Faculty of Technology

Dharmsinh Desai University, Nadiad

Department of Computer Engineering

B. Tech. CE Semester – IV

Project Title: E-Commerce Website

(Online Exchange System)

By:

1) **Dhruv Patel, Roll No: CE091, ID-19CEUBG166**

2) **Himanshu Parmar, Roll No: CE087, ID-19CEUSG047**

Contents:

1.Abstract

- E-commerce is an emerging and fast-growing technique which is used to buy and sell categorized product or services, generate report and track history of every customer. In the past 15 years, E-commerce businesses have touched the skies in terms of growth. E-commerce where a normal business integrated with technology is done globally by buying and selling goods on the internet.

- Start a business with E-commerce has a lesser cost than an offline store. 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.

- E-commerce is a web-based application or mobile application implemented for easy to access product or services around the corner of world, buy, sell, chat, transact, get service and maintains records. 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 it's a completely dynamic website
2. New products can be added and removed as well.
3. Users can see reviews of particular products by other users and give one review as well.
4. Buyer can directly chat with seller.

2.1 Technologies/Tools:

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

Tools:

- Visual Studio Code
- Git/GitHub

Platform:

- Local Server

3.Software Requirement Specification

End Users:

1. Admin
2. Buyers
3. Sellers

Functional Requirements:

R1 Registration

R.1.1] Signing up user

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

Input:

- Name, Email Id, 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 and showing username on navigation bar.

R2. Manage Products

R.2.1] Sell Product

- ➢ seller can add a new product on the website.

  Input:

  - Product details, Image, Price, location, owner information etc.

  Output:

  - Redirect to homepage and showing the added item.

R.2.2] Remove Product

- ➢ Only Admin can remove product from the website.

  Input:

  -Admin can remove item using admin panel. click to remove data.

  Output:

  -Product will be remove from the database.

R.2.3] Modify Product

- ➢ Admin can modify details of an existing product on the website.

  Input:

  - Click to edit product.

  Output:

  -Modifications are performed on the database with a confirmation message.

R4 Search and View Product

R.4.1] Search

- ➢ Customer can write product name and this function show the item which he/she is looking for.

  Input:

- In search box write product name.

Output:

- Loads total products and display it on screen.

R.4.2] Quick-view

- ➢ Customer can view details of the product

Input:

-Click on the **Quickview** button.

Output:

-Shows all the details of product like Price, Address, Description, Product name etc.

R5 Manage Cart

R.5.4] Contact Now

- ➢ Contact to buyer and buy the item.

Input:

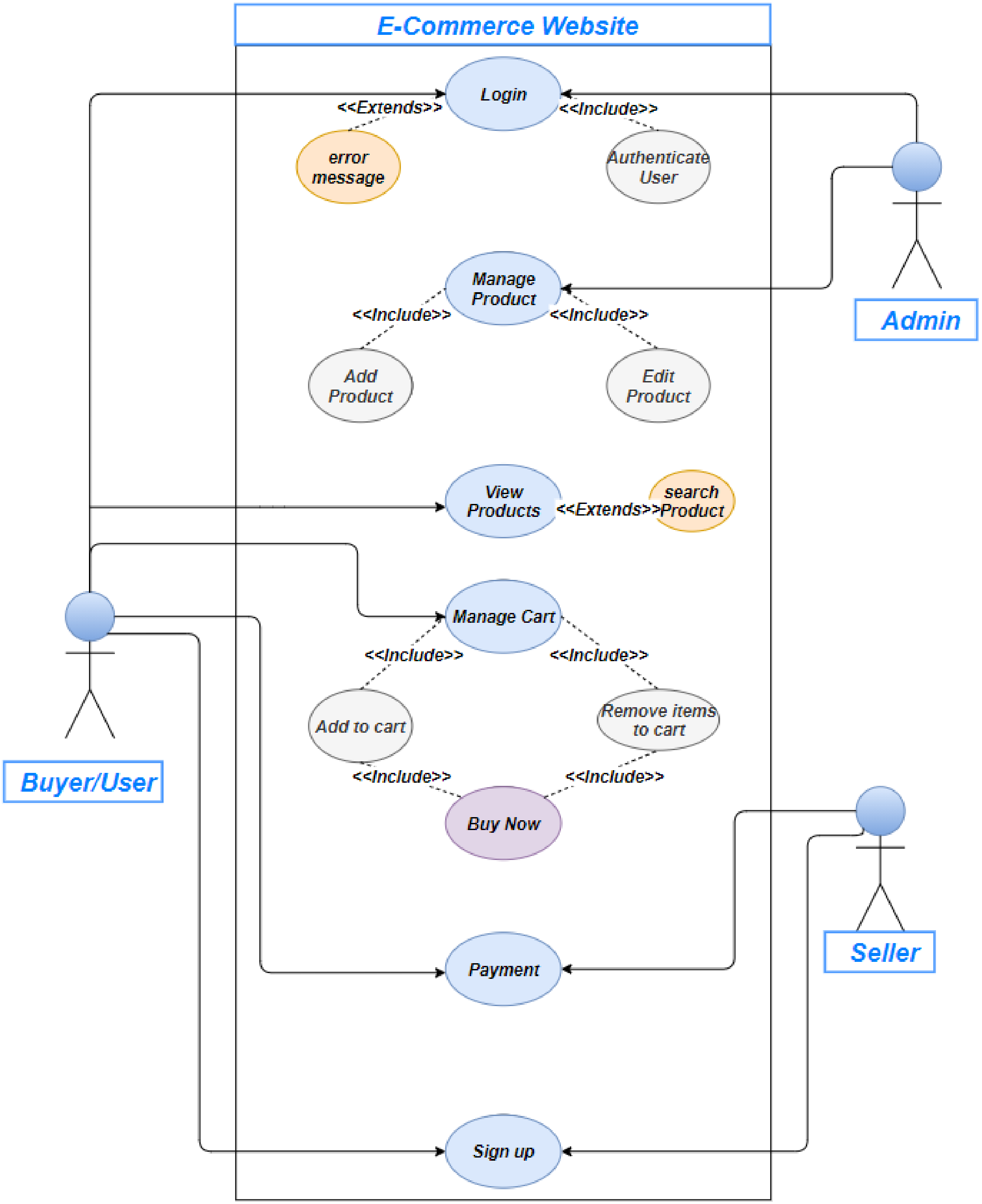
- Click on the proceed to checkout button.

Output:

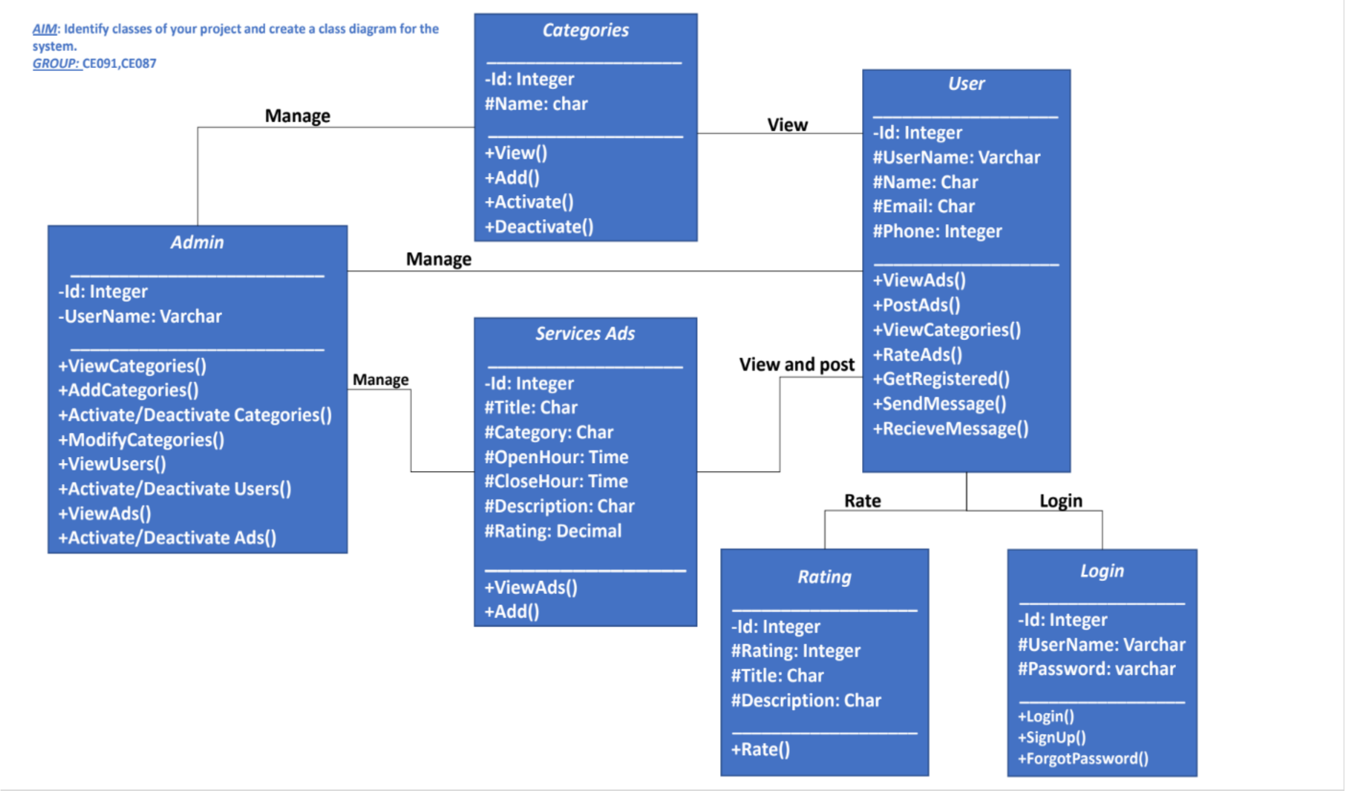
- Redirects buyer to another web-page (Of place order).

