A
PROJECT REPORT
ON

ONLINE PHARMACY STORE (HEALTHOCARE)

By

CHAUHAN RIKIN DIPAKKUMAR (CE-016) (19CEUOS106)
SHAH VIDIT PARESH (CE-002) (19CEUON075)

B.Tech CE Semester-IV
Subject: Software Engineering Principles and Practices (SEPP)

Guided by:
Dr. Brijesh S. Bhatt Professor
Prof. Jigar M. Pandya Assistant Professor
Prof. Ankit P. Vaishnav Assistant Professor
Prof. Pinkal C. Chauhan Assistant Professor
Dept. of Computer Engg.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

CERTIFICATE

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

Software Engineering Principles and Practices and recorded

in this journal is the bonafide work of

CHAUHAN RIKIN DIPAKKUMAR (CE-016) (19CEUOS106)
SHAH VIDIT PARESH (CE-002) (19CEUON075)

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

during the academic year **2020-2021**.

Project Guide:

Dr. Brijesh S. Bhatt
Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Prof. Jigar M. Pandya
Assistant Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Prof. Ankit P. Vaishnav
Assistant Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

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

Head of the Department
Dr. C. K. Bhensdadia,
Professor & Head,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Table of Contents

1. Abstract

"HEALTHOCARE" is an online pharmacy store. At healthocare.com, we make a wide range of prescription medicines and other health products conveniently available all across India. Since we also offer generic alternatives to most medicines, online buyers can expect significant savings. At healthocare.com, we not only provide you with a wide range of medicines listed, we also offer a wide choice of over the counter (OTC) products including wellness products, vitamins, diet / fitness supplements, herbal products, pain relievers, diabetic care kits, baby / mother care products, beauty care products and surgical supplies.

2. Introduction

2.1 Brief Introduction

In this fast-moving world it is necessary to develop online pharmacies because they offer better pricing than offline stores with increased access, lower transaction and product costs, convenience and greater anonymity for consumers. They offer accessibility to people with limited mobility and people in remote areas. These also provide media-alerts (personalized medicine reminder service), discounts, doorstep delivery within a short time, and validation of prescription through licensed pharmacists. Information about substitutes and adverse effects is also available on these sites.

2.2 Tools/Technologies Used

Technologies:

- Django
- Python
- MySQL
- Bootstrap
- JavaScript
- JSON
- HTML

Tools:

- Git
- IDLE Shell 3.9.1
- Notepad++

Platform:

- Local development server

3. Software Requirement Specifications

3.1 Product Scope

This system is designed to provide the user online pharmacy services.

3.2 System Functional Requirements

R.1 Manage User

R.1.1 Sign up / Register

Description: User can register.

Input: User details.

Output: Login page is displayed If user is successfully registered.

Processing: User details verification and validation.

Next functions: R.1.2 if user is registered and R.1.1 if user is not registered due to some reason.

R.1.2 Login

Description: User logins to the system by entering valid user-id and password.

Input: User-id and Password.

Output: Home page is displayed if user successfully logs in.

Processing: User-id and Password validation.

R.1.3 Password Reset

Description: This function lets the user to reset its password.

Input: Select “Forgotten Password?” option.

Output: It prompts the user to enter valid email-id / mobile number to locate their account.

Processing: Validates the email-id / mobile number and sent the reset password link to the provided email-id / mobile number.

Next functions: R.1.2 if password is successfully changed and R.1.3 if password is not changed.

R.1.4 Password Change

Description: Users can change their login password.

Input: Select “Password Change” option

Output: Confirmation Message (i.e., Password is successfully changed / updated)

R.1.5 Mobile Number Change

Description: Users can change their mobile number.

Input: Select “Mobile Number Change” option

Output: Confirmation Message (i.e., Mobile Number is successfully changed / updated)

R.1.6 Delivery Address Change

Description: Users can change their delivery address.

