Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer Engineering

B. Tech. CE Semester – IV

Project Title : E-Commerce Website

By :

1] Harpritsinh Yadav , roll no:CE148,ID-19CEUOS129

2] Bhagatsinh Zankat , roll no:CE149,ID-19CEUOG054

Contents :

1.Abstract

E-commerce where a normal business integrated with technology is done globally by buying and selling goods on the internet. By doing business with E-commerce websites, one can expand their market margins to global horizons or squeeze them into highly focused market segments. In the past 15 years, E-commerce businesses have touched the skies in terms of growth.

Setting up a business with E-commerce has a lesser cost than setting up it in an offline market. E-commerce businesses can be operated from anywhere. Also, there are no restrictions on opening-closing times. Easy access to new customers as anyone from anywhere can access the online store. E-commerce can study consumer behaviour from their usage and plan their development strategies in accordance.

People of this developing era can now do shopping at the fingertips while sitting on their couch with reduced physical efforts. Customers get a wide variety of products to choose from. Moreover, it is much convenient and safe as we can see from this **COVID-19** pandemic period.

2.Introduction

“E-commerce Website” is a place where customers can shop any kind of product from their devices which they shop from offline markets like clothing,electronics,groceries etc.To satisfy shopping experience of customers we have implemented a number of features.

Features :

1. First and foremost it’s a completely dynamic website
2. New category of products can be added and removed as well.Products of that category can be added or removed.
3. Customers can add the product they wish to buy into their cart,increase quantity or decrease and remove as well.
4. Anonymous users can visit our website but they must be logged in to checkout whatever products they have in their cart.
5. Customers can apply a valid discount coupoun that is active on our website. This discount coupouns are in form of “Promo codes” which can be added or removed by admin.
6. Customers can see reviews of particular products by other users and give one review as well.
7. Customers finally place order and get delivery and order details.

2.1 Tools/Technologies :

Technologies/Languages :

- ❖ Python
- ❖ Django Web Framework
- ❖ MySQL Database
- ❖ HTML
- ❖ CSS
- ❖ Bootstrap
- ❖ Javascript

Tools :

- ❖ Visual Studio Code
- ❖ Git/Github
- ❖ XAMPP control panel

Platform :

- ❖ Local Server

3.Software Requirement Specification

End Users :

1. Admin
2. Buyers
3. Suppliers

Functional Requirements :

R1 Registration

R.1.1] Signing up a user

- ➢ The users who are using the website for the first time are added to the database

Input :

- Name,Email address,phone no,address,username,password.

Output :

- User is redirected to the homepage and user info is added into the database.

R.1.2] Login by user

- ➢ Existing user can login with his/her credentials.Authorization is done by admin.

Input :

- Username/Password.

Output :

- Opens up homepage of the website

R2. Manage Products

R.2.1] Add Product

- Admin can add a new product on the website.
  Input :
  - Product details,Image,Price.

  Output :
  - Message "Product added successfully"

R.2.2] Remove Product

- Admin can remove an existing product from the website.
  Input :
  -Product code of the item to removed.
  Output :
  -Product will be remove from the database and confirmation message will be output to screen.

R.2.3] Modify Product

- Admin can modify details of an existing product on the website.
  Input :
  - Product code of the item whose details are to be modified.
  Output :
  -Modifications are performed on the database with a confirmation message.

R3 Manage Suppliers and supplies

R.3.1] View Stocks

- Admin can keep track of stocks.
  Input :
  - Click on stocks of a particular product.

  Output :
  - "Quantity available for product 'p' is 'x' ".

R.3.2] Order for resupply:

- Admin can order for more stocks of a specific product.

  Input :
  - Specify product code and required quantity.

  Output :
  - Notifies supplier with an email regarding quantities

R4 Search and View Product

R.4.1] Navigation Bar

- Customer can navigate through different categories of products for the category item which he is looking for.

  Input :
  - Hover over the categories and follow the dropdown menus.

  Output :
  - Loads multiple dropdown menus for each category and open up products of desired category.

R.4.2] View Product

- Customer can view details of the product

  Input :

  Click on the product.

  Output :

  Shows all the details of product like description, price, availability, also shows how many people have bought it already.

R5 Manage Cart

R.5.1] Add product

- Customer can add product to the cart.

  Input :

  Click on Add to Cart button.

Output :
“Item added to Cart”

R.5.2] Remove product

- Customer can remove the product from the cart.
  Input :
  Click on Remove to Cart button.
  Output :
  “Item Removed from cart”

R.5.3] Specify Quantity

- User can specify the quantity to be added into cart.
  Input :
  - Increase or decrease with buttons or specify quantity as text.

  Output :
  - “Quantity increased/decreased to ‘x’ ”

R.5.4] Checkout

- Proceed to checkout all the products selected in the cart.
  Input :
  - Click on the proceed to checkout button.

  Output :
  - Redirects buyer to another webpage(Of place order).

R6 Place Order

R.6.1] Specify address, pincode

- Here the Buyer needs to specify his address and pincode for delivery.

Input :
- Address and pincode.

Output :
- “Your product will be delivered to address ‘x’ ”.

R.6.2]Specify Phone no.

- ➢ Here the Buyer needs to specify his phone number for receiving details about his order.

Input :

- 10-digit Phone number.
- 1 more optional 10-digit phone number.

Output :

- “You will receive updates on your phone no. +91 ********** ”

R7 Complete Payment

R.7.1] Select Payment Method

- ➢ Buyer is provided with multiple payments method like credit/debit card, Net banking,3rd party payment apps, Cash On Delivery.

Input :

- Select a radio button from multiple options.

Output :

- “Payment method selected”

R.7.2] Specify Promo/Discount Codes.

- ➢ User can avail a pre-decided discount on final amount through a valid promo/discount code.

Input :

- Promo code/discount code.

Output :

- “Hurray!!! Discount of ‘x%’ is availed on final amount.”

R.7.3] Specify Card Details

- ➢ Buyer needs to provide his card details for payment.

Input :

- Card number and CVV.

Output :

- “Payment Completed through Card”

R.7.4] Redirect to 3rd party apps

- ➢ If buyer wants to pay through some other apps

  Input :

  - Select from available apps for payment.

  Output :

  - Redirect to the payment app with final payment amount.

R.7.5] Generate Invoice/Bill

- ➢ Buyer can generate invoice for the products he bought and the payment he made.

  Input :

  - Click on “generate bill” button.

  Output :

  - “Please check your email for invoice.”

R8 Give review and ratings

R.8.1] Give rating

- ➢ Buyer can provide a rating on scale of 1-5 for a product.

Input :

- Slide on a slider from 1-5.

Output :

- “Thank you for rating this product.”

R.8.2]Give review

- ➢ Buyer can provide a detailed review of a product.

Input :

- Review in text form.

Output :

- “Thank You for Reviewing this product.”

R9 Manage Promo Codes

R.9.1] Add Promo code

- Admin adds the promo code.

Input :
- Promo code and discount %.

Output :
- Promo code ‘ x ’ added successfully.

R.9.2] Remove Promo code

- Admin can remove the existing promo code.

Input :
- Promo code to be removed.

Output :
- “Promo code ‘x’ has been removed successfully.”

R10 Manage Homepage

R.10.1] Feature a product

- Admin can feature specific products on homepage during different sales.

Input :
- Product code

Output :
- Featured Product is displayed on Homepage.

R.10.2] Remove a featured product

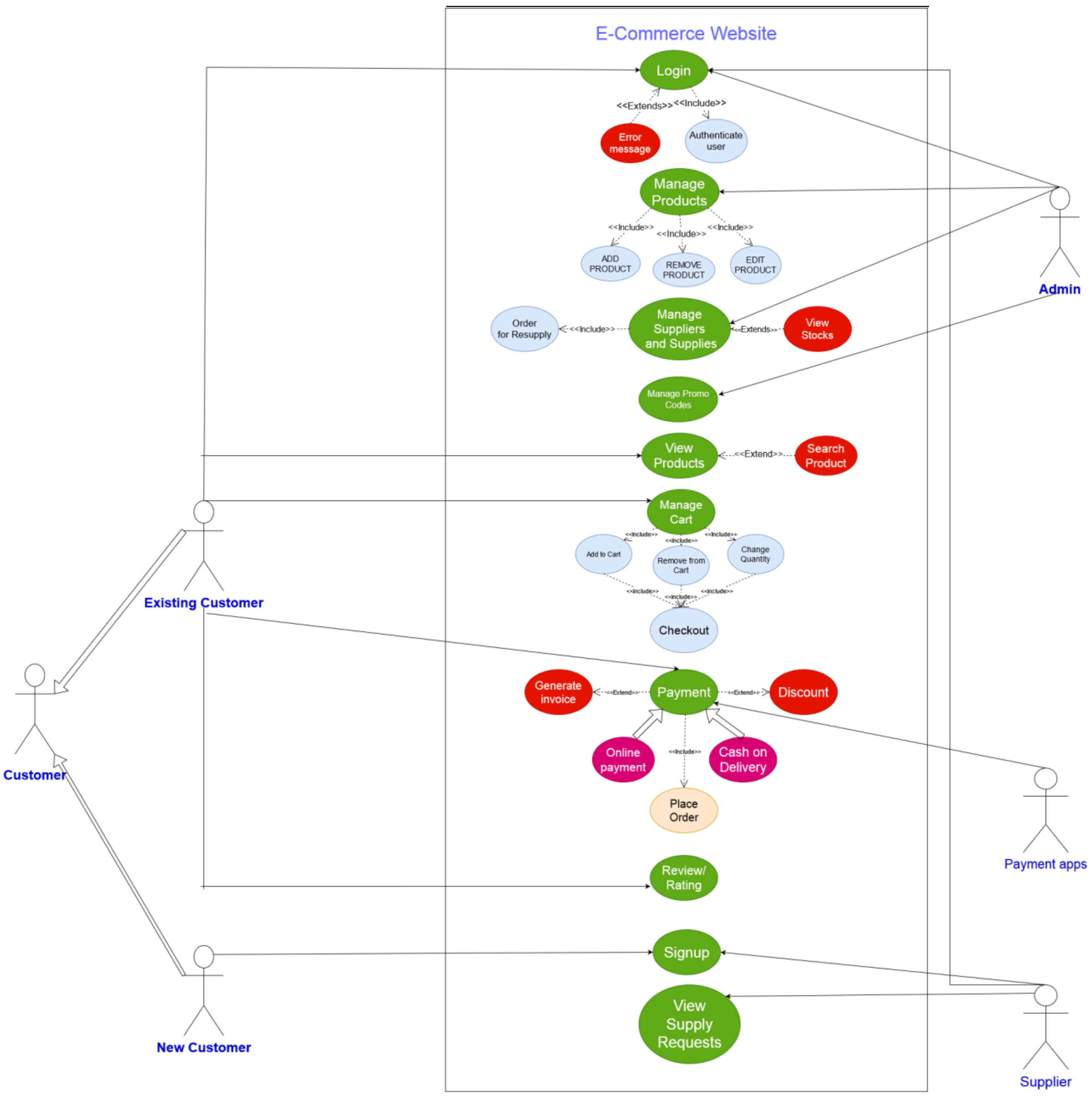
- Admin removes the any featured homepage product.