4.Design Documents

I. Use Case Diagram

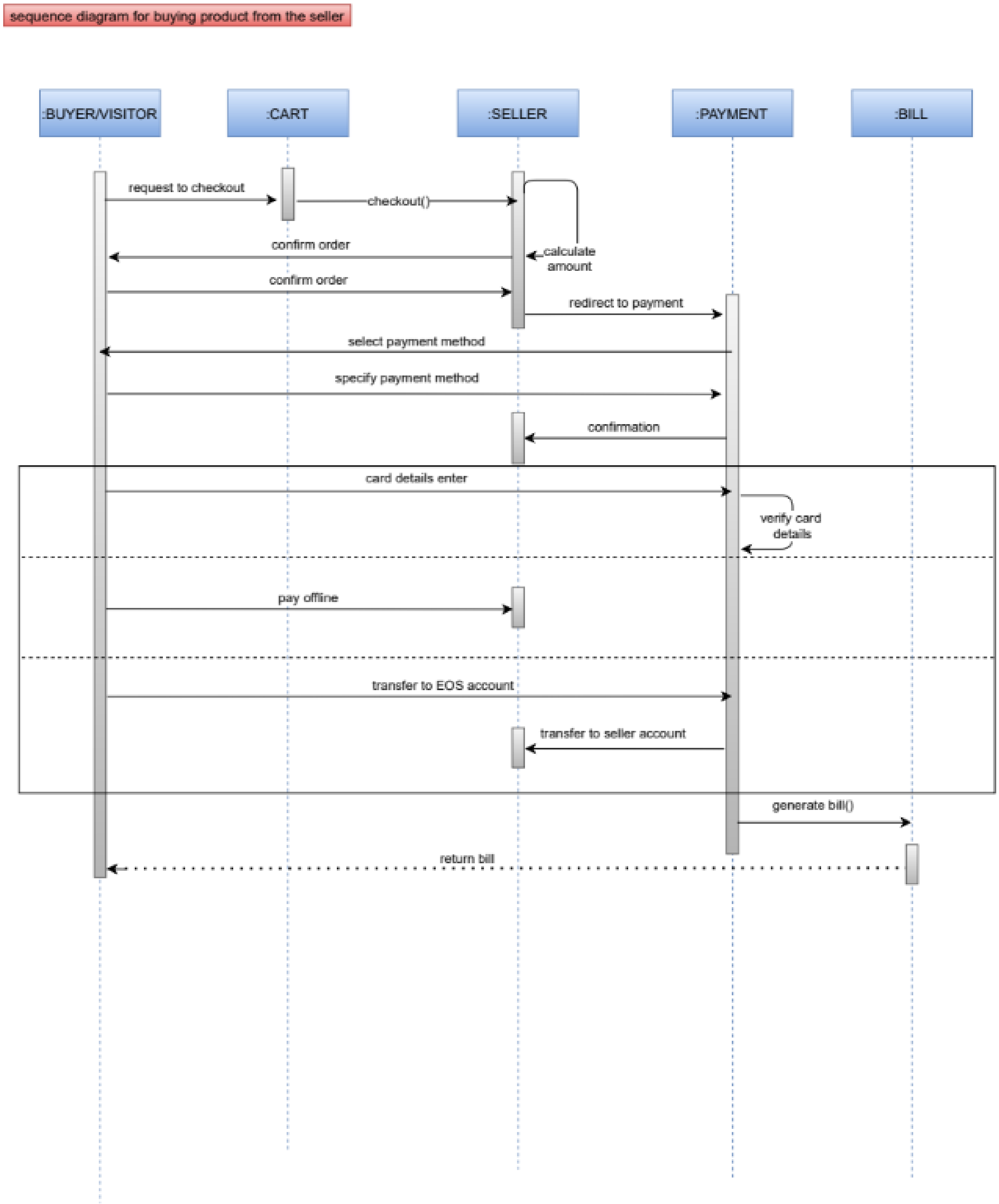


II. Class Diagram

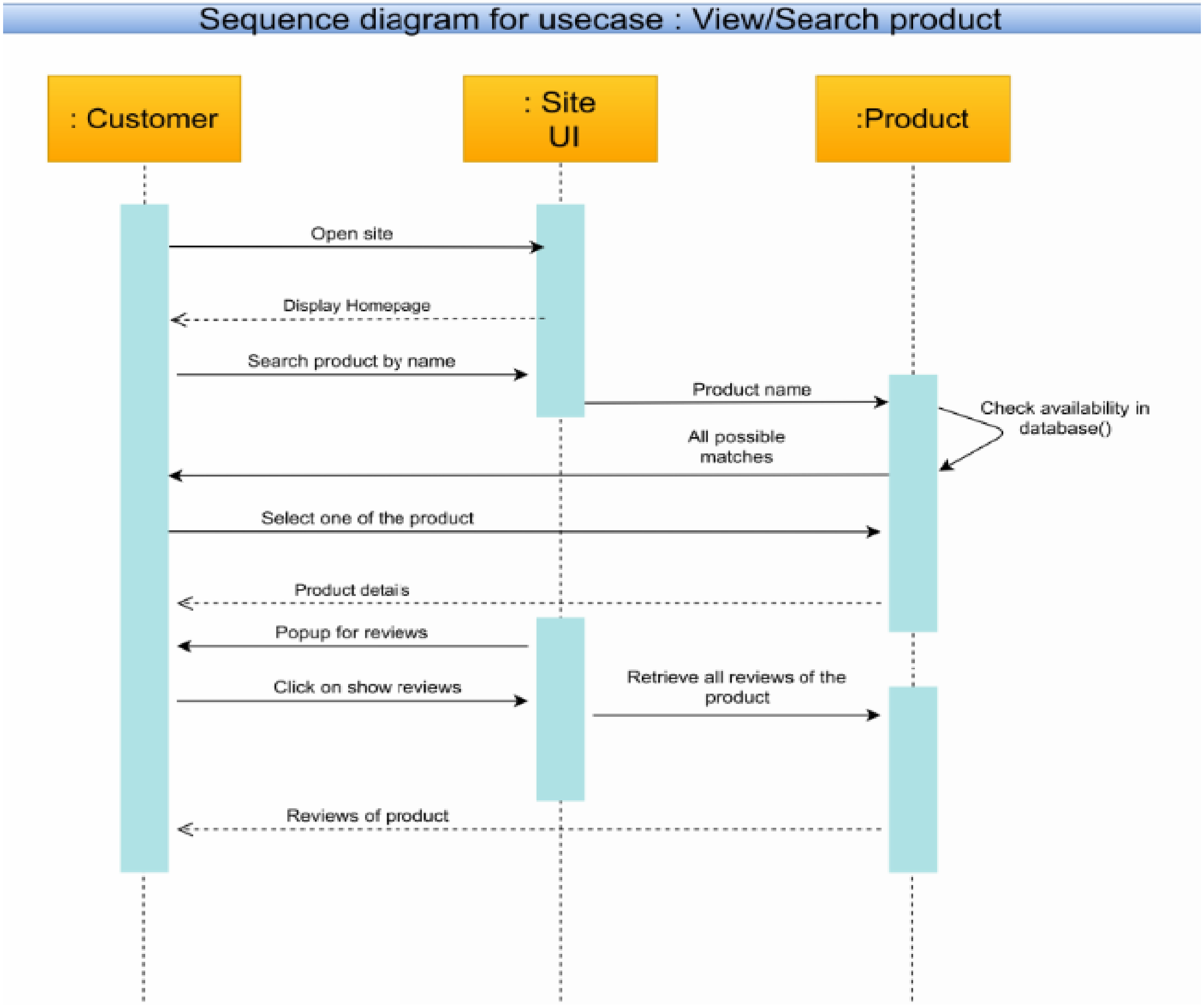


AIM: Identify classes of your project and create a class diagram for the system.
GROUP: CE091,CE087

III. Sequence Diagram(1)