Input: Select “Delivery Address Change” option

Output: Confirmation Message (i.e., Delivery Address is successfully changed / updated)

R.2 Manage Product

R.2.1 Search Product

Description: Users can search for any pharmaceutical product.

Input: Select “Search” option

Output: User prompted to enter key words

Processing: Search the products based on the keywords entered

Next functions: R.2.2 based on keywords matched

R.2.2 Product Description

Description: Users gets the description of a product.

Input: “Select” option

Output: Detailed description of the product.

R.2.3 Add Product

Description: Admin / Seller can add any product which is out of stock / unavailable.

Input: “Add product” option

Output: Confirmation Message (i.e., Product successfully added)

R.2.4 Product Feedback

Description: User can provide feedback about any product.

Input: Select “Give Feedback” option

Output: Confirmation Message (i.e., Thanks for your feedback)

R.3 Manage Order

R.3.1 Buy Product

Description: The user can buy medicine only if he is a registered user and has logged into the system.

R.3.1.1 Buy now

Description: User can directly buy the product without adding it to cart

Input: Select “Buy Now” option

Output: Redirected to place your order page

Next functions: R.3.2 if the selected product is in stock

R.3.1.2 Add to cart / Remove from cart

Description: Here the user can add the product to the cart and can buy at later stage. Add to cart option helps the customer to buy more than one item at a time. For adding any particular item to the cart, the customer has to click the “+” symbol against the product and that will be added to the cart. Similarly, for removing any product from the cart the customer has to go to his cart and click “-” against the product which he wants to remove.

Input: Select “Add to cart / Remove from cart” option

Output: Redirected to place your order page

Next functions: R.3.2 if all selected products are in stock

R.3.2 Place your order

Description: This function helps the customer to review all the products that he wants to buy before making payment and it also displays the bill which includes price of all individual product, their taxes and delivery charges.

Input: Select “place your order” option

Output: Redirected to payment page

Next functions: R.3.3 if user wants to check their order status and R.3.4 if user wants to cancel the order placed

R.3.3 Order Status

Description: User can check their order status.

Input: Select “order status” option

Output: Displays the status of your order

Next functions: R.3.4 if user wants to cancel the order placed and R.3.5 if the user wants to return the order received

R.3.4 Cancel Order

Description: User can cancel their order

Input: Select “cancel order” option

Output: Confirmation Message (i.e., order is successfully cancelled)

R.3.5 Return Order

Description: User can return their order

Input: Here the user has to specify the reason for returning the order and has to choose whether they want to exchange it or want to return it.

Output: Depending on the input, appropriate message is displayed on the screen.

Status: After successful exchanged / return of the product the status of order (i.e., new order) / refund is shown to the user.