Input :
- Product code

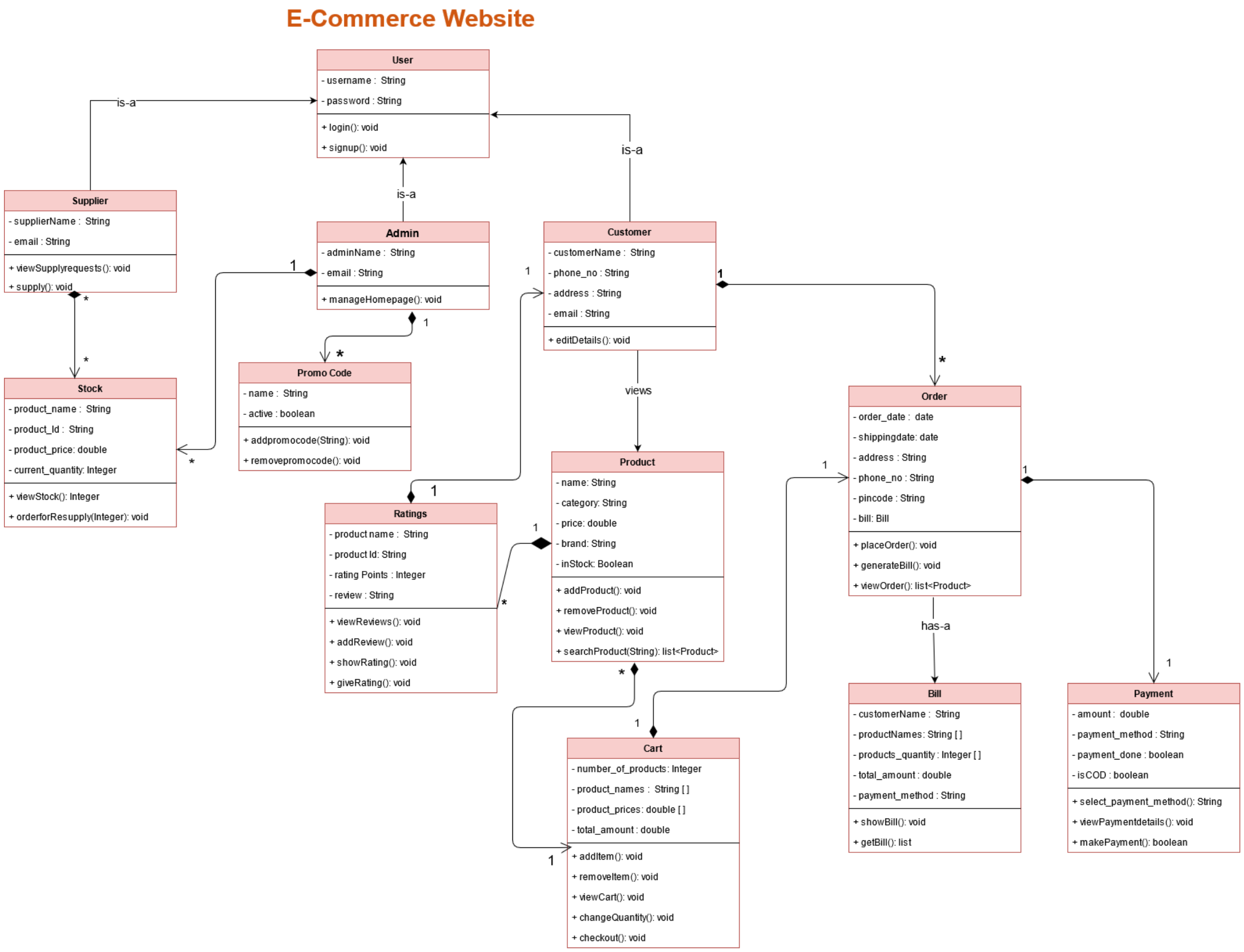
Output :
- “Product removed from homepage.”

4.Design Documents

I.Use Case Diagram

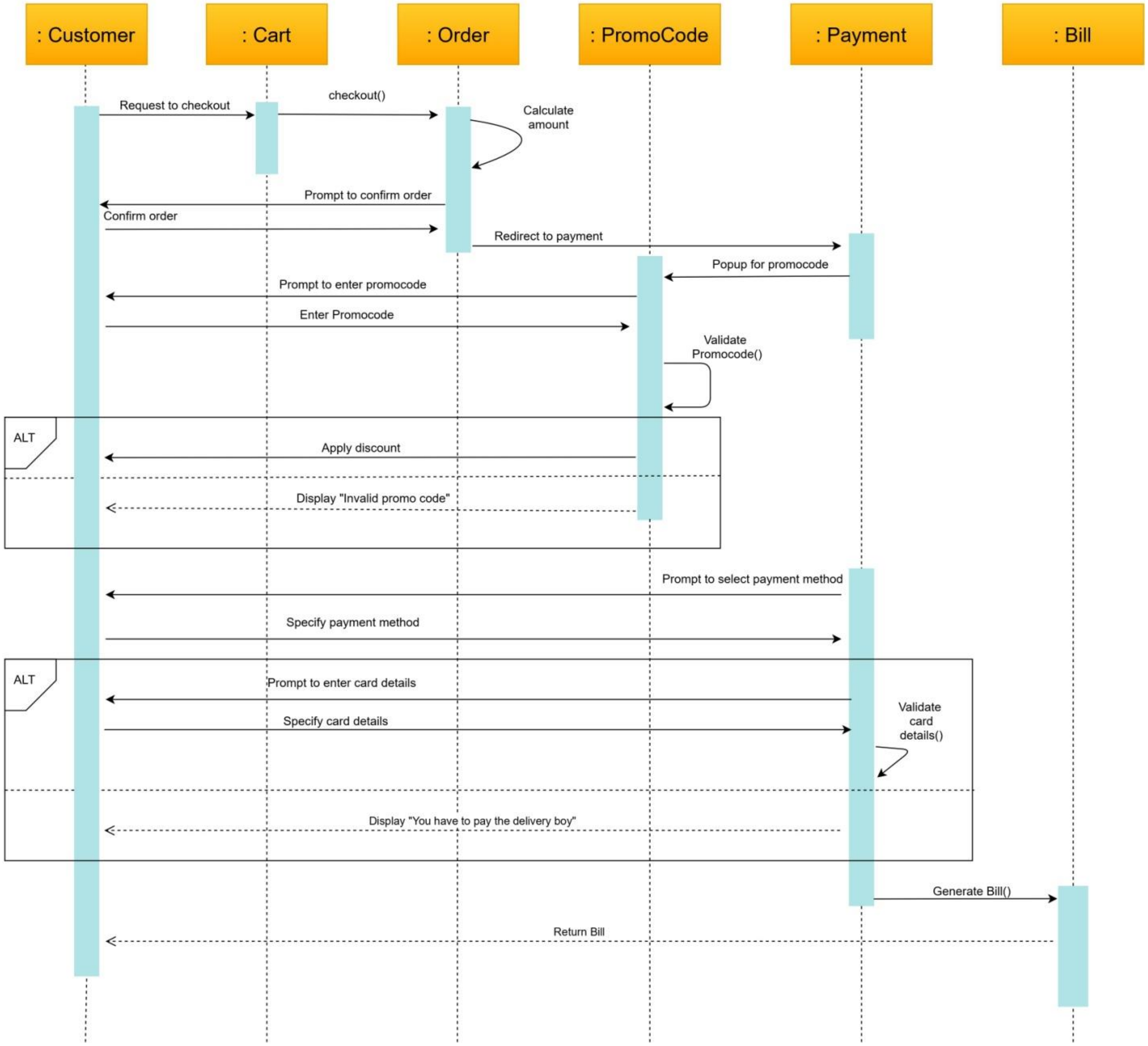


II.Class Diagram

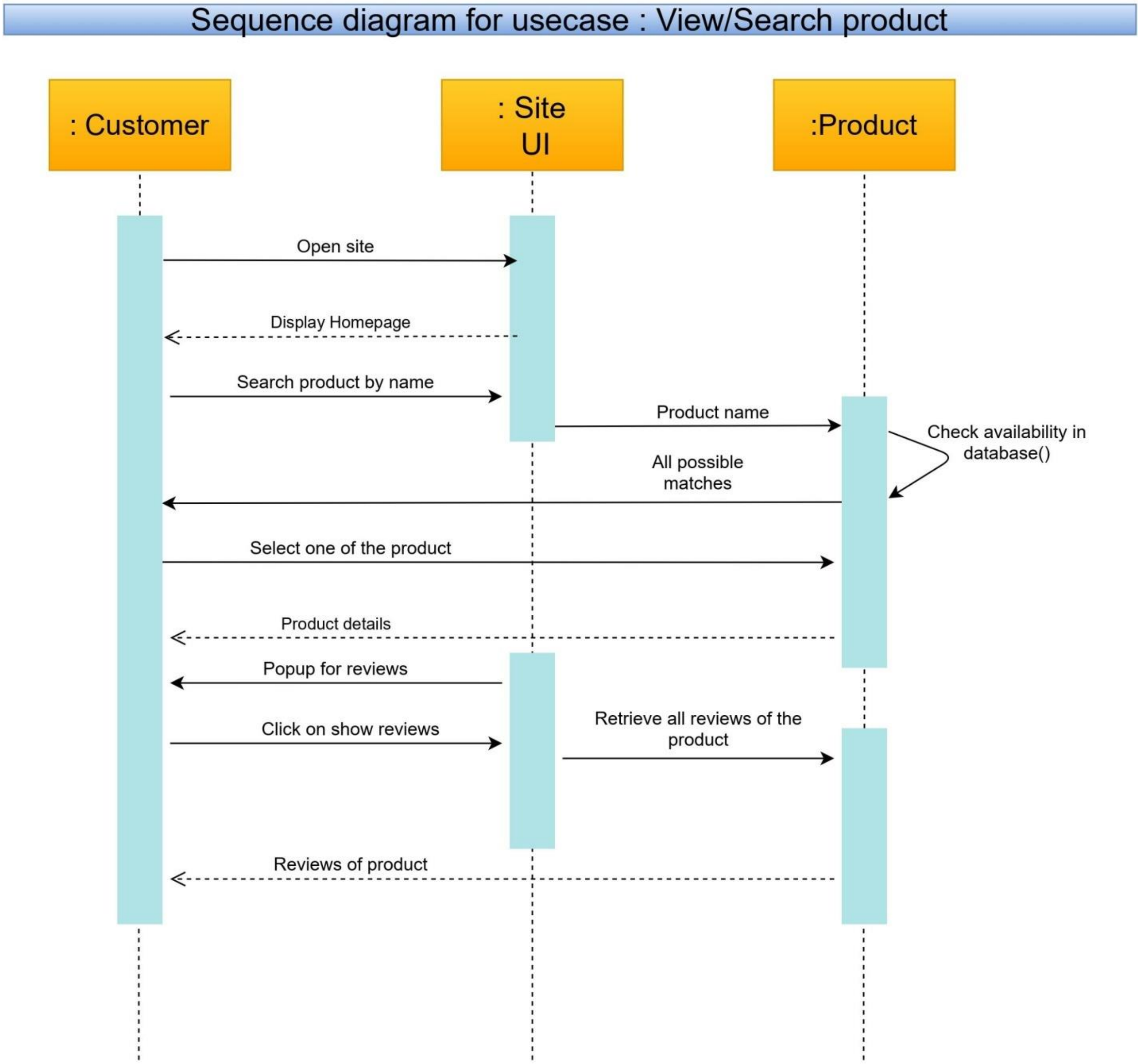


III.Sequence Diagram(1)

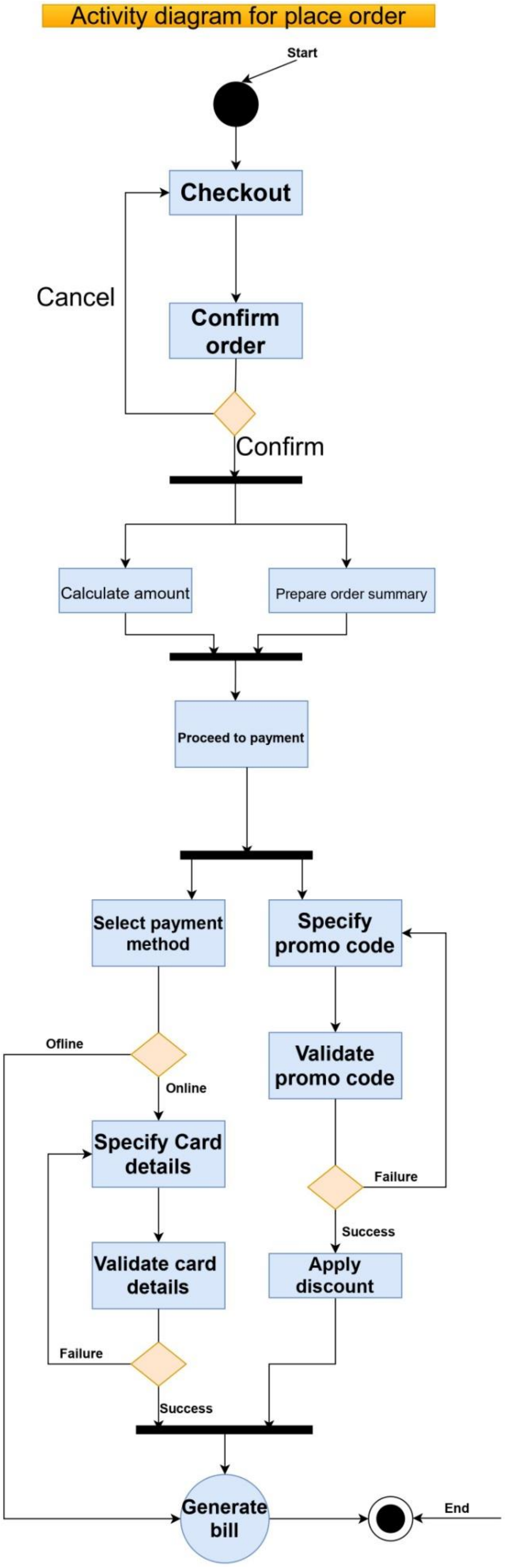
Sequence diagram for use case : Place Order