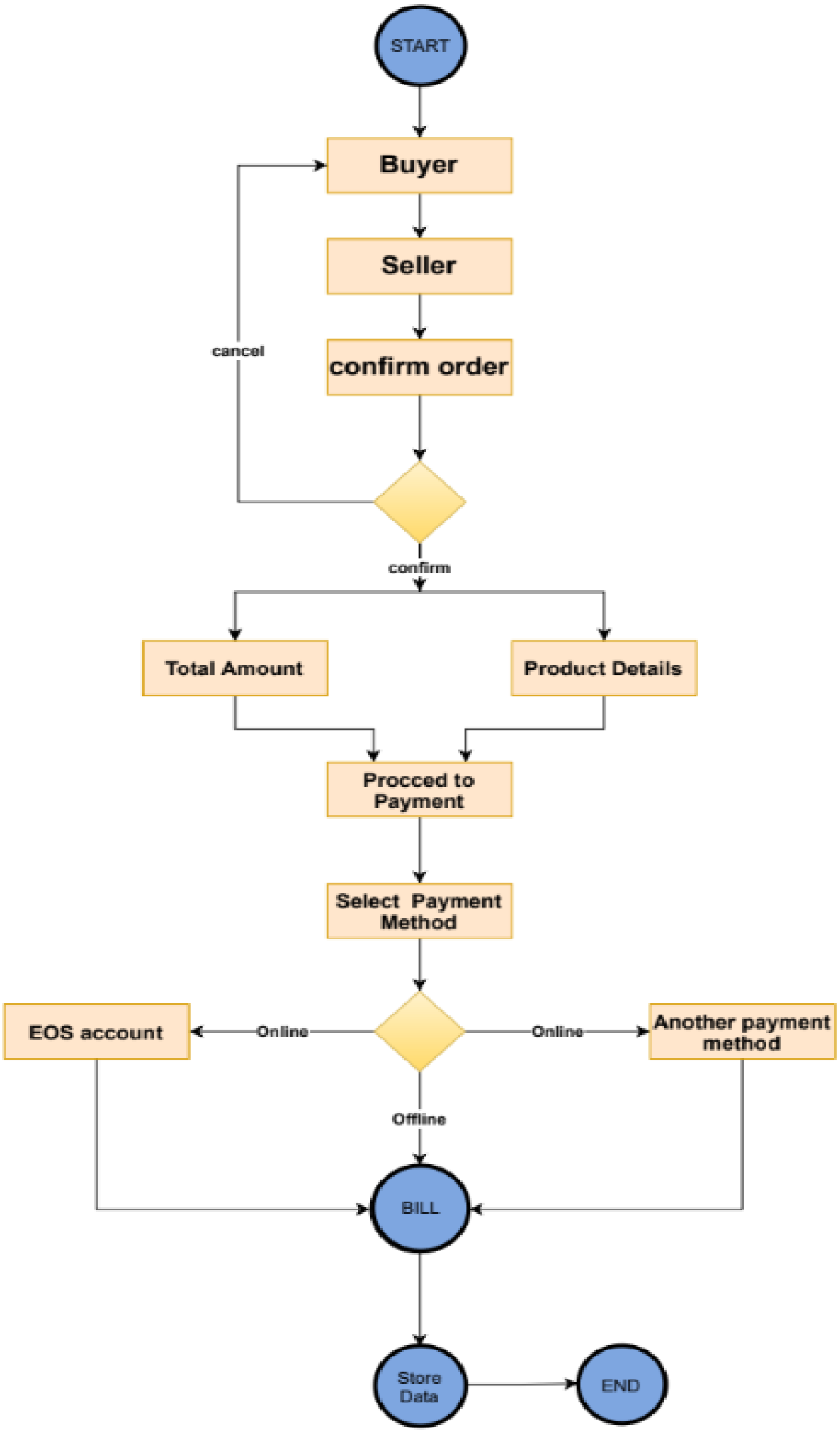


III. Sequence Diagram (2)

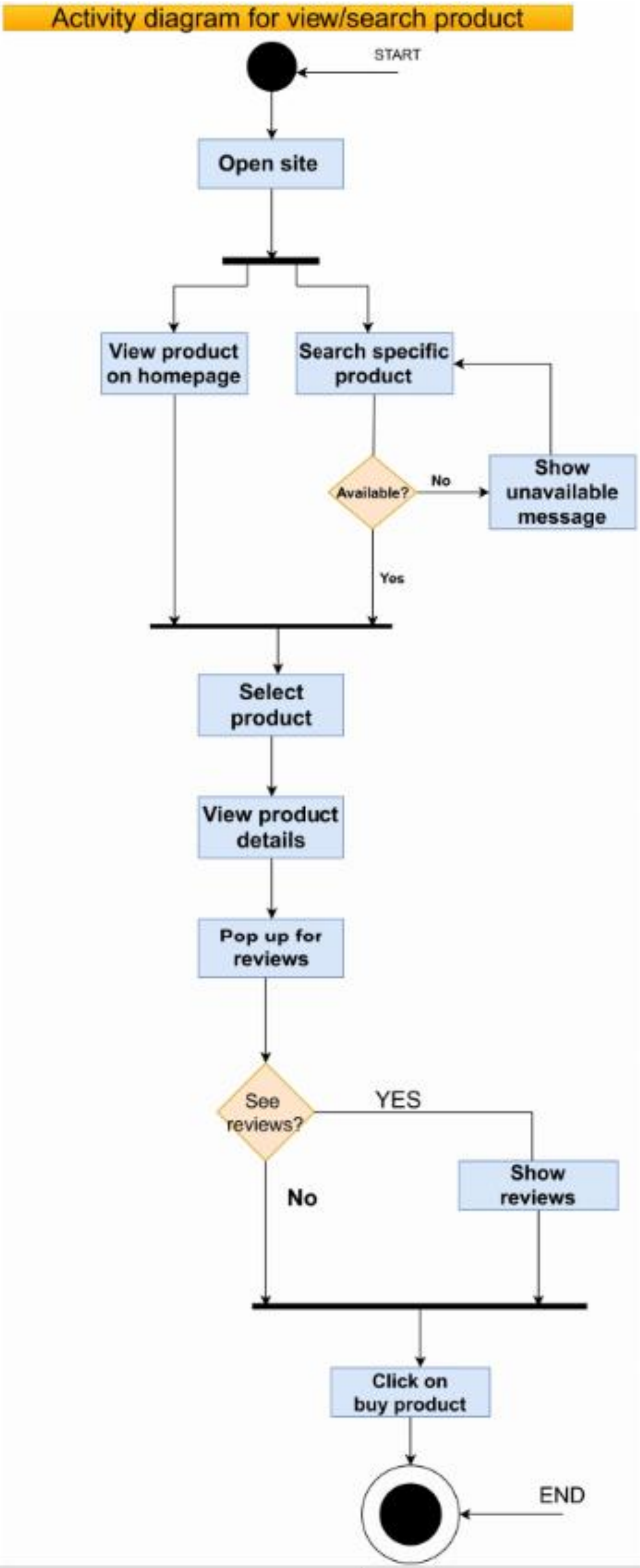


IV. Activity Diagram (1)

ACTIVITY DIAGRAM FOR BUYING PRODUCT FROM SELLER



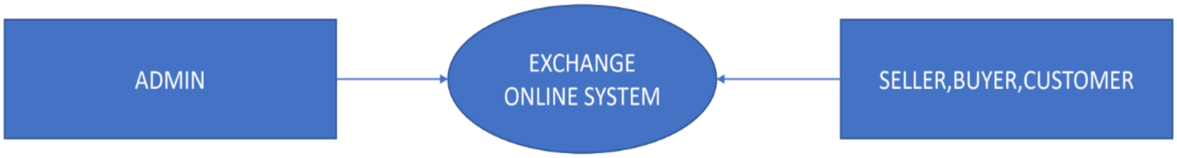
IV. Activity Diagram (2)