3.3 Other Non-functional Requirements

1. Performance

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

2. Safety

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

3. Reliability

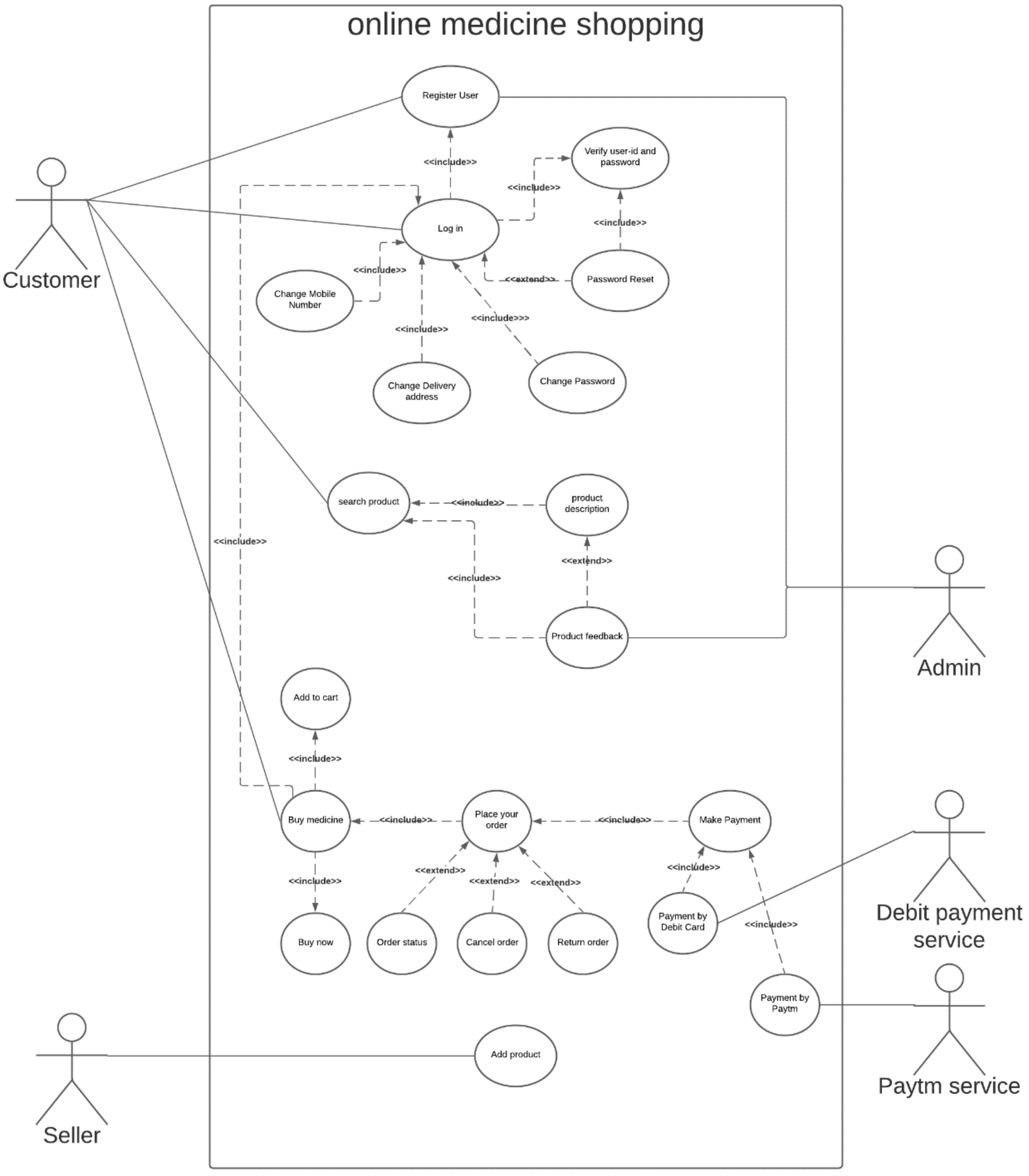
As the system has personal data, its reliability is the major factor for consideration.

4. Database

System requires to access user profile data fast to maintain the performance.