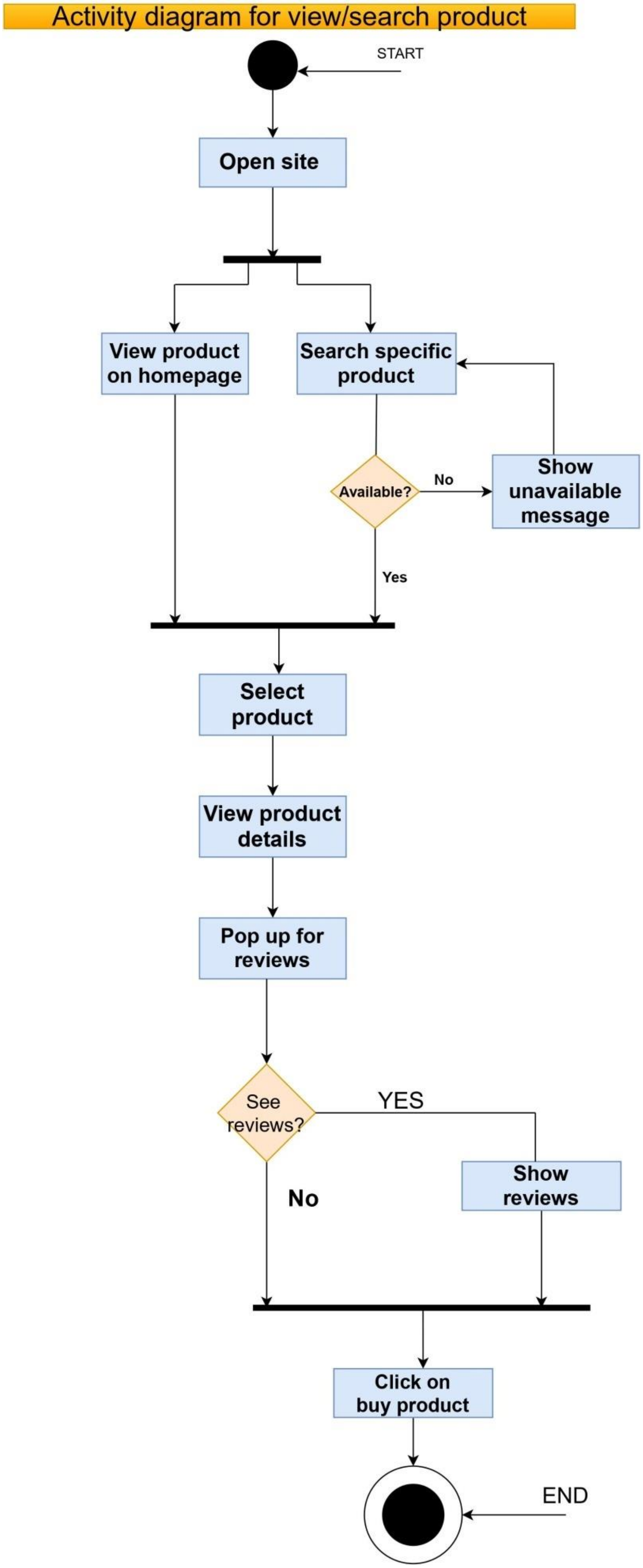
III.Sequence Diagram(2)



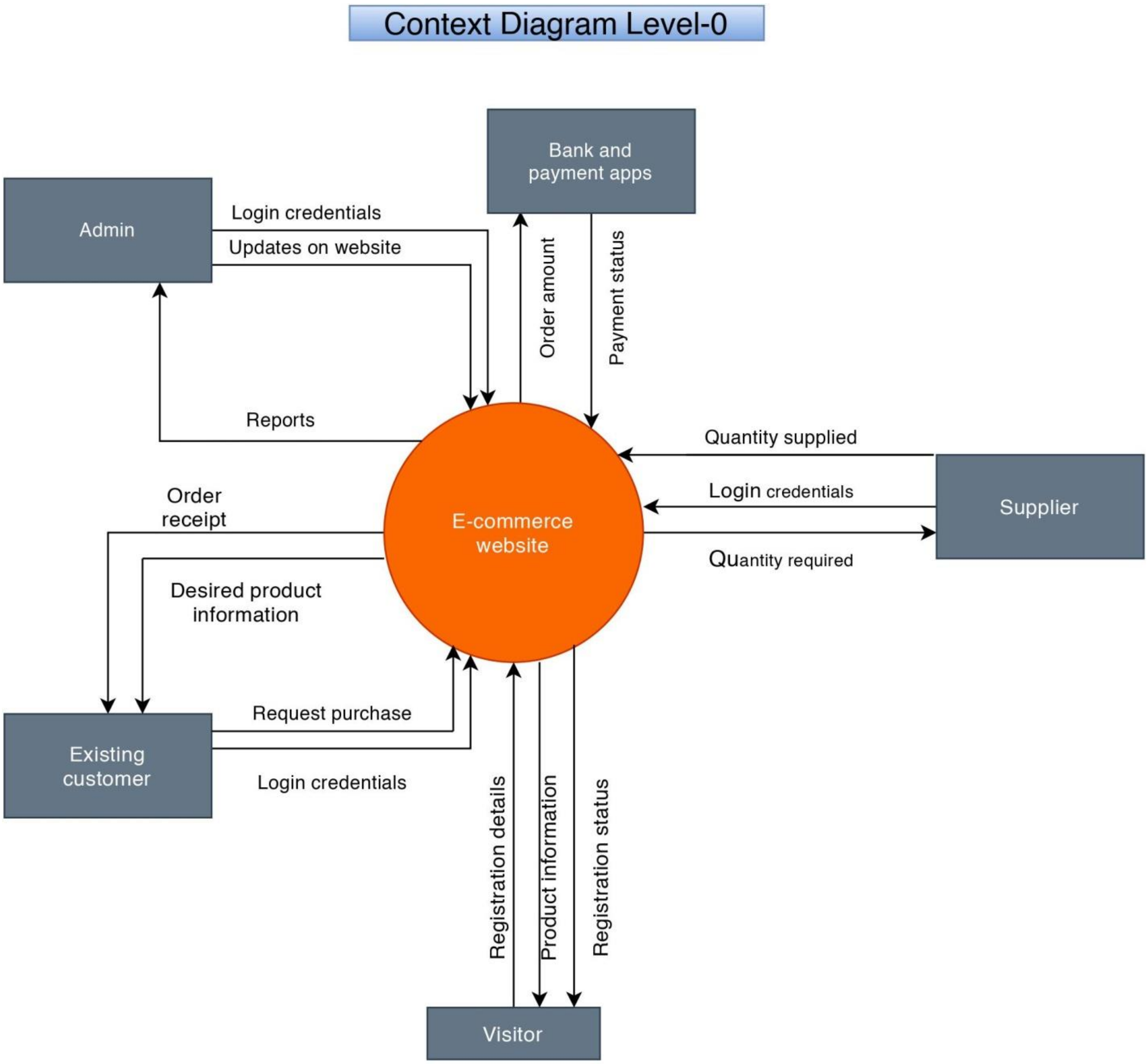
IV.Acitivity Diagram(1)

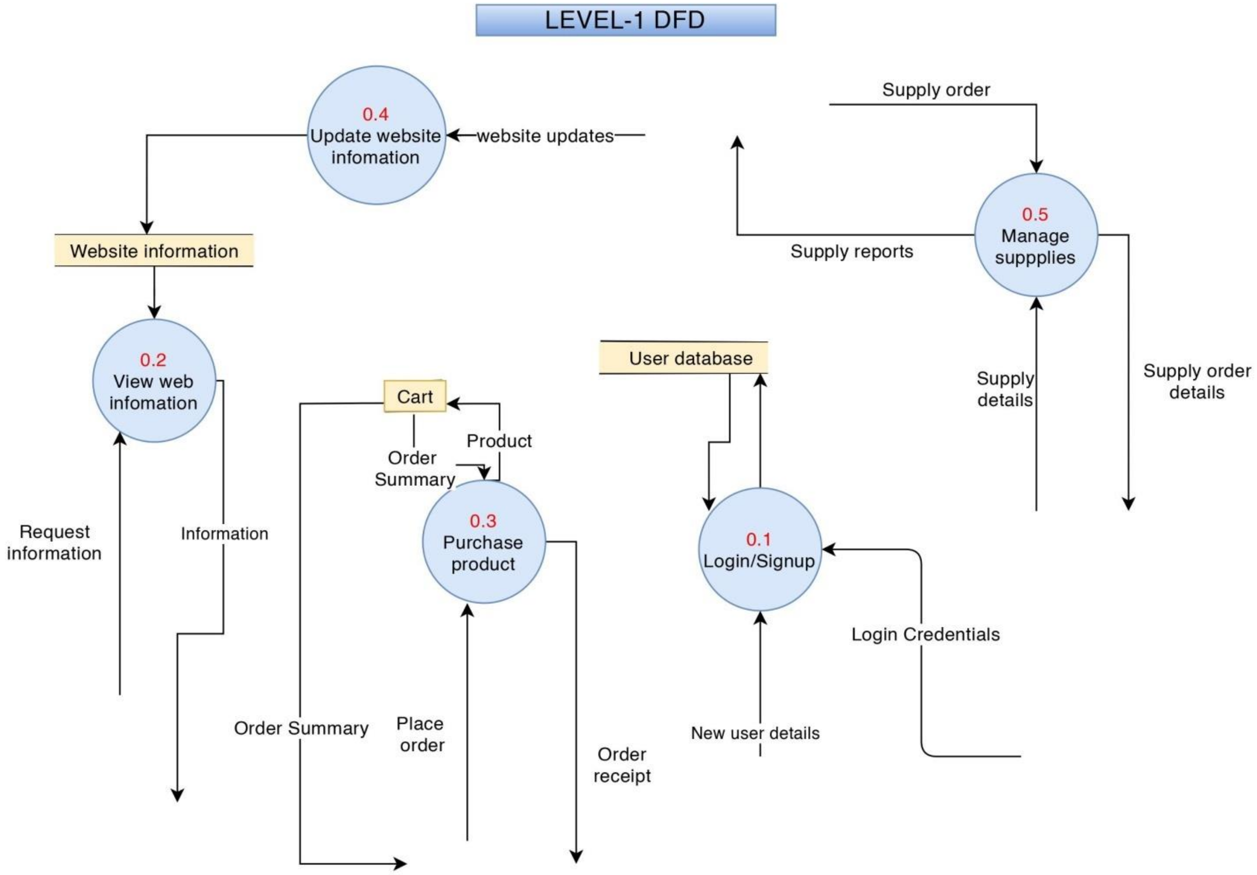


IV.Acitivity Diagram(2)