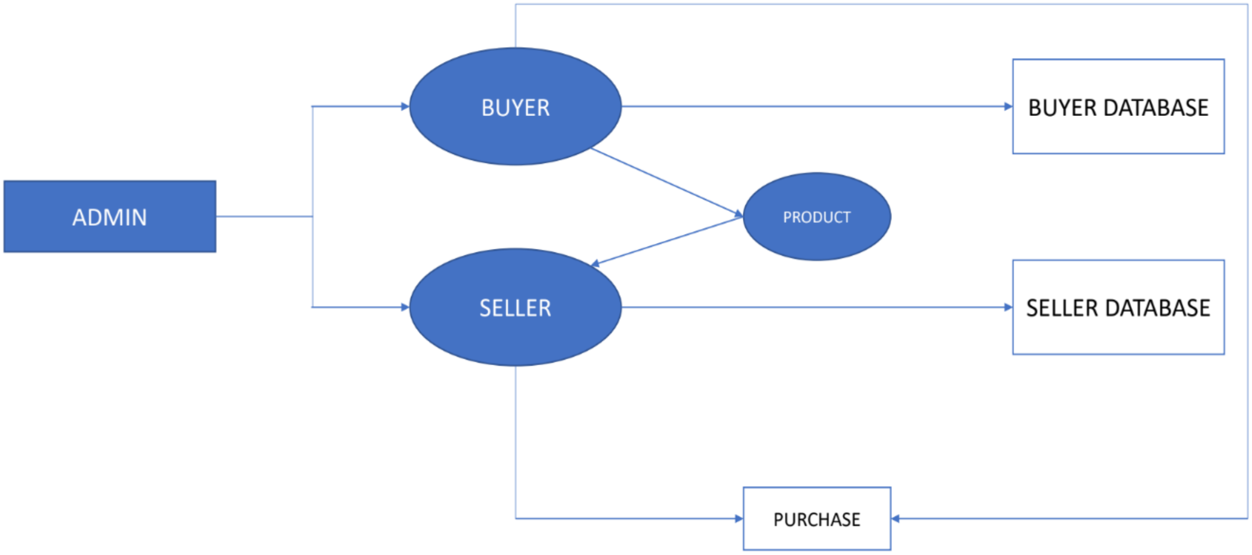


Activity diagram for view/search product

V. Data Flow Diagram(DFD)

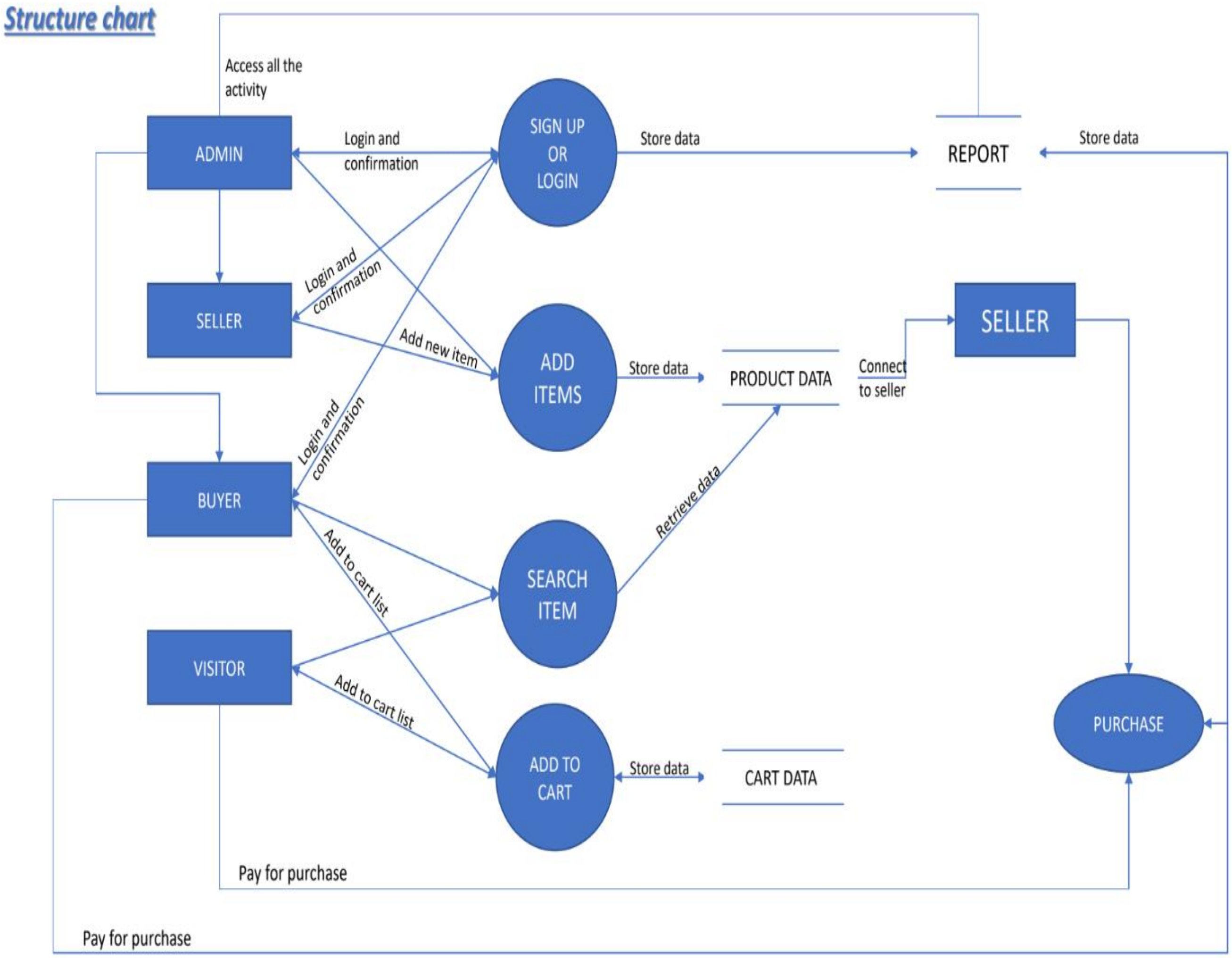
LEVEL ZERO DFD





LEVEL 1 DFD

VI. Structured Chart



5.Implementation Details

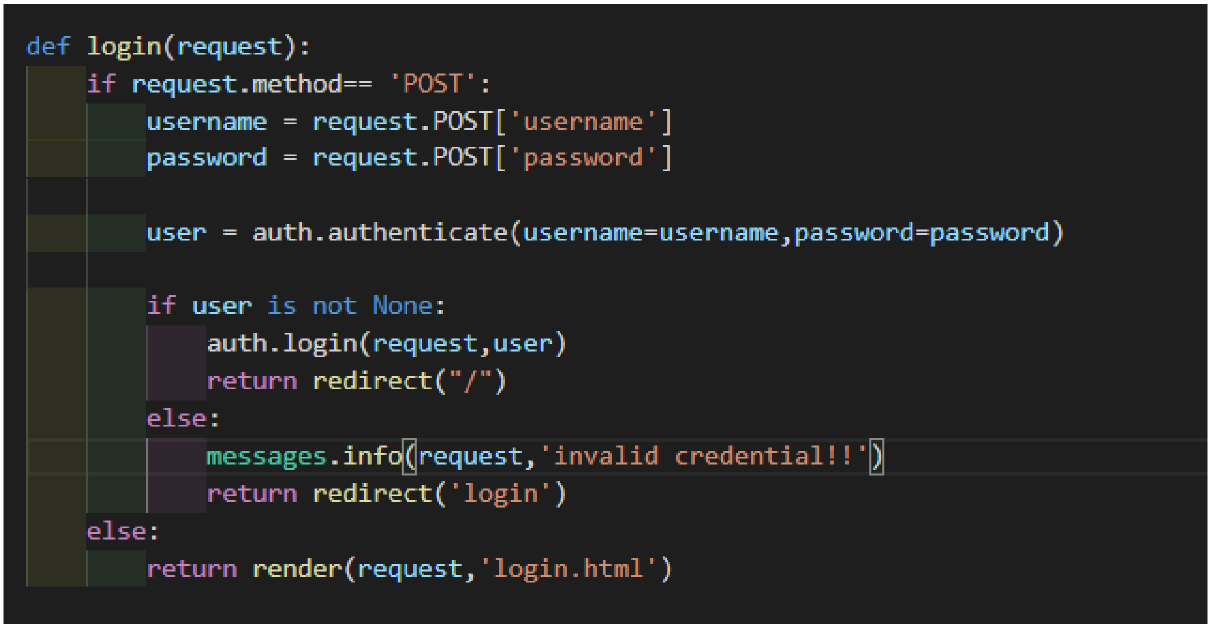
Modules:

1.User module
2.Product module

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

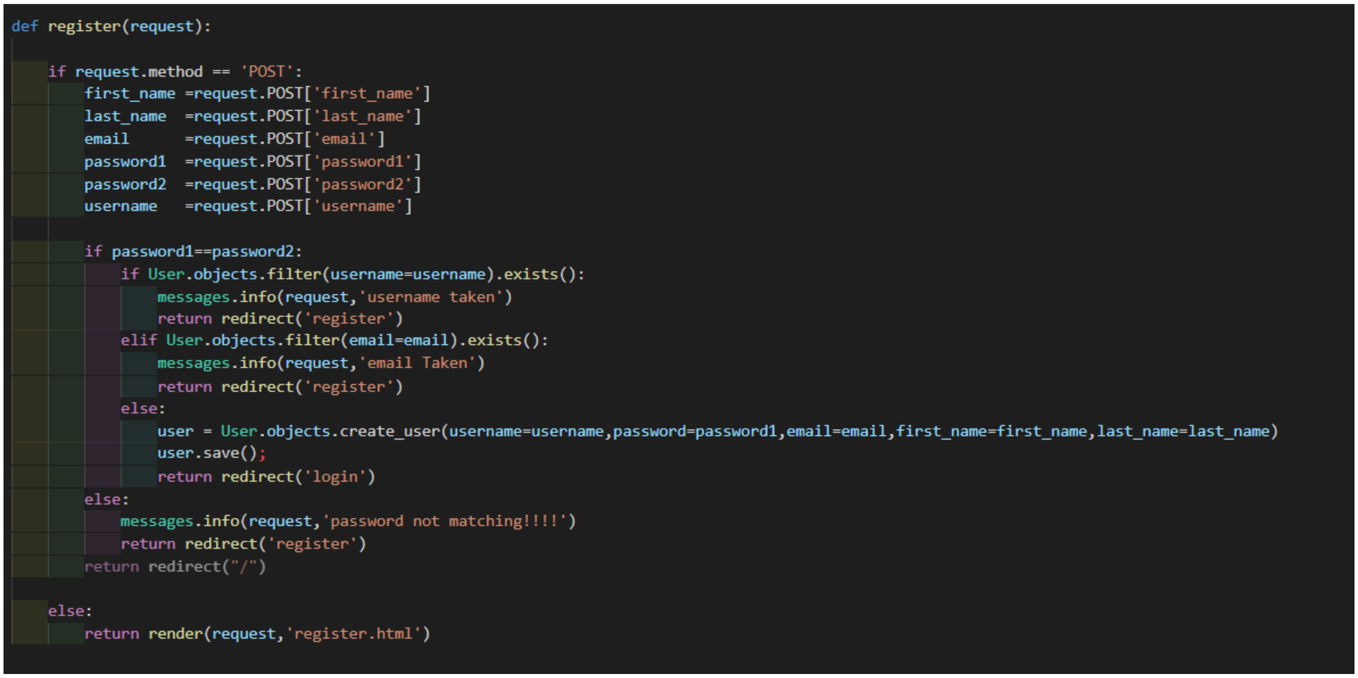
Major Functions:

1) User Login :- User provides login credentials and upon successful authorization is redirected to the homepage.



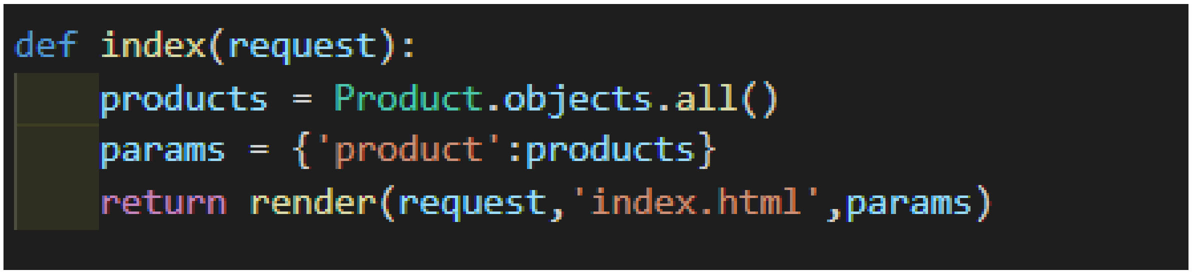
```
def login(request):
    if request.method== 'POST':
        username = request.POST['username']
        password = request.POST['password']

        user = auth.authenticate(username=username,password=password)

        if user is not None:
            auth.login(request,user)
            return redirect("/")
        else:
            messages.info(request,'invalid credential!!')
            return redirect('login')
    else:
        return render(request,'login.html')
```

2) User Registration :- User inputs his/her personal information and it will be stored into database and user redirected to homepage.