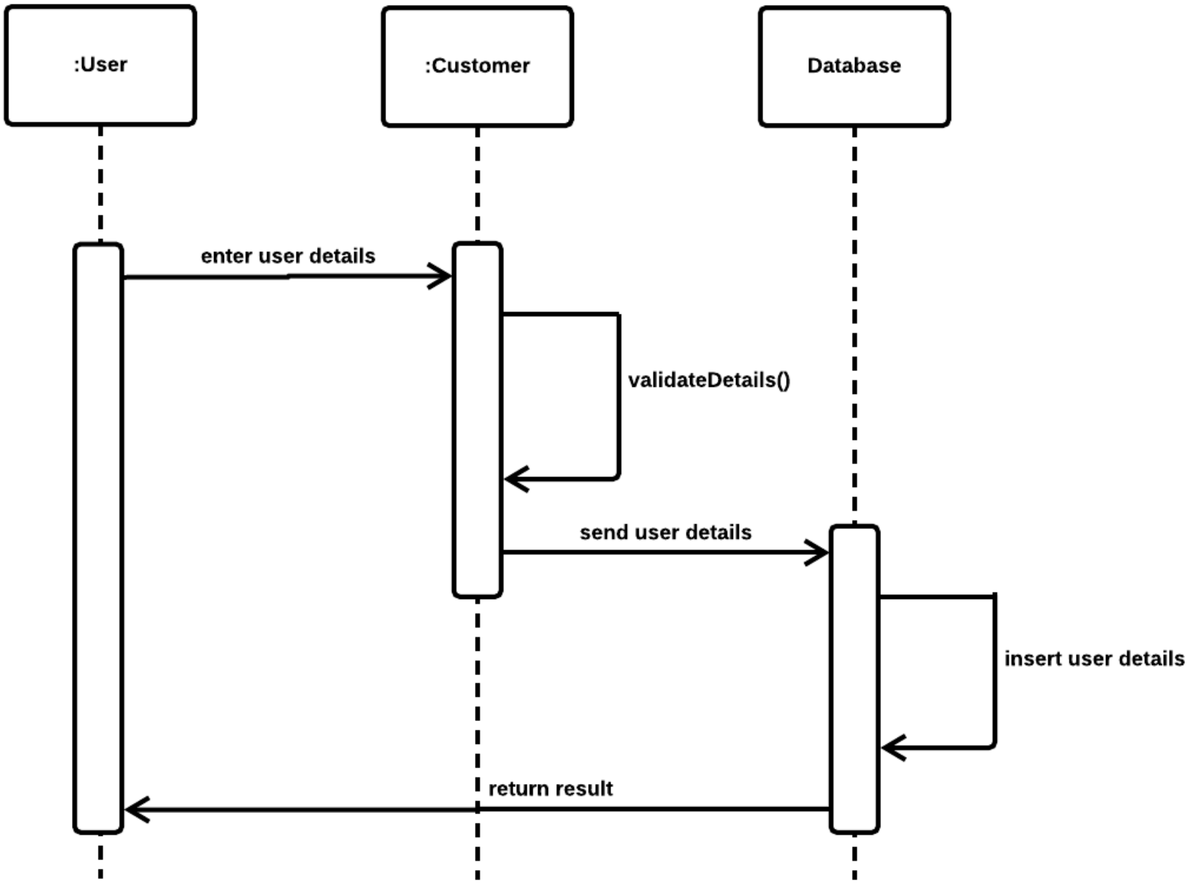
4. Design

4.1 Use Case Diagram

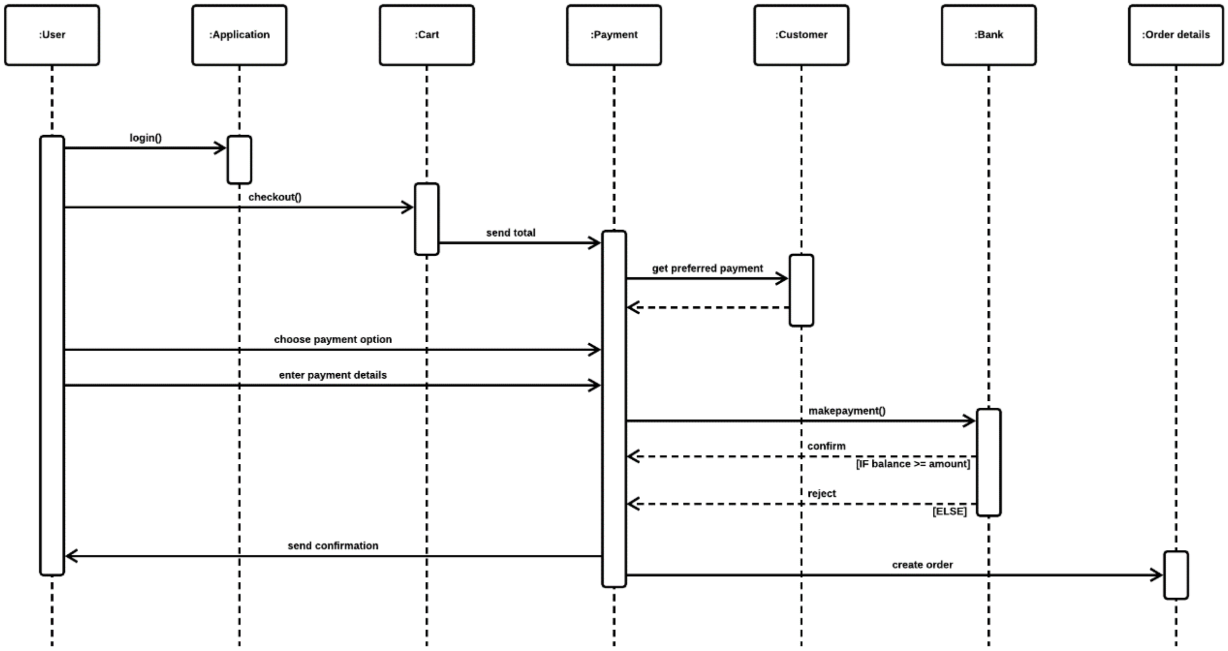