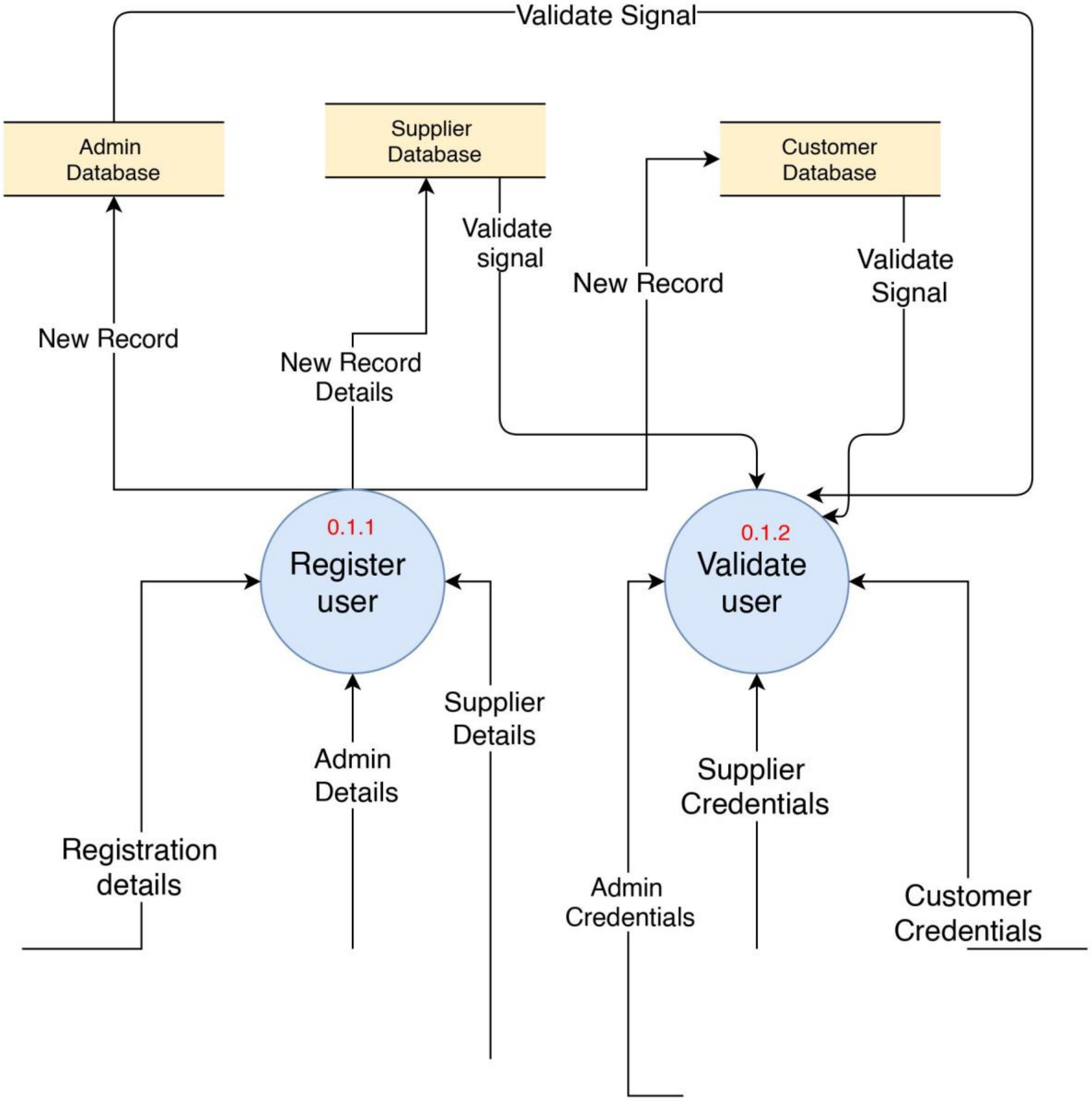


V. Data Flow Diagram(DFD)

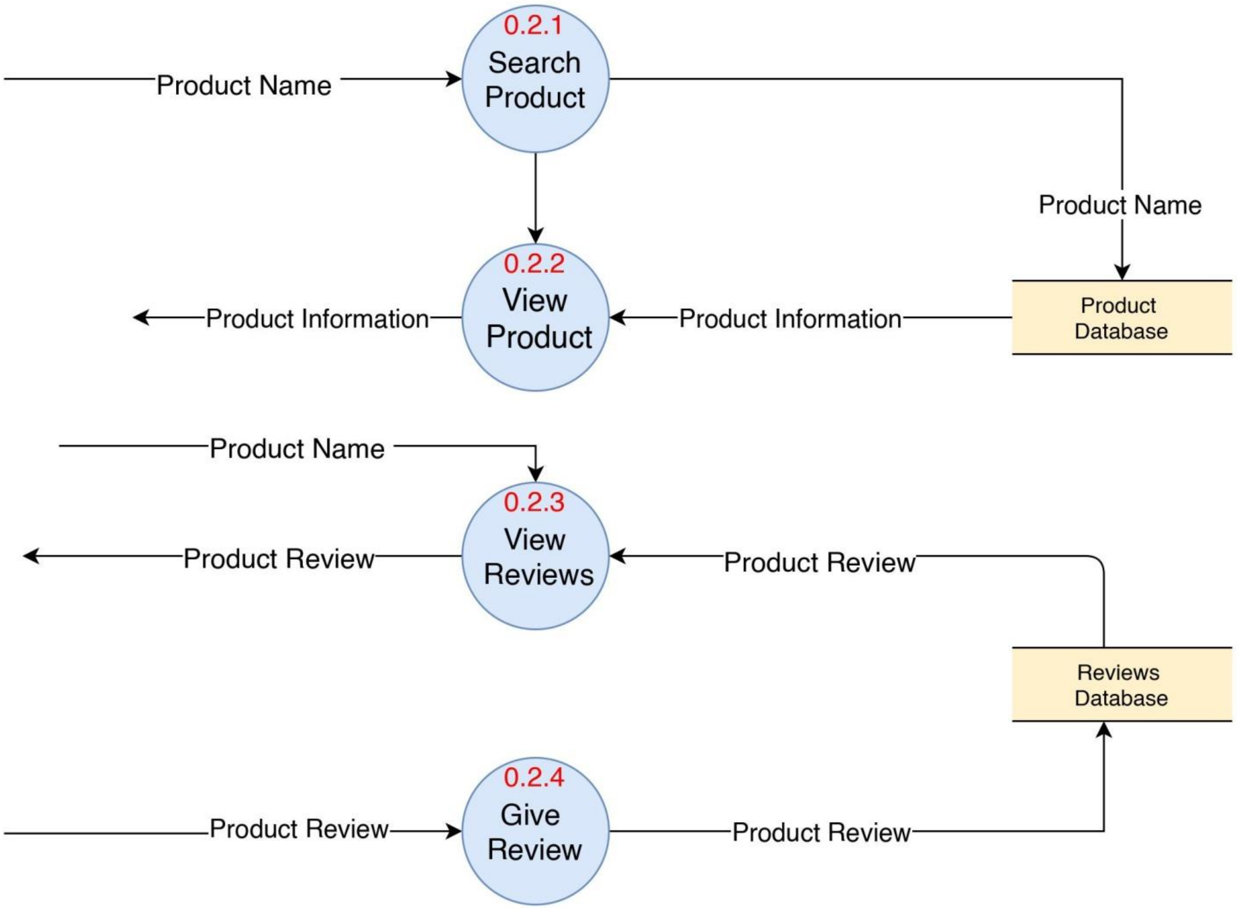



LEVEL-1 DFD
0.4
Update website infomation
website updates
Website information
0.2
View web infomation
Request information
Information
Supply order
Supply reports
0.5
Manage suppplies
Supply details
Supply order details
User database
Cart
Product
Order Summary
0.3
Purchase product
Order Summary
Place order
Order receipt
0.1
Login/Signup
Login Credentials
New user details

