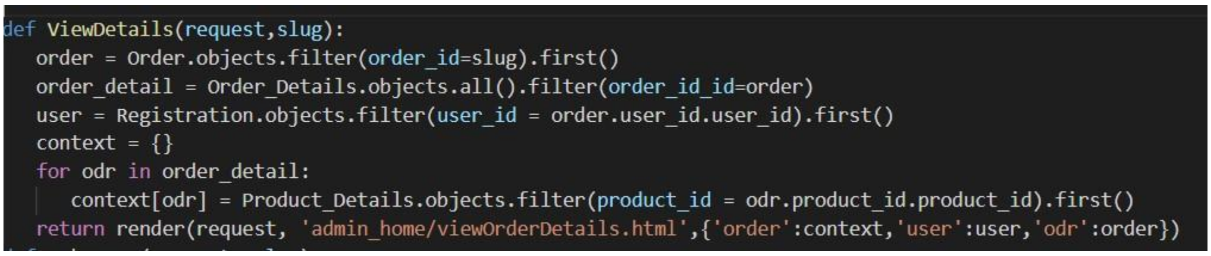
```
def place_order(request):
    cart = Cart.objects.get(customer_id=request.COOKIES['user_id'])
    cart_det= cart_detail.objects.filter(cart_id_id=cart).all()
    odr=Order()
    usr=Registration.objects.filter(user_id=request.COOKIES['user_id']).first()
    odr.user_id=usr
    odr.status='pending'
    odr.order_date=datetime.date.today()
    odr.shipping_address=request.POST.get('address')
    odr.amount=0
    odr.save()
    odr=Order.objects.filter(user_id=request.COOKIES['user_id']).last()
    idx=odr.order_id
    amount=0
    for p in cart_det:
        odr_det=Order_Details()
        odr_det.order_id=odr
        prdt=Product_Details.objects.filter(product_id=p.product_id.product_id)[0]
        odr_det.product_id=prdt
        odr_det.save()
        amount+=prdt.price
        cart_det= cart_detail.objects.filter(product_id=p.product_id).delete()
        Product_Details.objects.filter(product_id=p.product_id.product_id).update(available=False)
    Order.objects.filter(user_id=request.COOKIES['user_id'],order_id=odr.order_id).update(amount=amount)
    send_email(request,'Order Confirmation',usr.email, content = 'Your order on Old-One with following details is confirmed!
    request.session['order_confirmation']='Your Order Placed Successfully'
    return HttpResponseRedirect('customer_home_index')
```

Implementation of Place Order



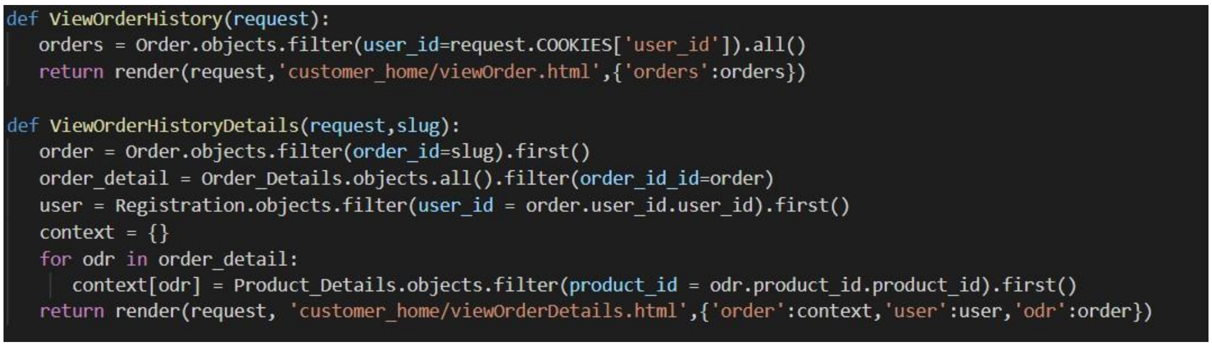
```
def ViewOrder(request):
    orders = Order.objects.all()
    odrs = {}
    for order in orders:
        user = Registration.objects.filter(user_id=order.user_id.user_id).first()
        odrs[order] = user
    return render(request,'admin_home/viewOrder.html',{'orders':odrs})
```

Implementation of View Order(Admin)



```
def ViewDetails(request,slug):
    order = Order.objects.filter(order_id=slug).first()
    order_detail = Order_Details.objects.all().filter(order_id_id=order)
    user = Registration.objects.filter(user_id = order.user_id.user_id).first()
    context = {}
    for odr in order_detail:
        context[odr] = Product_Details.objects.filter(product_id = odr.product_id.product_id).first()
    return render(request, 'admin_home/viewOrderDetails.html',{'order':context,'user':user,'odr':order})
```