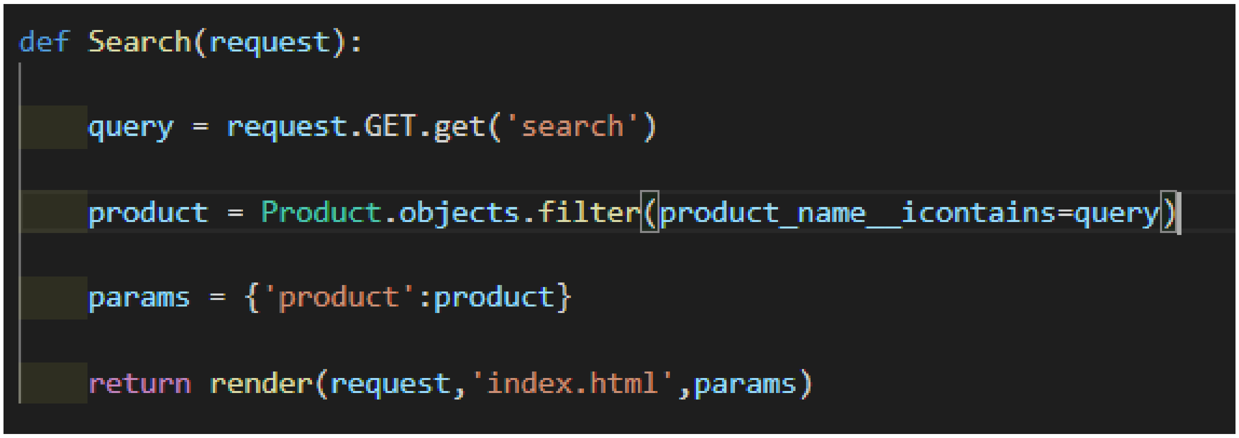
```
def register(request):

    if request.method == 'POST':
        first_name =request.POST['first_name']
        last_name  =request.POST['last_name']
        email      =request.POST['email']
        password1  =request.POST['password1']
        password2  =request.POST['password2']
        username   =request.POST['username']

        if password1==password2:
            if User.objects.filter(username=username).exists():
                messages.info(request,'username taken')
                return redirect('register')
            elif User.objects.filter(email=email).exists():
                messages.info(request,'email Taken')
                return redirect('register')
            else:
                user = User.objects.create_user(username=username,password=password1,email=email,first_name=first_name,last_name=last_name)
                user.save();
                return redirect('login')
        else:
            messages.info(request,'password not matching!!!!')
            return redirect('register')
        return redirect("/")

    else:
        return render(request,'register.html')
```

3) View Products :- All products are shown to user.