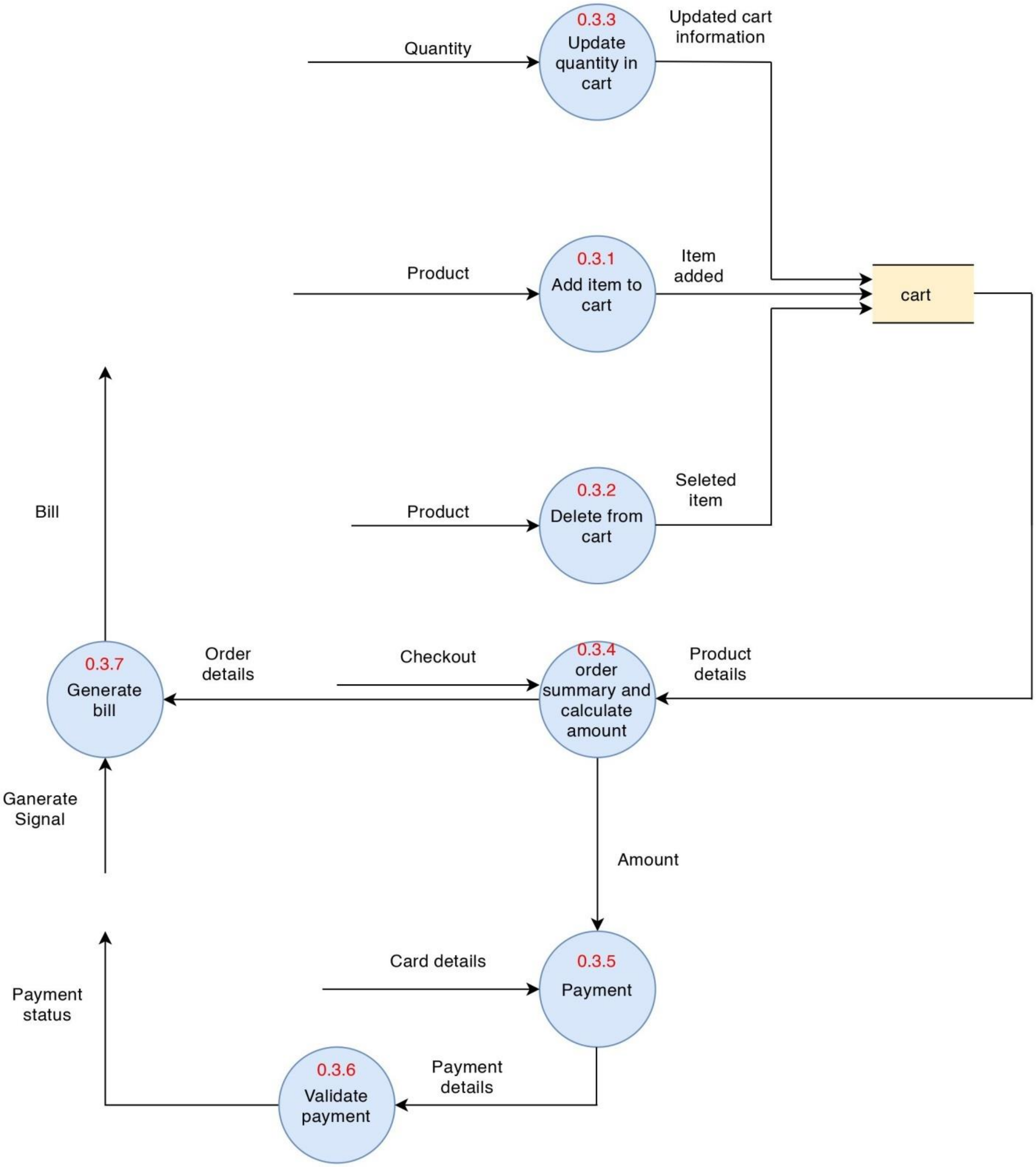
Level 2 DFD for process 1



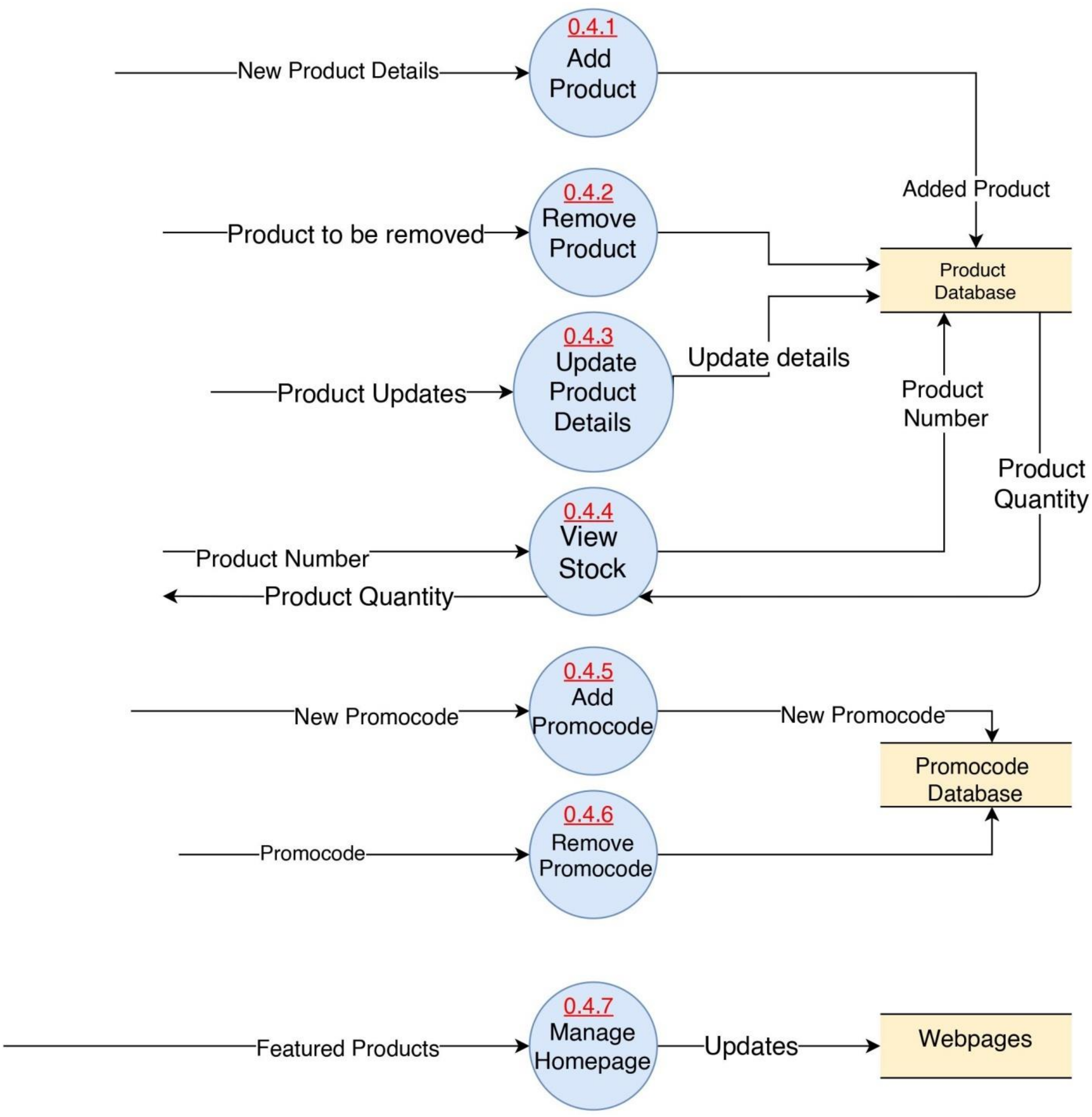
Level 2 DFD for process 2



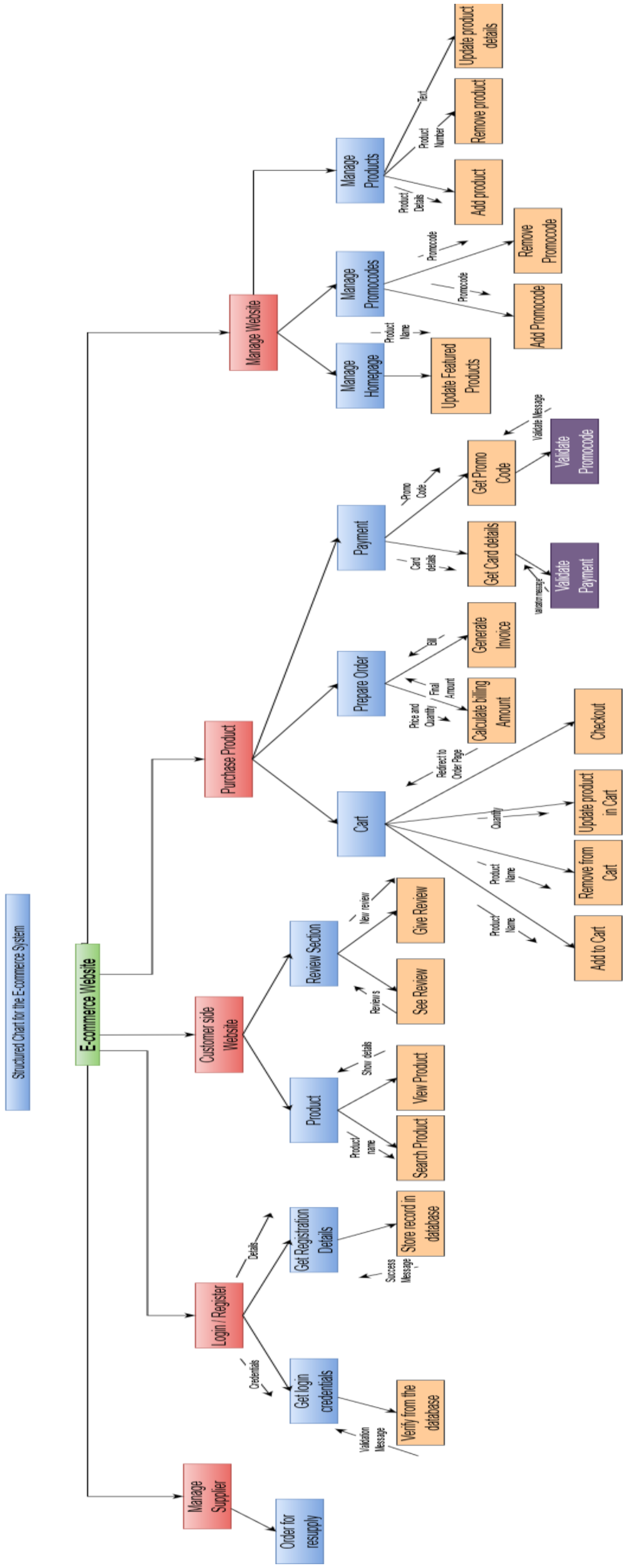
Level 2 DFD for process 3



Level 2 DFD for process 4



6.Structured Chart



5.Implementation Details

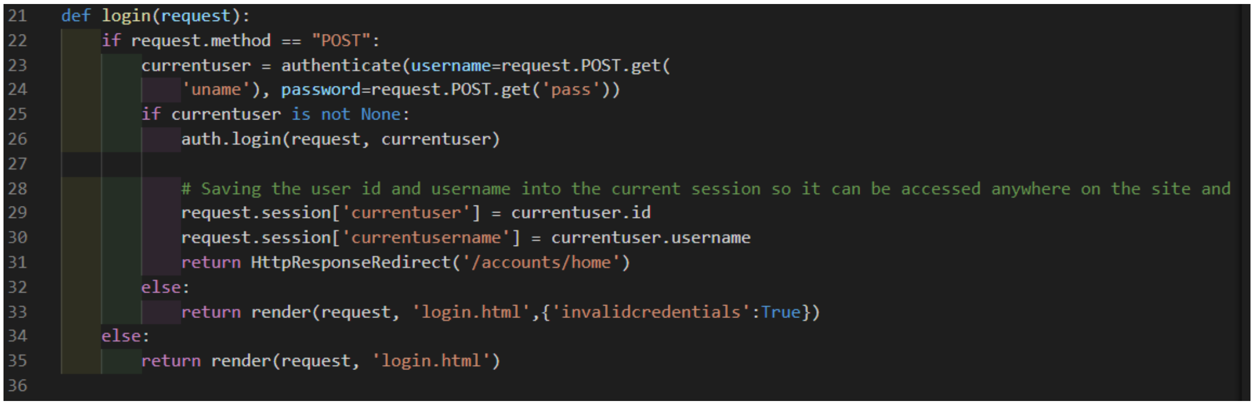
Modules :

1.Customer module(User)
2.Supplier module(User)
3.Product module
4.Cart module
5.Review module

- Each module is made up of number of methods.This methods are shown module-wise below.

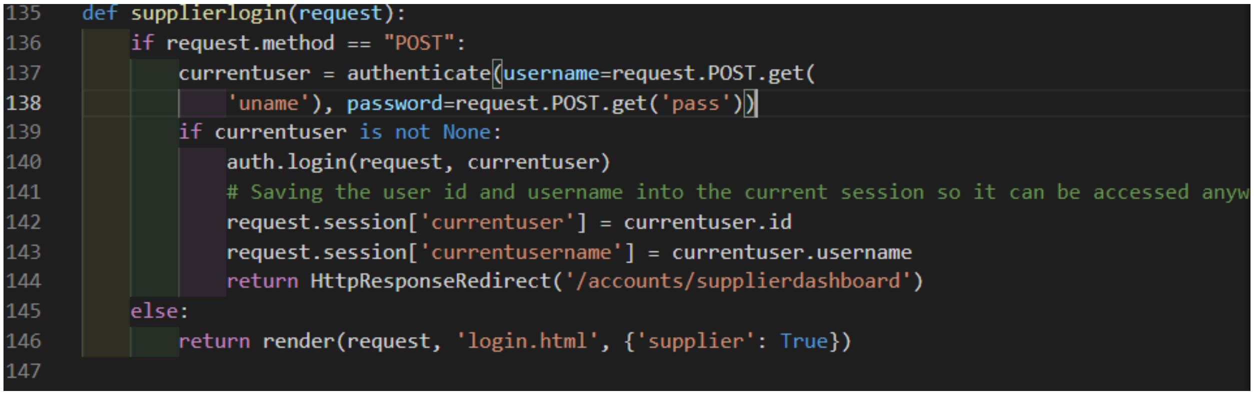
Major Functions :

1] Customer Login – Customer provides login credentials and upon successful authorization is redirected to the homepage.



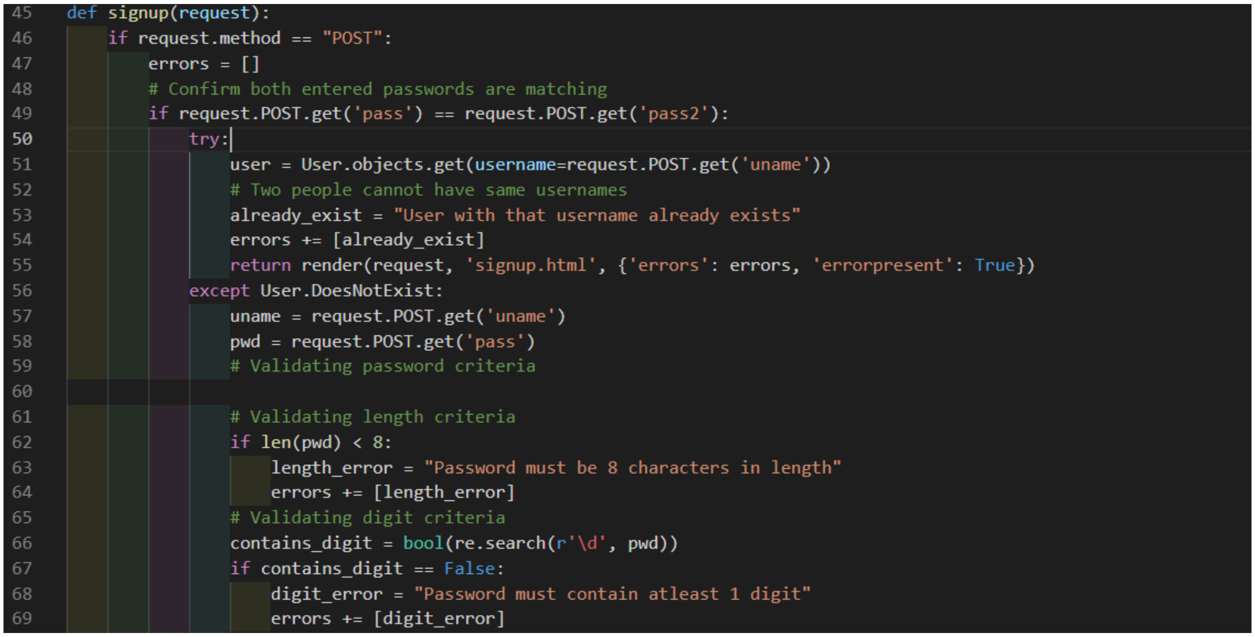
```
def login(request):
    if request.method == "POST":
        currentuser = authenticate(username=request.POST.get(
            'uname'), password=request.POST.get('pass'))
        if currentuser is not None:
            auth.login(request, currentuser)

            # Saving the user id and username into the current session so it can be accessed anywhere on the site and
            request.session['currentuser'] = currentuser.id
            request.session['currentusername'] = currentuser.username
            return HttpResponseRedirect('/accounts/home')
        else:
            return render(request, 'login.html',{'invalidcredentials':True})
    else:
        return render(request, 'login.html')
```

2] Supplier Login – Supplier enters his login credentials and upon successful authentication is redirected to his dashboard where he can see his pending supply orders.



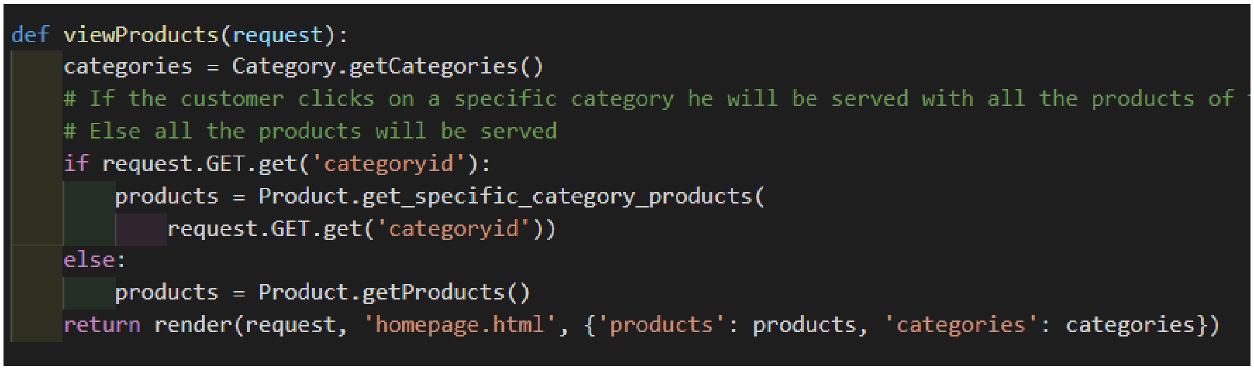
```python
def supplierlogin(request):
    if request.method == "POST":
        currentuser = authenticate(username=request.POST.get(
            'uname'), password=request.POST.get('pass'))
        if currentuser is not None:
            auth.login(request, currentuser)
            # Saving the user id and username into the current session so it can be accessed anyw
            request.session['currentuser'] = currentuser.id
            request.session['currentusername'] = currentuser.username
            return HttpResponseRedirect('/accounts/supplierdashboard')
    else:
        return render(request, 'login.html', {'supplier': True})
```

3] Customer Signup – Customer inputs his/her personal information and it will be stored into database and customer redirected to homepage.



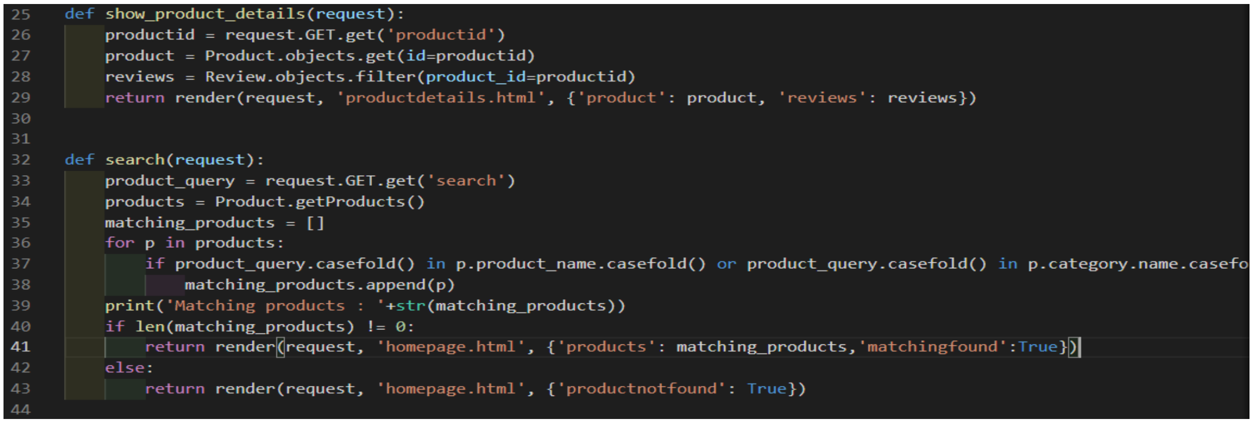
```python
def signup(request):
    if request.method == "POST":
        errors = []
        # Confirm both entered passwords are matching
        if request.POST.get('pass') == request.POST.get('pass2'):
            try:
                user = User.objects.get(username=request.POST.get('uname'))
                # Two people cannot have same usernames
                already_exist = "User with that username already exists"
                errors += [already_exist]
                return render(request, 'signup.html', {'errors': errors, 'errorpresent': True})
            except User.DoesNotExist:
                uname = request.POST.get('uname')
                pwd = request.POST.get('pass')
                # Validating password criteria

                # Validating length criteria
                if len(pwd) < 8:
                    length_error = "Password must be 8 characters in length"
                    errors += [length_error]
                # Validating digit criteria
                contains_digit = bool(re.search(r'\d', pwd))
                if contains_digit == False:
                    digit_error = "Password must contain atleast 1 digit"
                    errors += [digit_error]
```

4.View Products – All products or requested category products are shown to user.