Implementation of View Order Details(Admin)

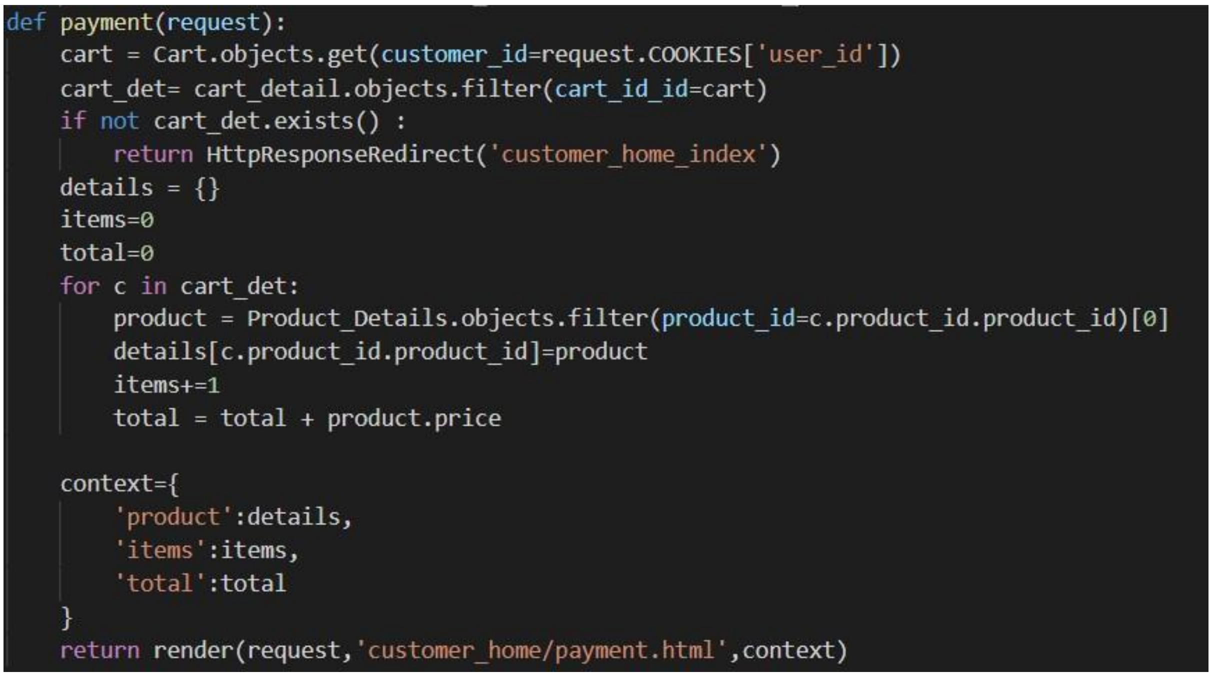


```
def ViewOrderHistory(request):
    orders = Order.objects.filter(user_id=request.COOKIES['user_id']).all()
    return render(request,'customer_home/viewOrder.html',{'orders':orders})

def ViewOrderHistoryDetails(request,slug):
    order = Order.objects.filter(order_id=slug).first()
    order_detail = Order_Details.objects.all().filter(order_id_id=order)
    user = Registration.objects.filter(user_id = order.user_id.user_id).first()
    context = {}
    for odr in order_detail:
        context[odr] = Product_Details.objects.filter(product_id = odr.product_id.product_id).first()
    return render(request, 'customer_home/viewOrderDetails.html',{'order':context,'user':user,'odr':order})
```

Implementation of View Order and View Order Details(Customer)

Manage Payment

This module implements payment method. Customer Can payment using any method like ‘COD’, ’Gpay’, ’Debit card’ etc..



```
def payment(request):
    cart = Cart.objects.get(customer_id=request.COOKIES['user_id'])
    cart_det= cart_detail.objects.filter(cart_id_id=cart)
    if not cart_det.exists() :
        return HttpResponseRedirect('customer_home_index')
    details = {}
    items=0
    total=0
    for c in cart_det:
        product = Product_Details.objects.filter(product_id=c.product_id.product_id)[0]
        details[c.product_id.product_id]=product
        items+=1
        total = total + product.price

    context={
        'product':details,
        'items':items,
        'total':total
    }
    return render(request,'customer_home/payment.html',context)
```