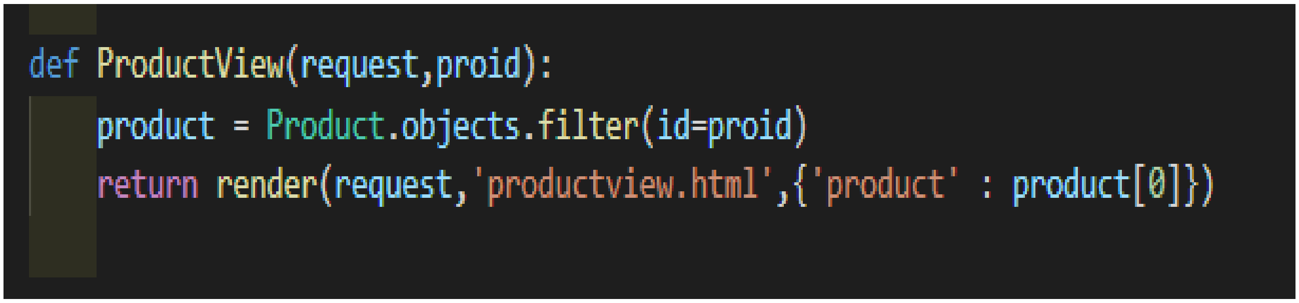
```
def index(request):
    products = Product.objects.all()
    params = {'product':products}
    return render(request,'index.html',params)
```

4) Search :- Customer can view particular product details
or search a specific product.



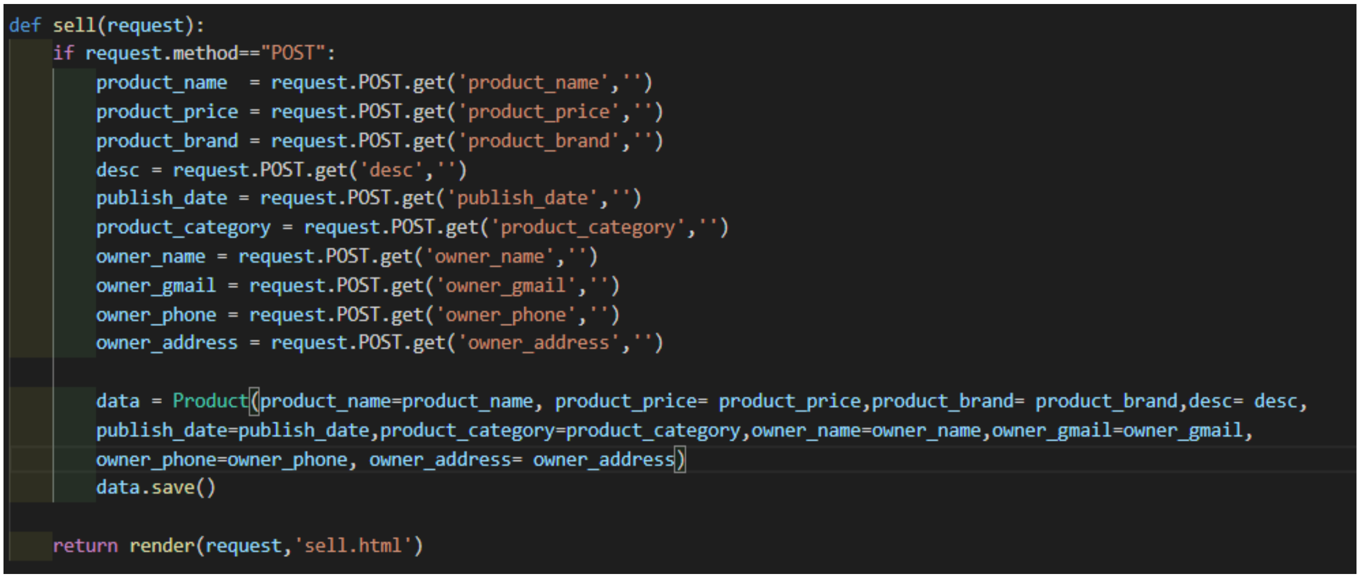
```
def Search(request):

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

    product = Product.objects.filter(product_name__icontains=query)

    params = {'product':product}

    return render(request,'index.html',params)
```

5) Product-view :- If user can click on Quickview then it will show the
Product detail.



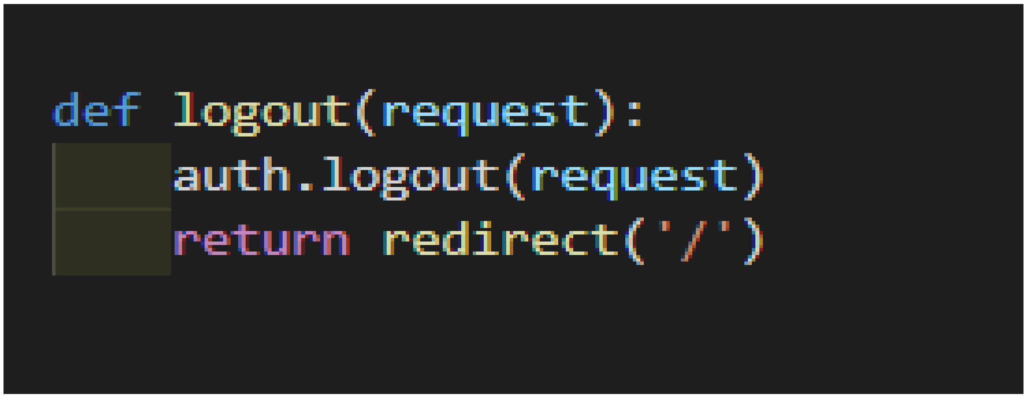
```
def ProductView(request,proid):
    product = Product.objects.filter(id=proid)
    return render(request,'productview.html',{'product' : product[0]})
```

6) Product Ads: If user can click on sell then user will redirect to Product ads form page.



```
def sell(request):
    if request.method=="POST":
        product_name  = request.POST.get('product_name','')
        product_price = request.POST.get('product_price','')
        product_brand = request.POST.get('product_brand','')
        desc = request.POST.get('desc','')
        publish_date = request.POST.get('publish_date','')
        product_category = request.POST.get('product_category','')
        owner_name = request.POST.get('owner_name','')
        owner_gmail = request.POST.get('owner_gmail','')
        owner_phone = request.POST.get('owner_phone','')
        owner_address = request.POST.get('owner_address','')

        data = Product(product_name=product_name, product_price= product_price,product_brand= product_brand,desc= desc,
        publish_date=publish_date,product_category=product_category,owner_name=owner_name,owner_gmail=owner_gmail,
        owner_phone=owner_phone, owner_address= owner_address)
        data.save()

    return render(request,'sell.html')
```