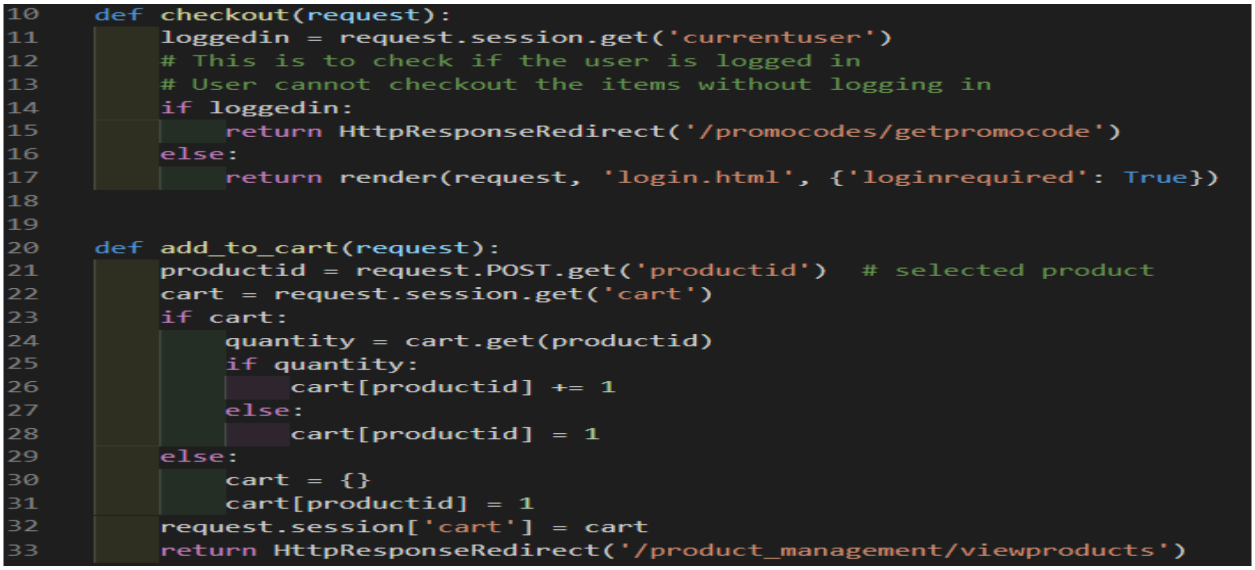
```
def viewProducts(request):
    categories = Category.getCategories()
    # If the customer clicks on a specific category he will be served with all the products of 
    # Else all the products will be served
    if request.GET.get('categoryid'):
        products = Product.get_specific_category_products(
            request.GET.get('categoryid'))
    else:
        products = Product.getProducts()
    return render(request, 'homepage.html', {'products': products, 'categories': categories})
```

5] View Product details and Search – Customer can view particular product details or search a specific product.



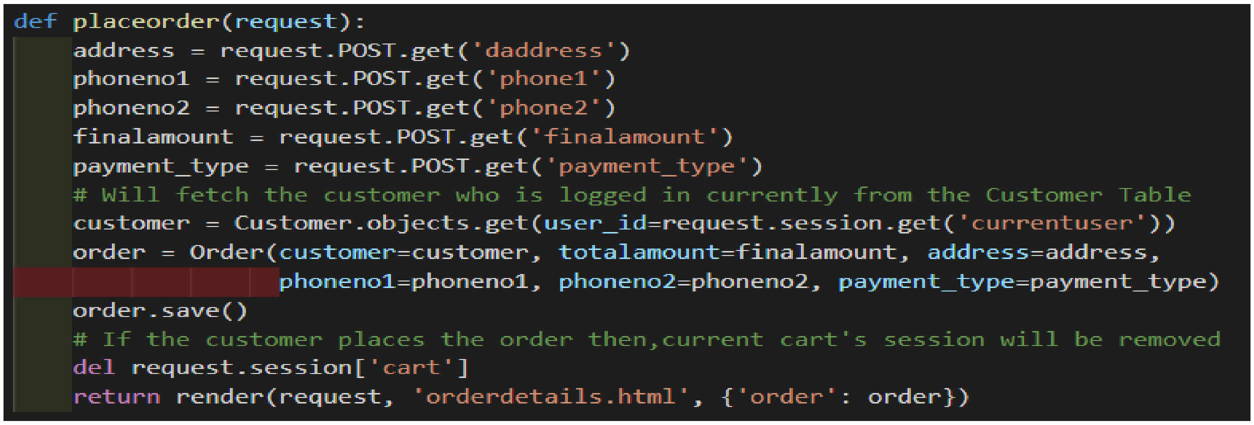
```
def show_product_details(request):
    productid = request.GET.get('productid')
    product = Product.objects.get(id=productid)
    reviews = Review.objects.filter(product_id=productid)
    return render(request, 'productdetails.html', {'product': product, 'reviews': reviews})


def search(request):
    product_query = request.GET.get('search')
    products = Product.getProducts()
    matching_products = []
    for p in products:
        if product_query.casefold() in p.product_name.casefold() or product_query.casefold() in p.category.name.casefo
            matching_products.append(p)
    print('Matching products : '+str(matching_products))
    if len(matching_products) != 0:
        return render(request, 'homepage.html', {'products': matching_products,'matchingfound':True})
    else:
        return render(request, 'homepage.html', {'productnotfound': True})

```

6] Add to cart and checkout – User can add product to cart but checkout only if he/her is logged in.



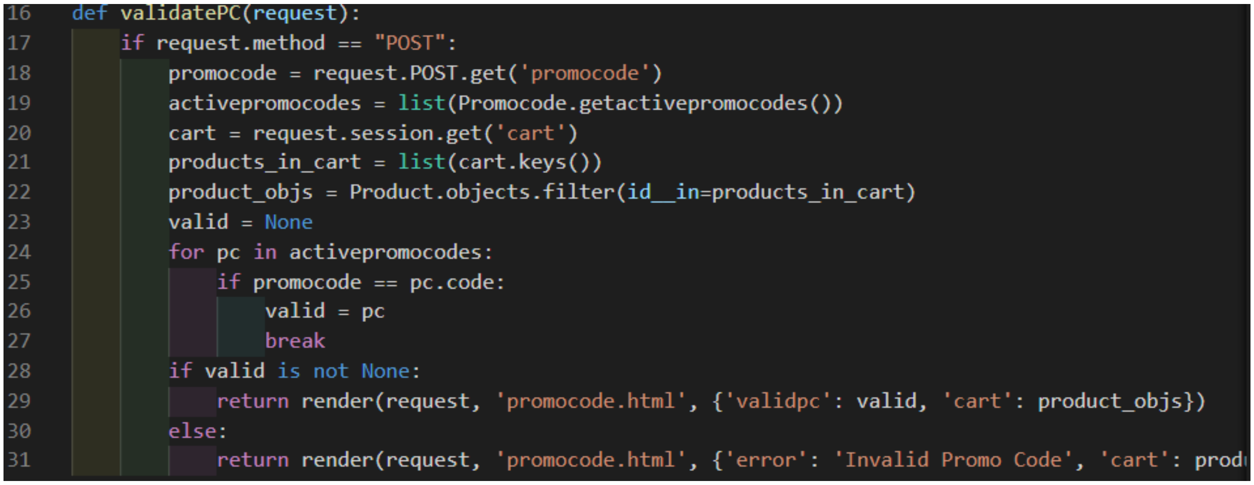
```python
def checkout(request):
    loggedin = request.session.get('currentuser')
    # This is to check if the user is logged in
    # User cannot checkout the items without logging in
    if loggedin:
        return HttpResponseRedirect('/promocodes/getpromocode')
    else:
        return render(request, 'login.html', {'loginrequired': True})


def add_to_cart(request):
    productid = request.POST.get('productid')  # selected product
    cart = request.session.get('cart')
    if cart:
        quantity = cart.get(productid)
        if quantity:
            cart[productid] += 1
        else:
            cart[productid] = 1
    else:
        cart = {}
        cart[productid] = 1
    request.session['cart'] = cart
    return HttpResponseRedirect('/product_management/viewproducts')
```

7] Place order – After completing checkout customer places order and provides necessary details for delivery.After successfully placing order customer is shown a message with all order details.



```python
def placeorder(request):
    address = request.POST.get('daddress')
    phoneno1 = request.POST.get('phone1')
    phoneno2 = request.POST.get('phone2')
    finalamount = request.POST.get('finalamount')
    payment_type = request.POST.get('payment_type')
    # Will fetch the customer who is logged in currently from the Customer Table
    customer = Customer.objects.get(user_id=request.session.get('currentuser'))
    order = Order(customer=customer, totalamount=finalamount, address=address,
                  phoneno1=phoneno1, phoneno2=phoneno2, payment_type=payment_type)
    order.save()
    # If the customer places the order then,current cart's session will be removed
    del request.session['cart']
    return render(request, 'orderdetails.html', {'order': order})
```

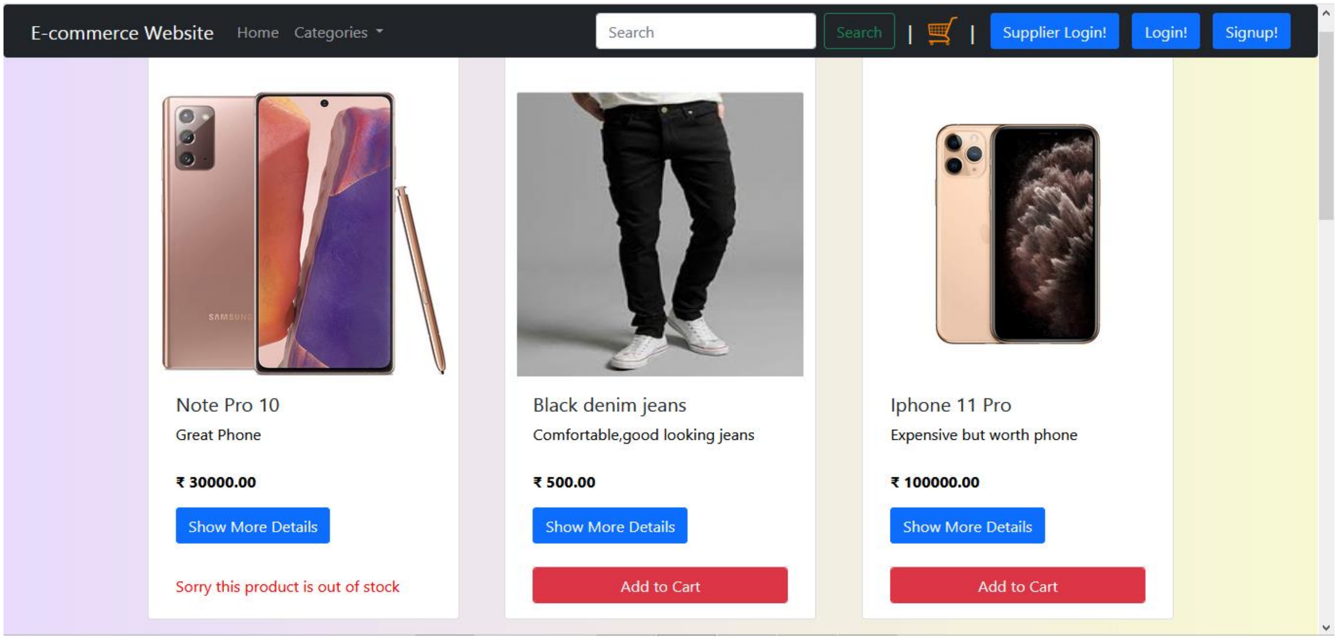
8]Validating promo codes – If user enters a coupon to get discount it must be first checked if it is active and valid.



```python
def validatePC(request):
    if request.method == "POST":
        promocode = request.POST.get('promocode')
        activepromocodes = list(Promocode.getactivepromocodes())
        cart = request.session.get('cart')
        products_in_cart = list(cart.keys())
        product_objs = Product.objects.filter(id__in=products_in_cart)
        valid = None
        for pc in activepromocodes:
            if promocode == pc.code:
                valid = pc
                break
        if valid is not None:
            return render(request, 'promocode.html', {'validpc': valid, 'cart': product_objs})
        else:
            return render(request, 'promocode.html', {'error': 'Invalid Promo Code', 'cart': prod
```