Implementation of Payment

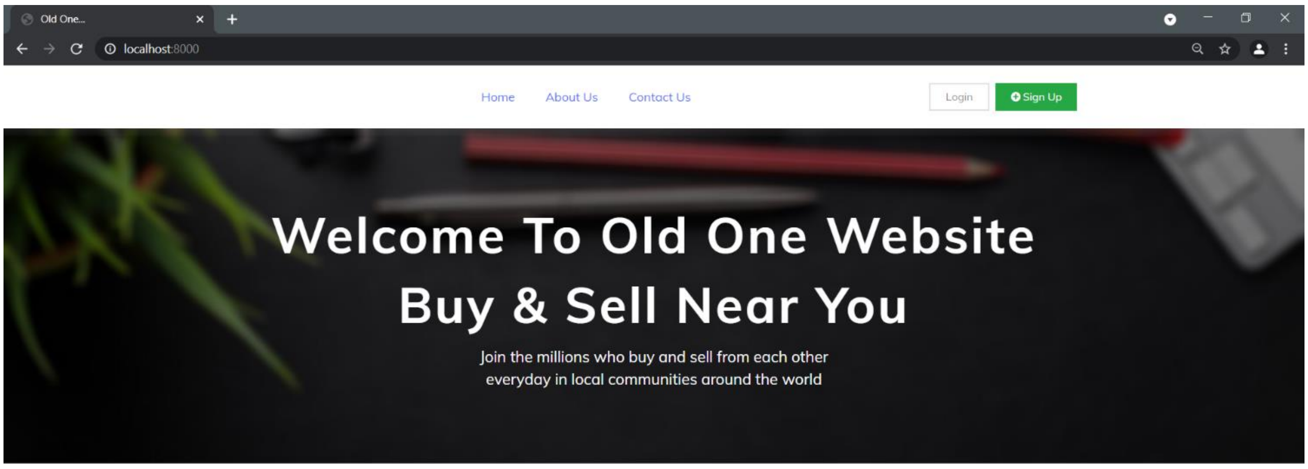
6. Testing

❖ Testing Method ::Manual

Sr. No.	Test Case Objective	Expected Result	Actual Result	Status
1.	Registration with already used email	System should display message 'Email is already registered'	Pop up message:- 'Email is already registered'	Success
2.	Registration with correct Credentials	System should waiting for OTP verification.	Asked for OTP which sent to user's email	Success
3.	Login with incorrect Credentials	System should given message :-'incorrect email or password'	Pop up message:- 'Email and password must be correct..'	Success
4.	Login with correct credentials	System should Redirect to home according to user's role(customer/admin)	Redirect to home page of user	Success
5.	Forgot Password	System should send mail of user's password	User receive mail of password	Success
6.	Sell Product	System should give confirmation like 'your Request is submitted for sell product'	Pop up confirmation message	Success
7.	Place Order	System should give confirmation Order Placed Successfully	Order Confirmation pop up	Success

7. Snap Shots Of System

❖ Home Page Of Website

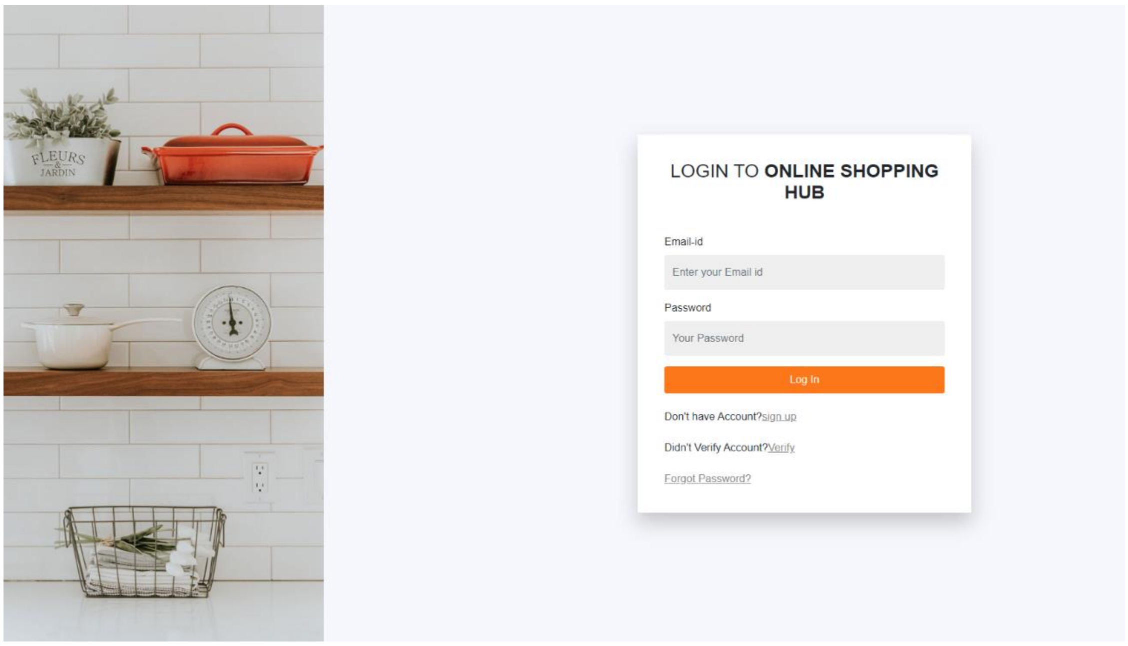


Welcome To Old One Website

Buy & Sell Near You

Join the millions who buy and sell from each other everyday in local communities around the world