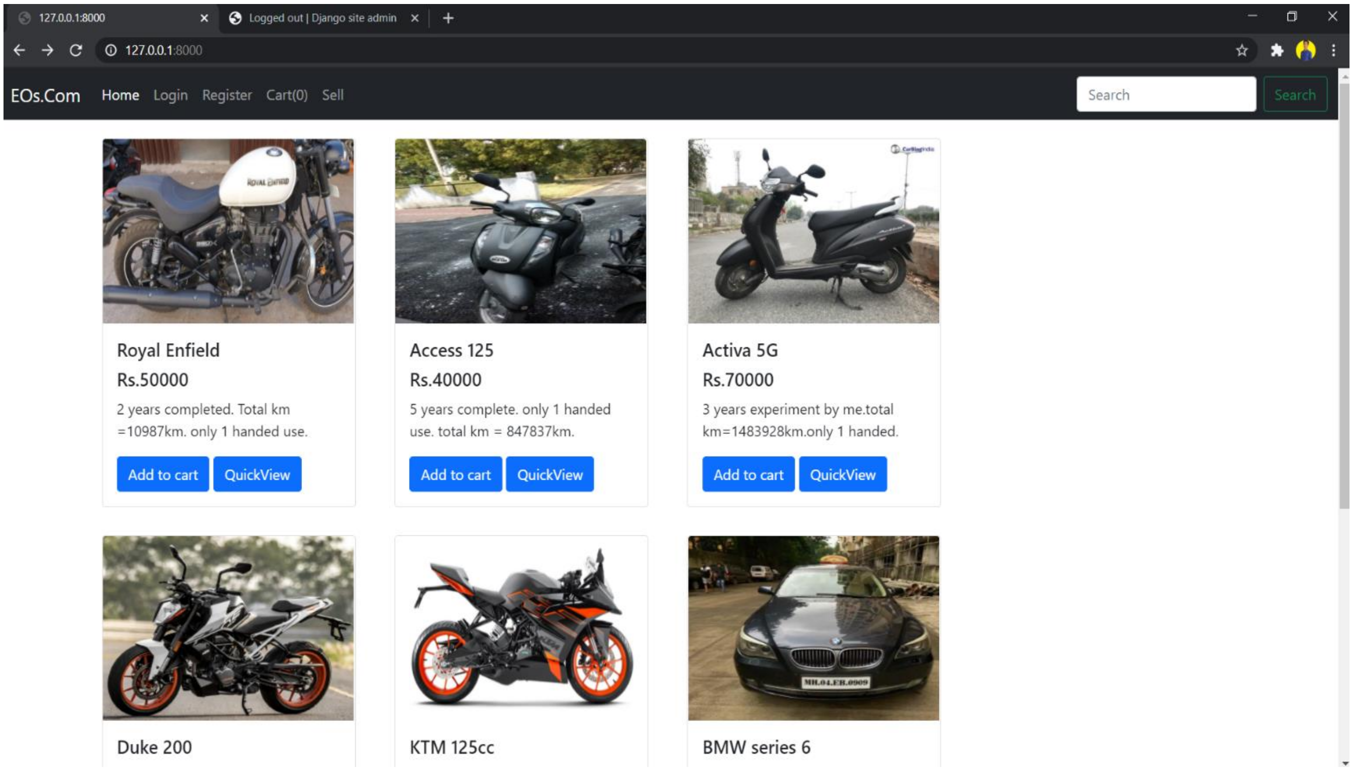
7) User Logout :- If user click on logout button then it will simply logout.



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

6.Workflow/Layout/Screenshots:

I. User/Customer home page



Royal Enfield
Rs.50000
2 years completed. Total km =10987km. only 1 handed use.

Access 125
Rs.40000
5 years complete. only 1 handed use. total km = 847837km.

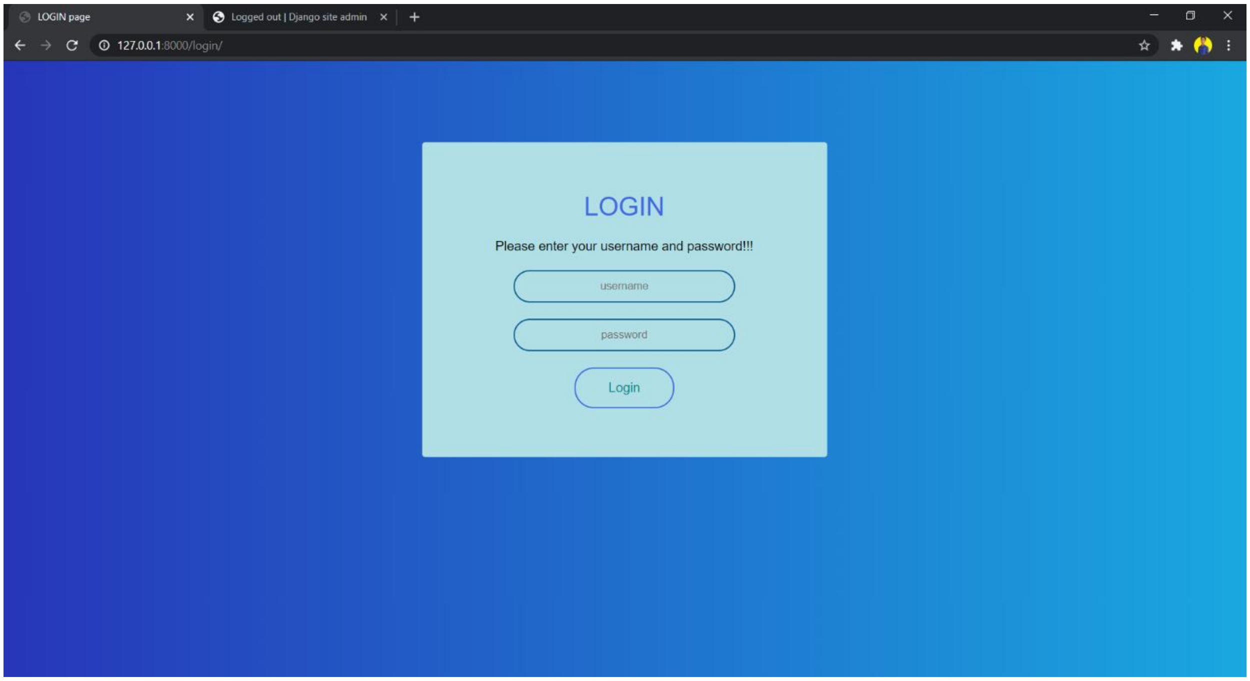
Activa 5G
Rs.70000
3 years experiment by me.total km=1483928km.only 1 handed.