6.Workflow/Layout/Screenshots :

1] Guest user home page



Note Pro 10

Great Phone

₹ 30000.00

Sorry this product is out of stock

Black denim jeans

Comfortable,good looking jeans

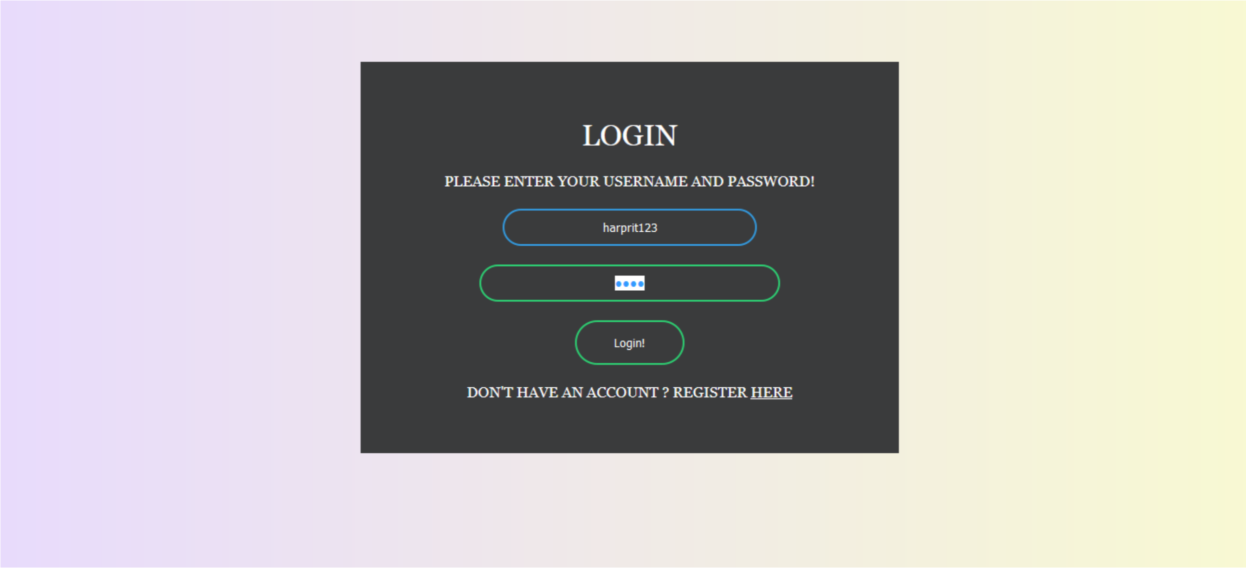
₹ 500.00

Iphone 11 Pro

Expensive but worth phone

₹ 100000.00

2] Login page



LOGIN

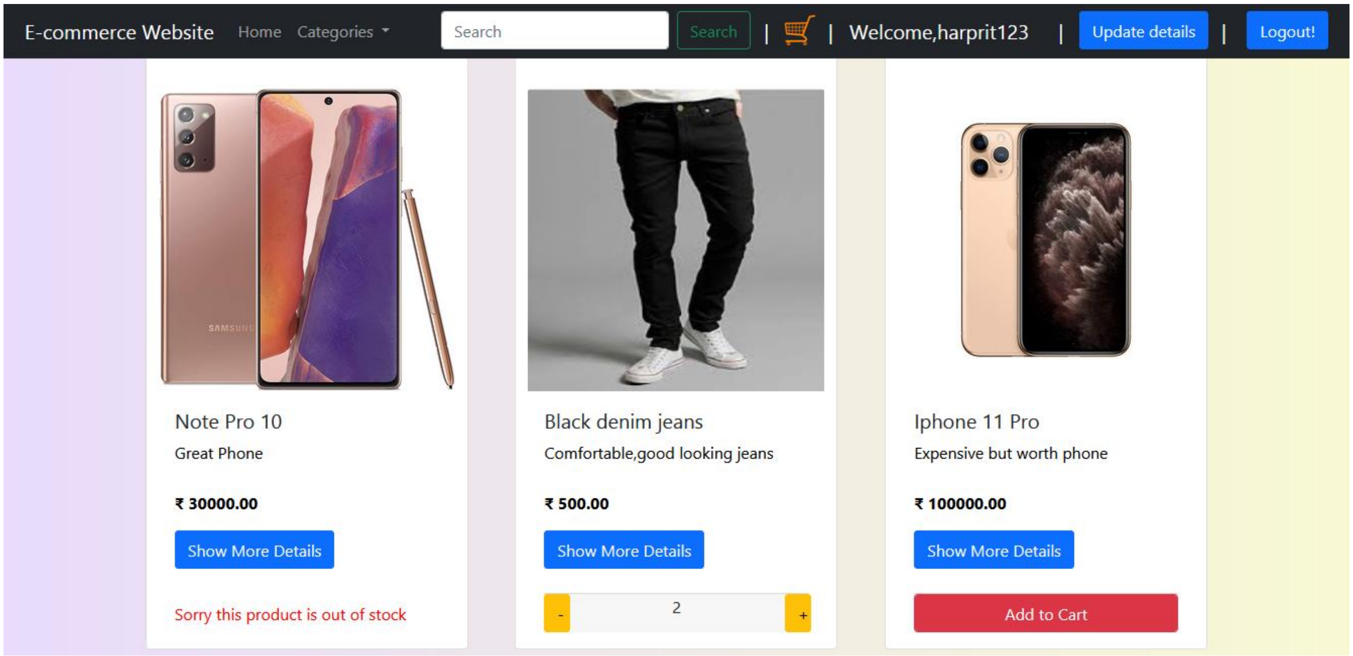
PLEASE ENTER YOUR USERNAME AND PASSWORD!

harprit123

••••

DON'T HAVE AN ACCOUNT ? REGISTER HERE

3]Adding 2 quantity of “Black denim jeans” into cart

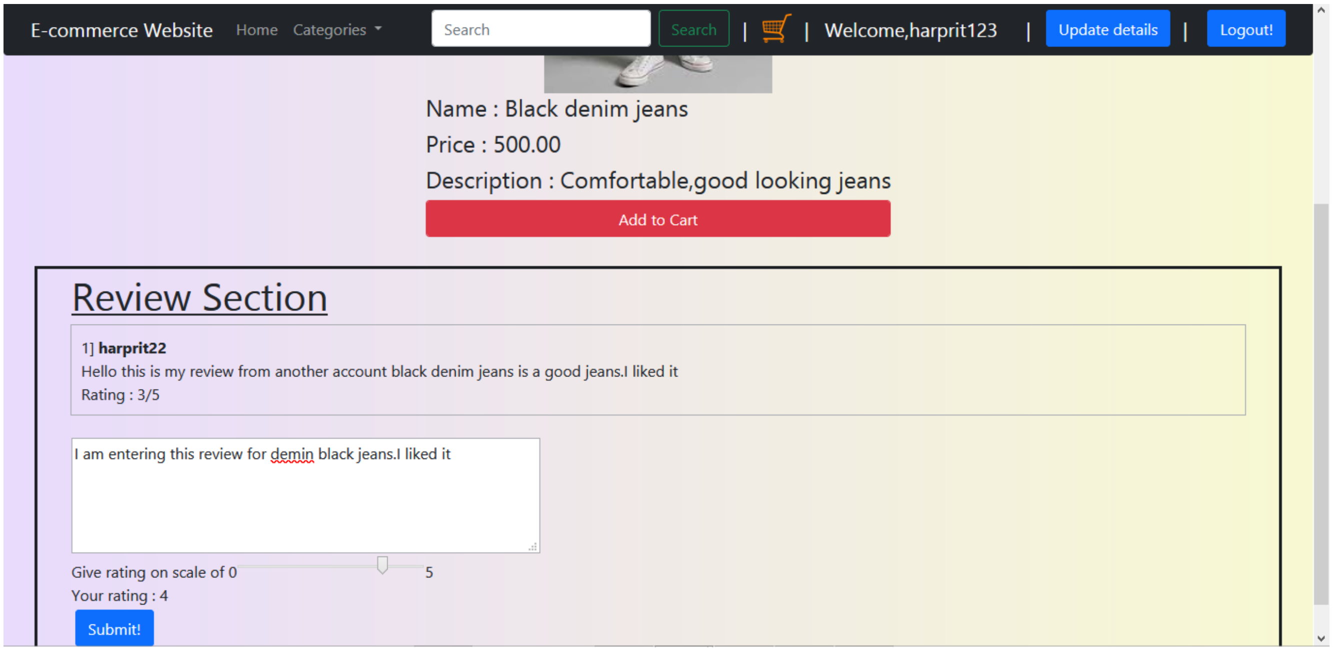


Note Pro 10
Great Phone
₹ 30000.00
Sorry this product is out of stock

Black denim jeans
Comfortable,good looking jeans
₹ 500.00

Iphone 11 Pro
Expensive but worth phone
₹ 100000.00

4] See reviews of that product from other customers and add one review.



Name : Black denim jeans
Price : 500.00
Description : Comfortable,good looking jeans

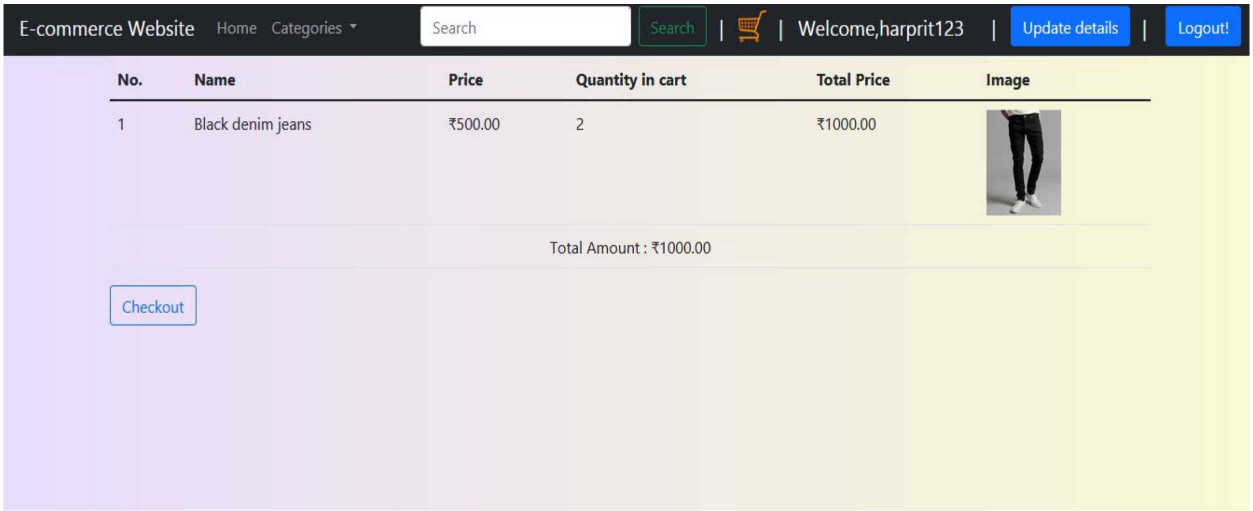
Review Section

1] **harprit22**
Hello this is my review from another account black denim jeans is a good jeans.I liked it
Rating : 3/5

I am entering this review for demin black jeans.I liked it

Give rating on scale of 0 5
Your rating : 4

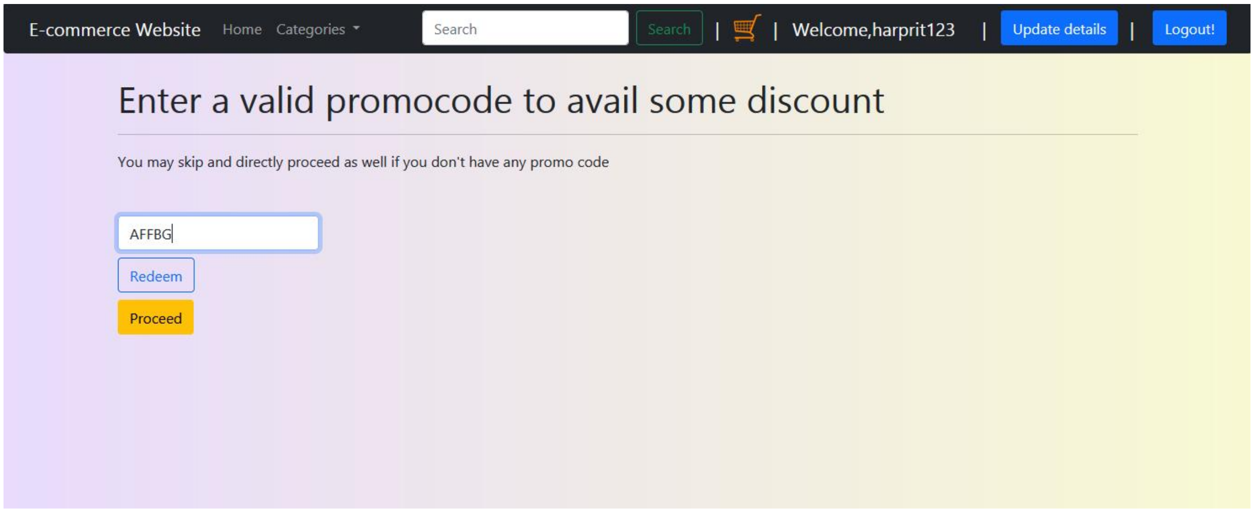
5] View cart and checkout



No.	Name	Price	Quantity in cart	Total Price	Image
1	Black denim jeans	₹500.00	2	₹1000.00	