❖ Login /Sign Up



LOGIN TO **ONLINE SHOPPING HUB**

Email-id

Enter your Email id

Password

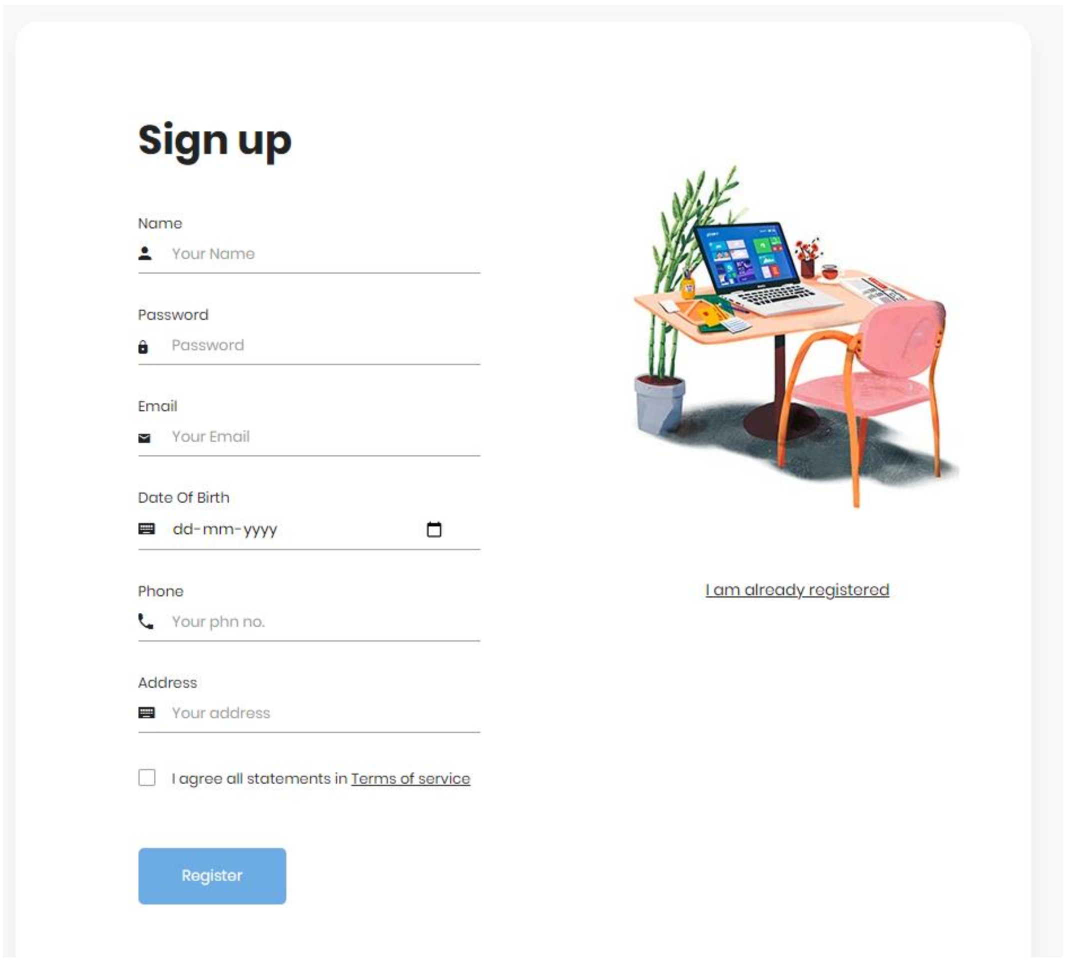
Your Password

Log In

Don't have Account?sign up

Didn't Verify Account?Verify

Forgot Password?