Duke 200

KTM 125cc

BMW series 6

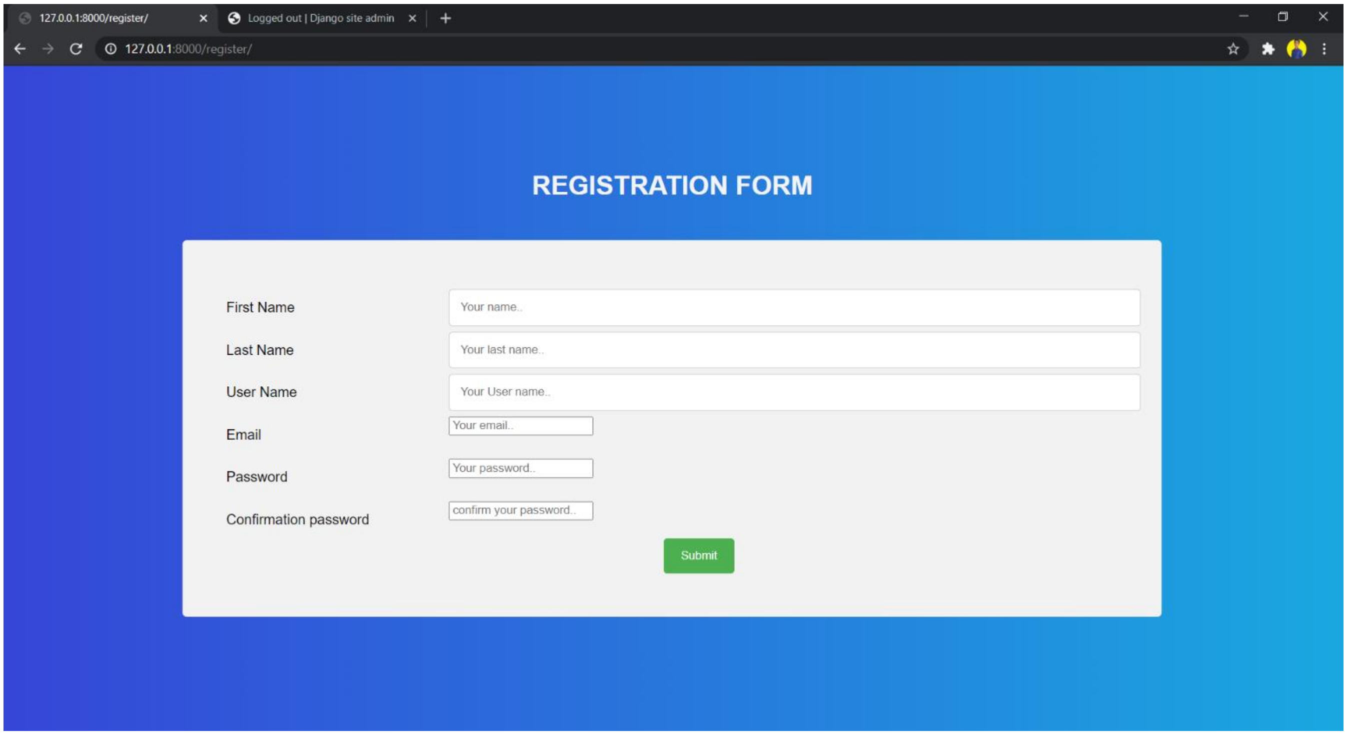
II. Login page



LOGIN

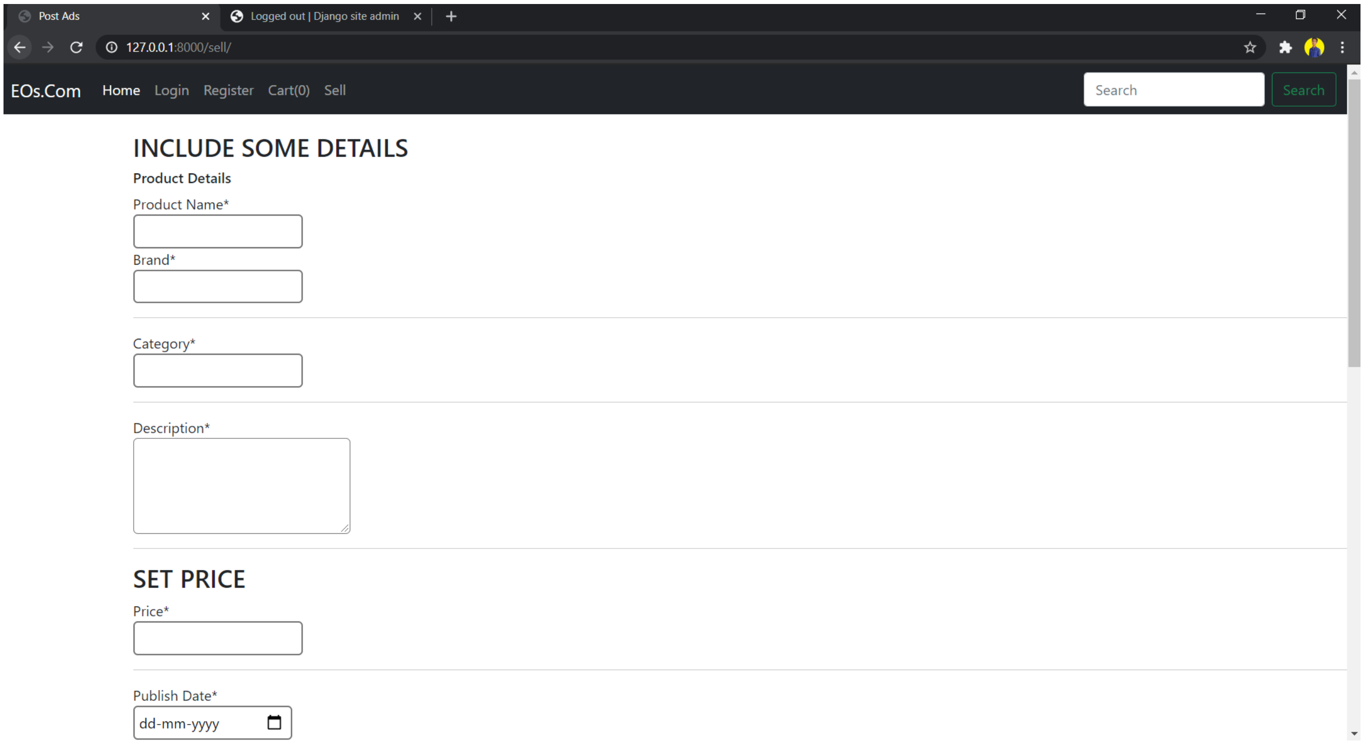
Please enter your username and password!!!

III. Registration Page



REGISTRATION FORM

IV. Post Ads



INCLUDE SOME DETAILS

Product Details

Product Name*

Brand*

Category*

Description*

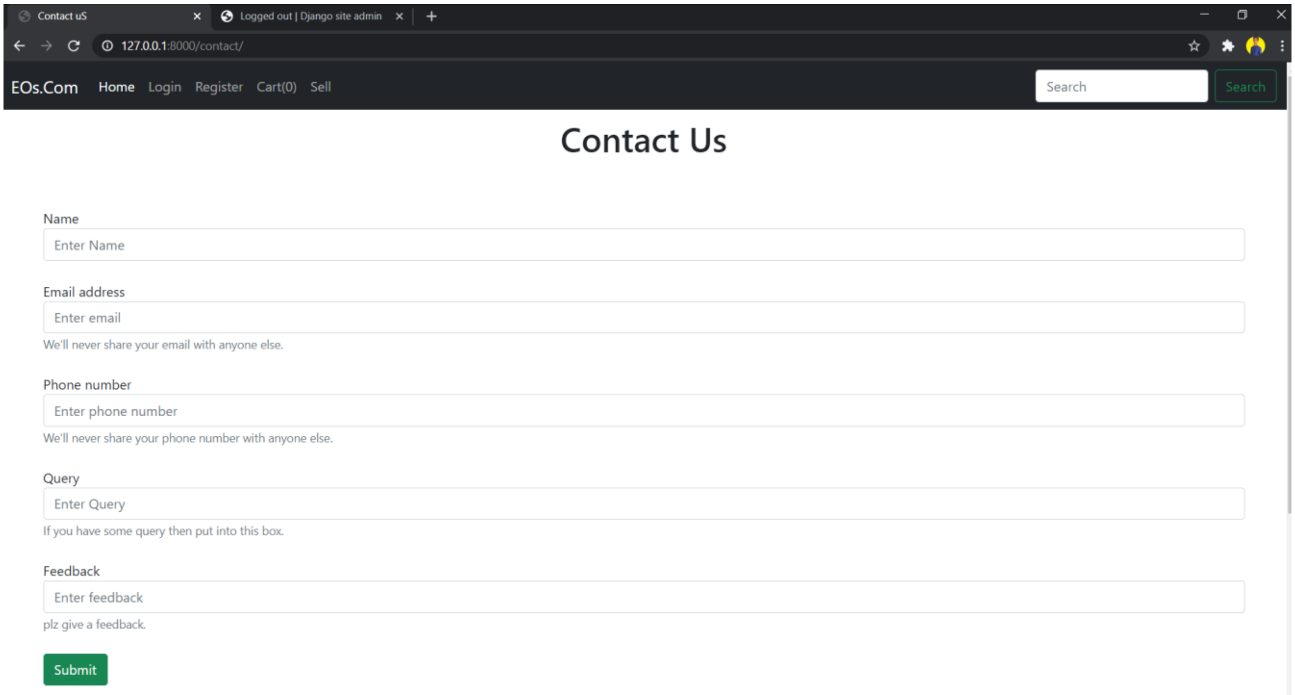
SET PRICE

Price*

Publish Date*

dd-mm-yyyy

V. Contact Us Page



Contact Us

Name

Enter Name

Email address

Enter email

We'll never share your email with anyone else.

Phone number

Enter phone number

We'll never share your phone number with anyone else.

Query

Enter Query

If you have some query then put into this box.

Feedback

Enter feedback

plz give a feedback.

7.Conclusion

Hereby, we conclude that we have successfully implemented the admin, buyer, seller module.

Admin will be able to add/remove the category and products.

Customer will be able to view, search the products, add them to the cart, place an order, give a review to seller for individual products, view reviews of other buyer product.

The seller will be able to confirm the 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.

8.1 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 Buyer can see at what place their order is.

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

4] Improve UI for better user interaction.

9. Bibliography

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