Total Amount : ₹1000.00

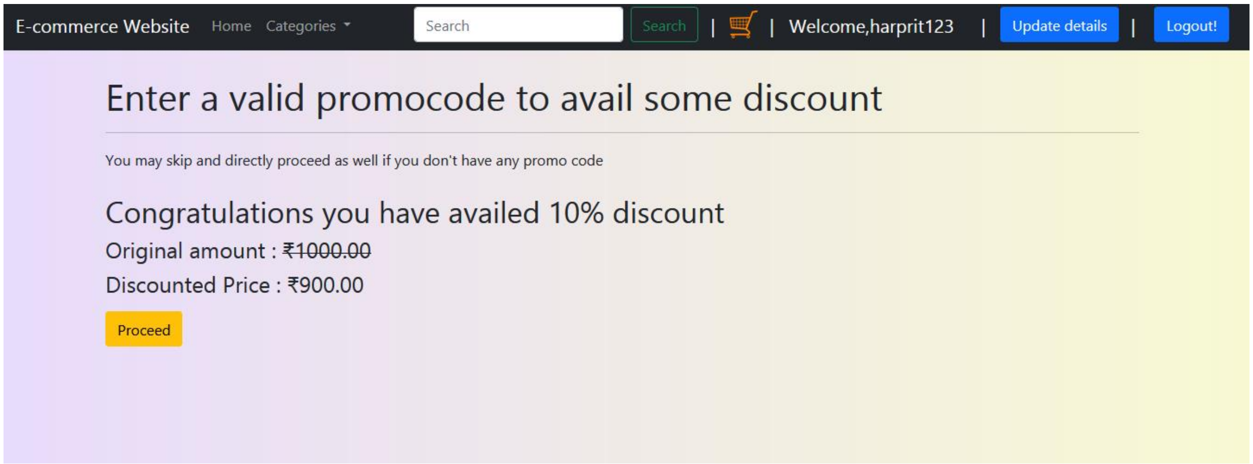
6] Enter discount promo code and validate



Enter a valid promocode to avail some discount

You may skip and directly proceed as well if you don't have any promo code

AFFBG



Enter a valid promocode to avail some discount

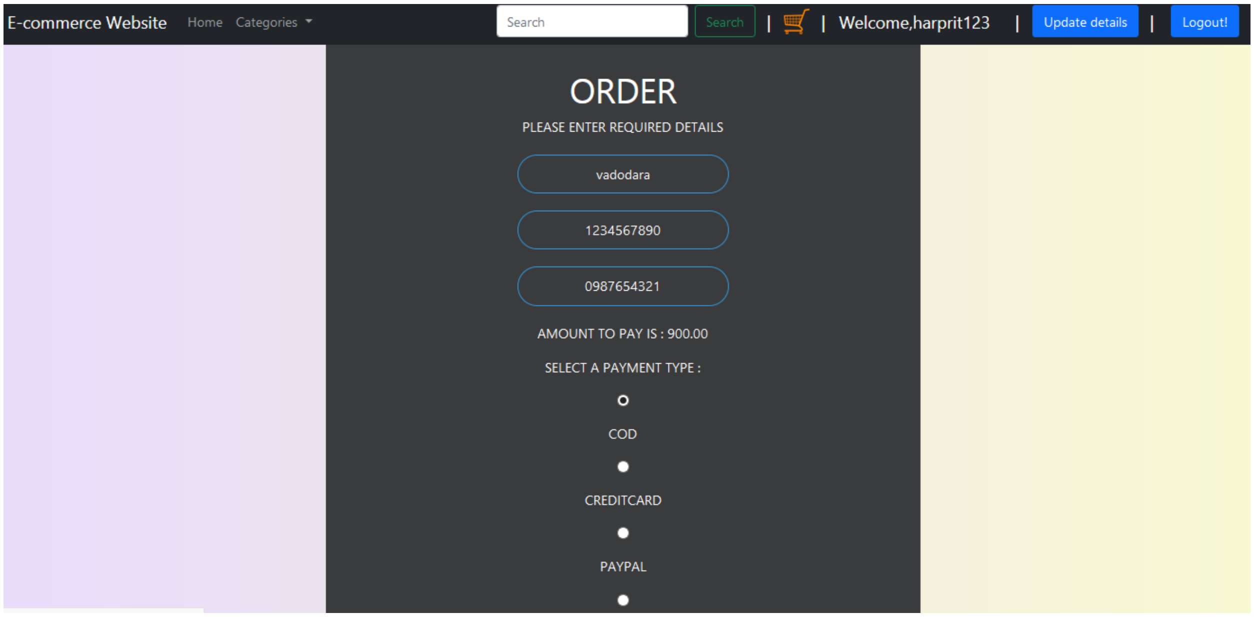
You may skip and directly proceed as well if you don't have any promo code

Congratulations you have availed 10% discount

Original amount : ~~₹1000.00~~

Discounted Price : ₹900.00

7] Place order and give details for delivery and get order summary.



ORDER

PLEASE ENTER REQUIRED DETAILS

vadodara

1234567890

0987654321

AMOUNT TO PAY IS : 900.00

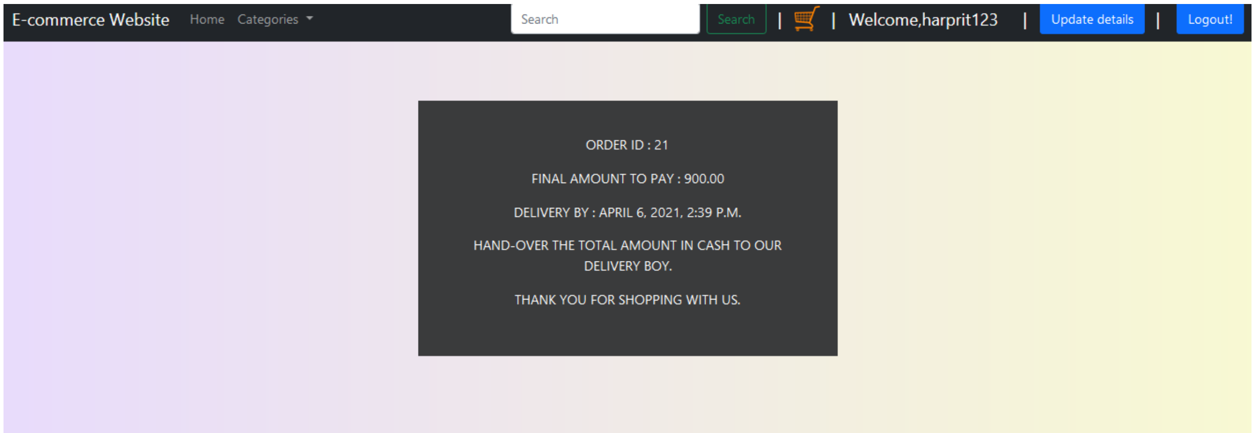
SELECT A PAYMENT TYPE :

○ COD

● CREDITCARD

● PAYPAL

●



ORDER ID : 21

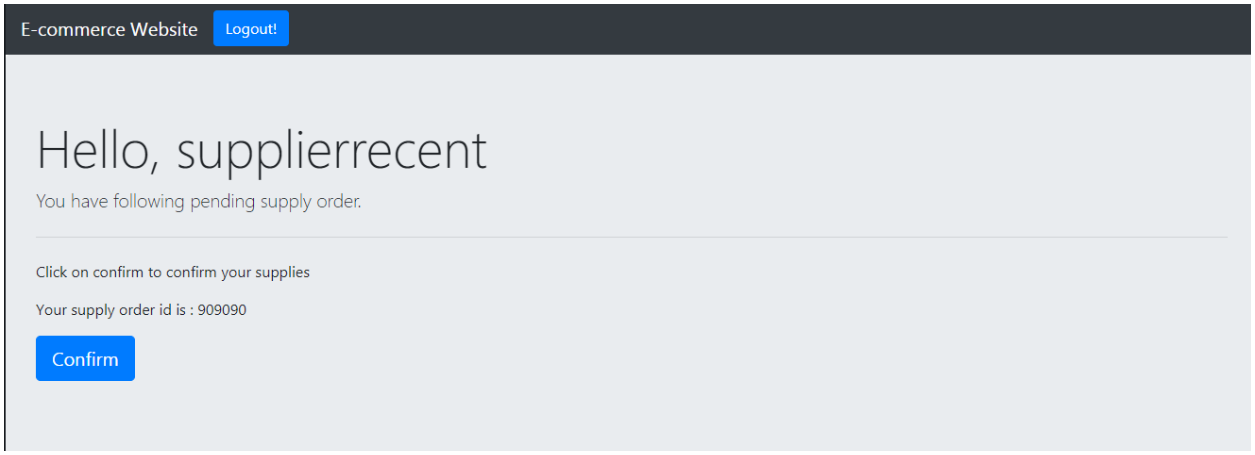
FINAL AMOUNT TO PAY : 900.00

DELIVERY BY : APRIL 6, 2021, 2:39 P.M.

HAND-OVER THE TOTAL AMOUNT IN CASH TO OUR DELIVERY BOY.

THANK YOU FOR SHOPPING WITH US.

8] Supplier Panel : Supplier can see his pending supply order and confirm that order.



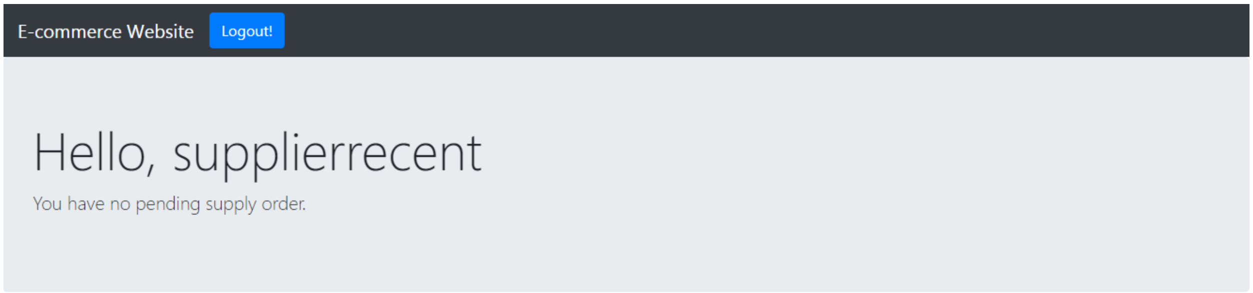
Hello, supplierrecent

You have following pending supply order.

Click on confirm to confirm your supplies

Your supply order id is : 909090

After confirming supply order.



Hello, supplierrecent

You have no pending supply order.

7.Conclusion

Hereby, we conclude that we have successfully implemented the admin, customer, supplier module.

Admin will be able to add/remove the category and products, add/remove promo codes,place supply orders given by the management team.

Customers will be able to view, search the products, add them to the cart, apply a valid promo code to get a discount, place an order, give a review for individual products, view reviews of other customers for a particular product.

The supplier will be able to confirm the supply orders.

After successful implementation of all the functionalities, testing was performed by us. Also, we asked different end-users(Family members) who were completely unaware of the system so that we can get a better understanding of how a normal end-user will use the system and we can detect if there are any chances of improvement or flaws.

8. Limitations

1.This project 'E-commerce Website' is not yet hosted on the world wide web. Everything done till now is tested and run on the local server.

2.Payment gateway is not integrated.

3.Supplier module is implemented in a way that he can just confirm the supply orders which are placed by the admin with an supply_order_id. In future versions, plan is to modify it in a way such that the admin can order a specific product from a particular supplier.

8. Future extensions:

1]Integrating the payment gateways of different payment options like Paytm, PayPal and Stripe as well as with other banks.

2]Adding an order tracking functionality so that customers can see at what place their order is.

3]Adding comparing functionality so that 2 products can be compared by a customer.

4]Improve UI for better user interaction.

5] Use API’s for on page features which can be done without refreshing page again and again.

9. Bibliography

- https://docs.djangoproject.com/en/3.1/
- https://www.w3schools.com/css/
- https://www.w3schools.com/howto/howto_js_rangeslider.asp
- https://stackoverflow.com/
- https://www.geeksforgeeks.org/