Sign up

Name

Your Name

Password

Password

Email

Your Email

Date Of Birth

dd-mm-yyyy

Phone

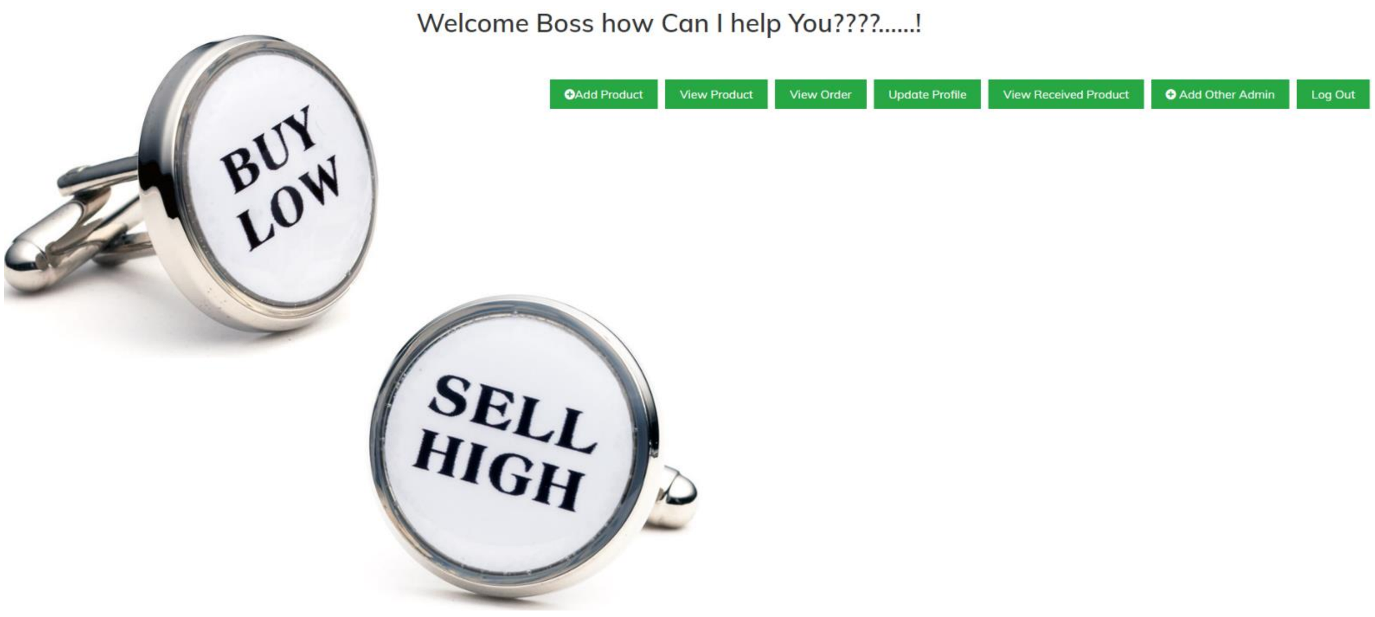
Your phn no.

Address

Your address

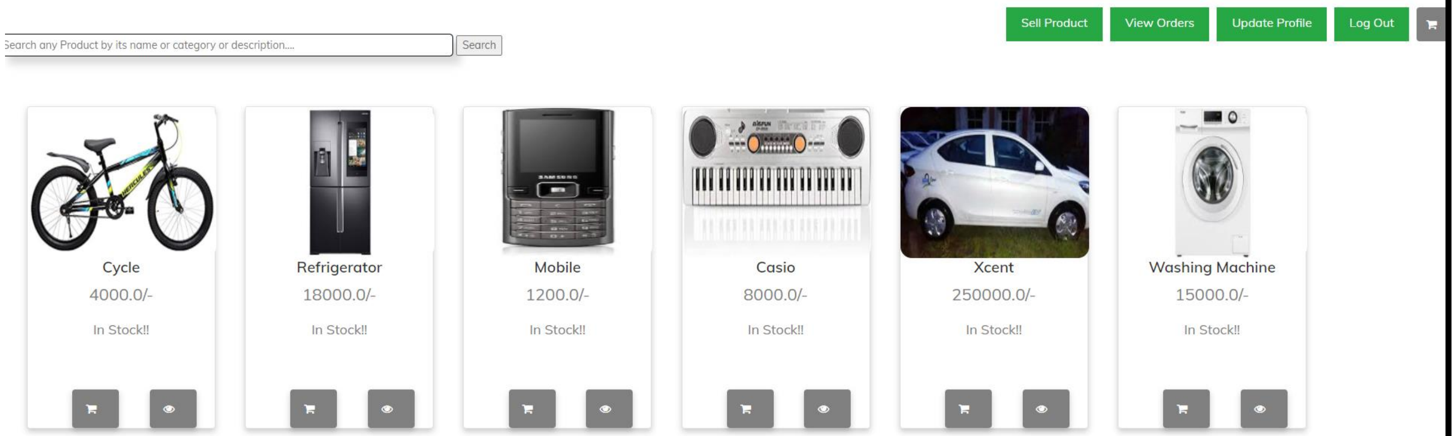
I agree all statements in Terms of service

❖ Admin Home Page



Welcome Boss how Can I help You????......!

❖ Customer Home Page


Sell Product
View Orders
Update Profile
Log Out
Search any Product by its name or category or description....
Search

Cycle
4000.0/-
In Stock!!

Refrigerator
18000.0/-
In Stock!!

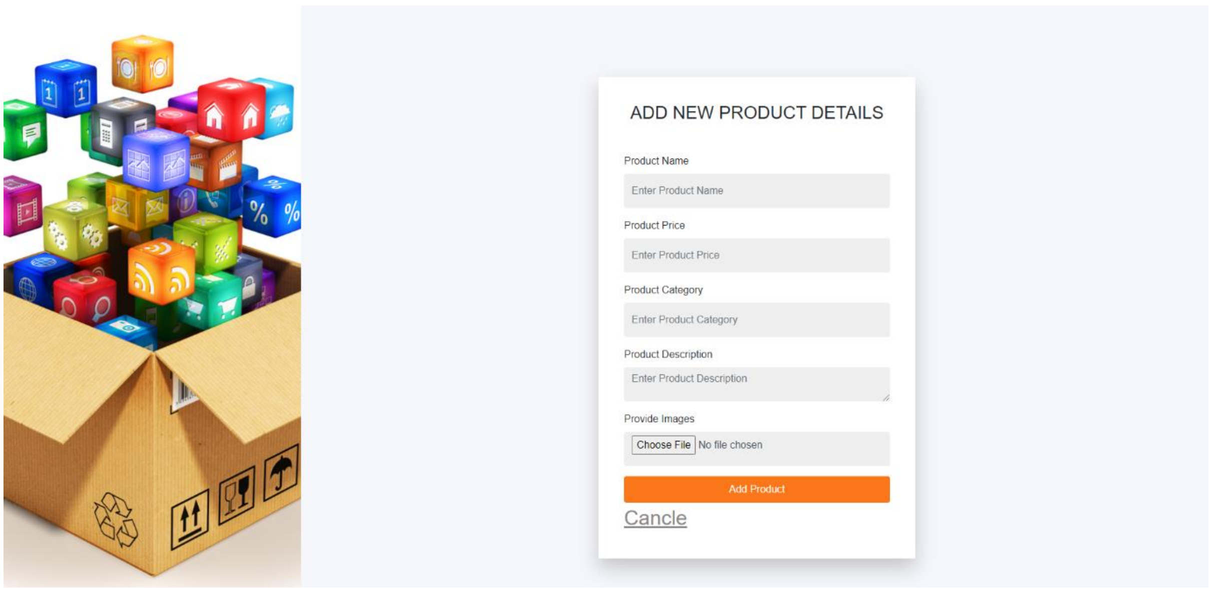
Mobile
1200.0/-
In Stock!!

Casio
8000.0/-
In Stock!!

Xcent
250000.0/-
In Stock!!

Washing Machine
15000.0/-
In Stock!!

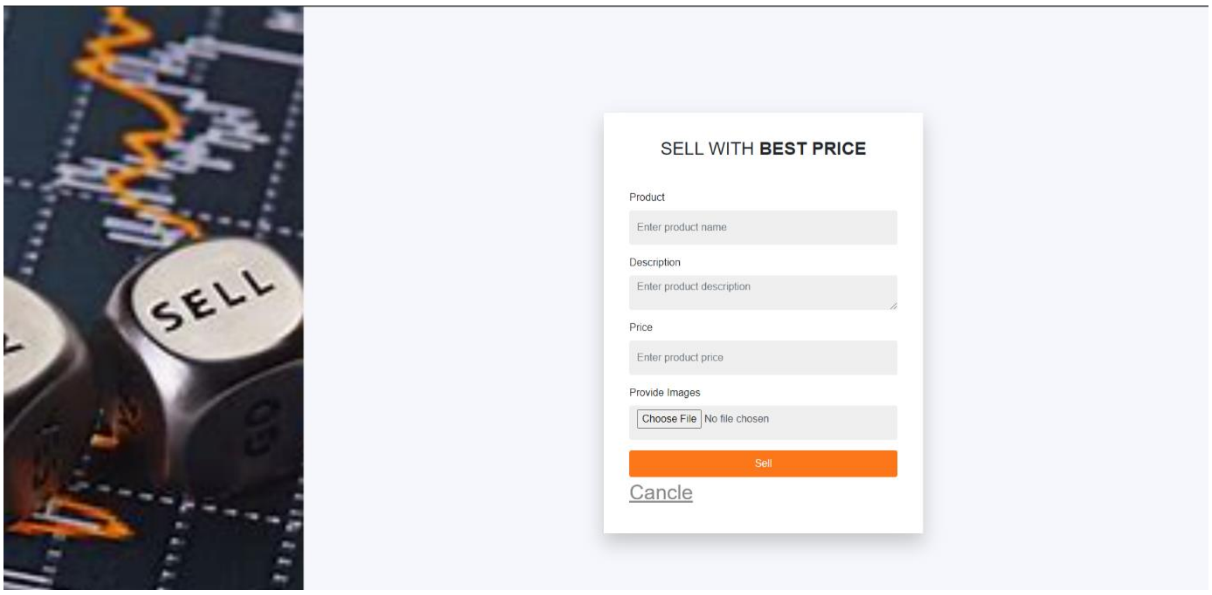
❖ Add Product



ADD NEW PRODUCT DETAILS

Product Name
Enter Product Name
Product Price
Enter Product Price
Product Category
Enter Product Category
Product Description
Enter Product Description
Provide Images
Choose File No file chosen

❖ Sell Product(customer)


SELL WITH **BEST PRICE**

Product

Enter product name

Description

Enter product description

Price

Enter product price

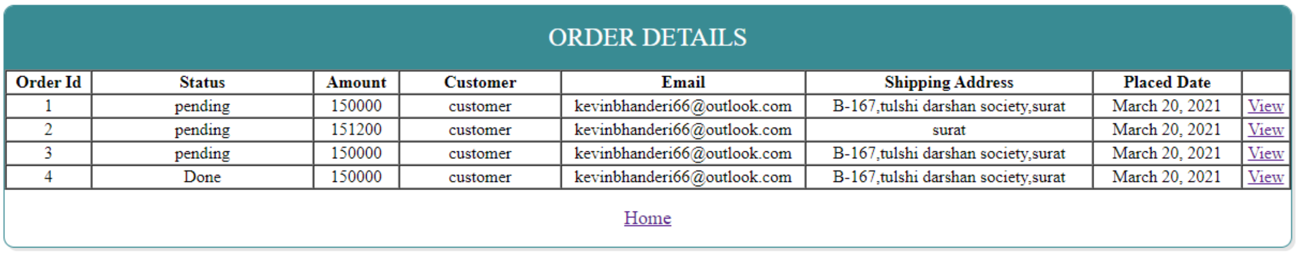
Provide Images

Choose File No file chosen

Sell

Cancle

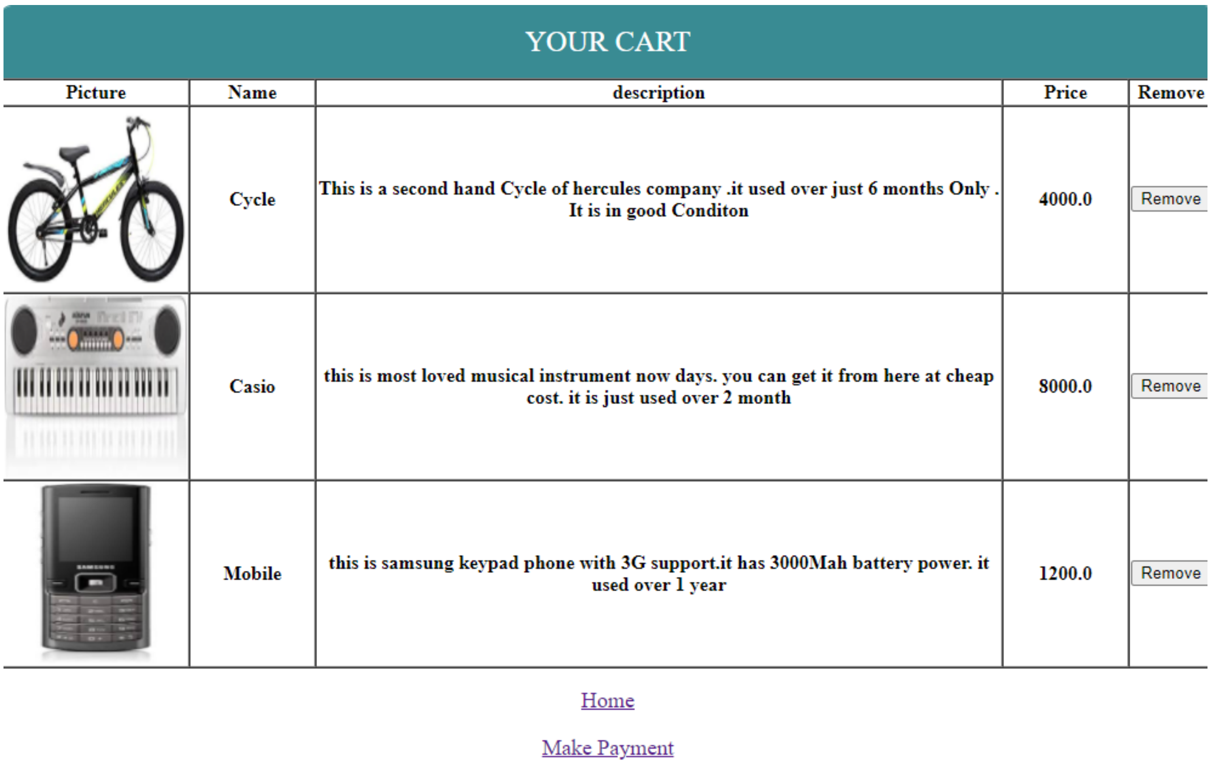
❖ View Orders


ORDER DETAILS

Order Id	Status	Amount	Customer	Email	Shipping Address	Placed Date	
1	pending	150000	customer	kevinbhanderi66@outlook.com	B-167,tulshi darshan society,surat	March 20, 2021	View
2	pending	151200	customer	kevinbhanderi66@outlook.com	surat	March 20, 2021	View
3	pending	150000	customer	kevinbhanderi66@outlook.com	B-167,tulshi darshan society,surat	March 20, 2021	View
4	Done	150000	customer	kevinbhanderi66@outlook.com	B-167,tulshi darshan society,surat	March 20, 2021	View

Home

❖ Cart



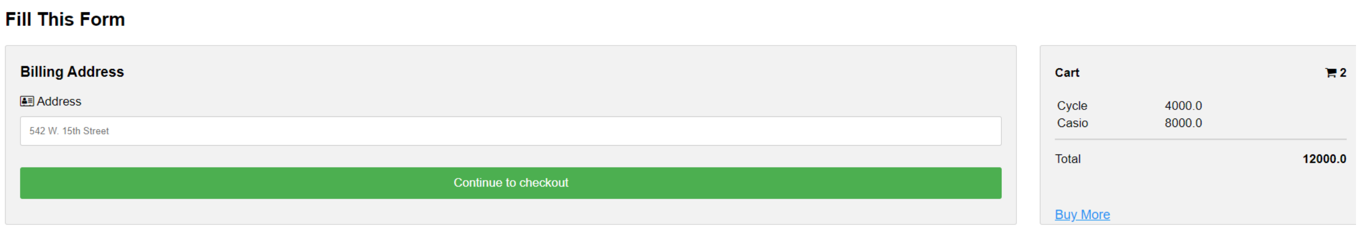
YOUR CART

Picture	Name	description	Price	Remove
	Cycle	This is a second hand Cycle of hercules company .it used over just 6 months Only . It is in good Conditon	4000.0	Remove
	Casio	this is most loved musical instrument now days. you can get it from here at cheap cost. it is just used over 2 month	8000.0	Remove
	Mobile	this is samsung keypad phone with 3G support.it has 3000Mah battery power. it used over 1 year	1200.0	Remove

Home

Make Payment

❖ Payment



Fill This Form

Billing Address

Address

542 W. 15th Street

Continue to checkout

Cart 2

Cycle	4000.0
Casio	8000.0
Total	12000.0

Buy More

8. Conclusion

- Hence-forth in this project we have successfully implemented the Admin-side & Customer-side functionality, Admin can add the products or change the product information. Admin can show all the products with their necessary details, admin can also delete the particular product from website. Here two types of customer are there so

  1)Seller
    - Seller can sell the products with product's details, after based on admin response the customer either will get confirmation or rejection.

  2)Buyer
    - Buyer can buy the all the products which are in stock right now, otherwise they see the message that, 'This Product is not in stock'. Buyer can also use the cart for buying more product and make payment for all once.
- Once Customer Places Order Confirmation mail would be sent to the customer.
- Basically, this website works perfectly fine by both customer and admin's view.

9.Limitations And Future Extensions

9.1 Limitations

1. For making payment in this site, we have used only COD(Cash On Delivery), so right now no any other method for payment.
2. Right now customer can only place the order, customer will not be able to cancel the order.
3. As per now Admin can not see the report(yearly/monthly) for revenue generated by the company and website's condition means how many sells per day or per month or per year.
4. Seller will not be able to see how many products he/she has sold to the system.

9.2 Future Extensions

To take over the limitations we are planning this future extension in our system.

1) Good and better User Interface.
2) Implements Payment Gateway.
3) Implements Method to display statistics data.
4) Customer will be able to cancel order in future.
5) By modifying this system, system can be used broader scale.
6) Customer can filter the product according to price, condition.

10.Bibliography

- Github.com
- Stackoverflow.com
- docs.djangoproject.com