Use case diagram

4.2 Sequence Diagram

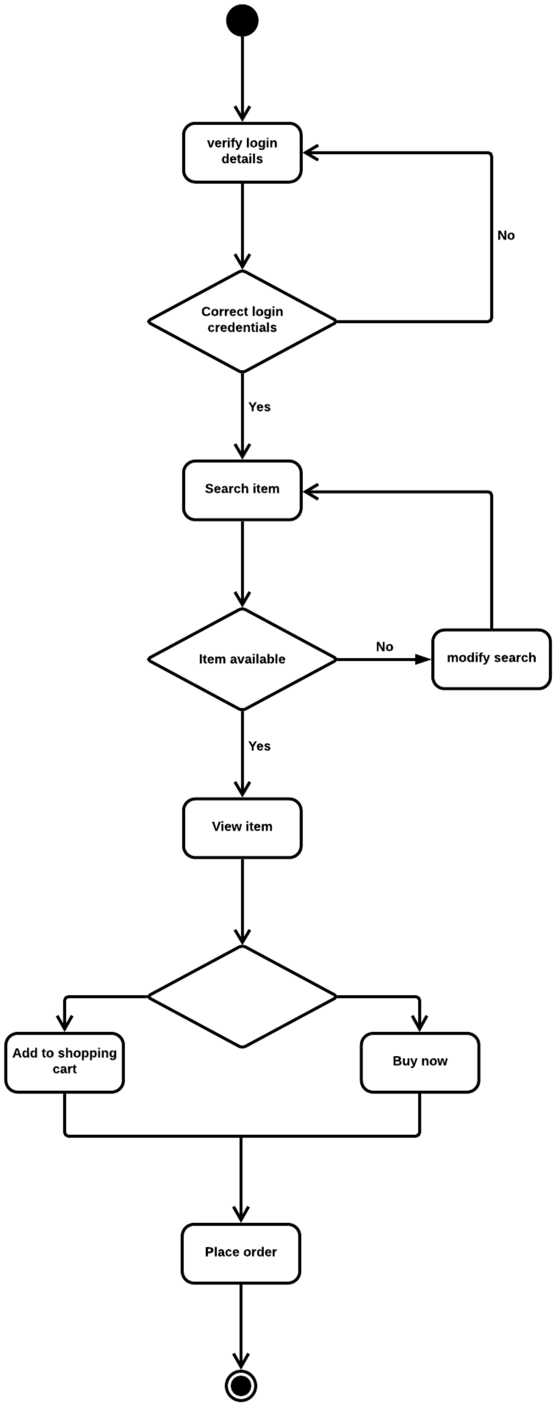


For “Register user” use case

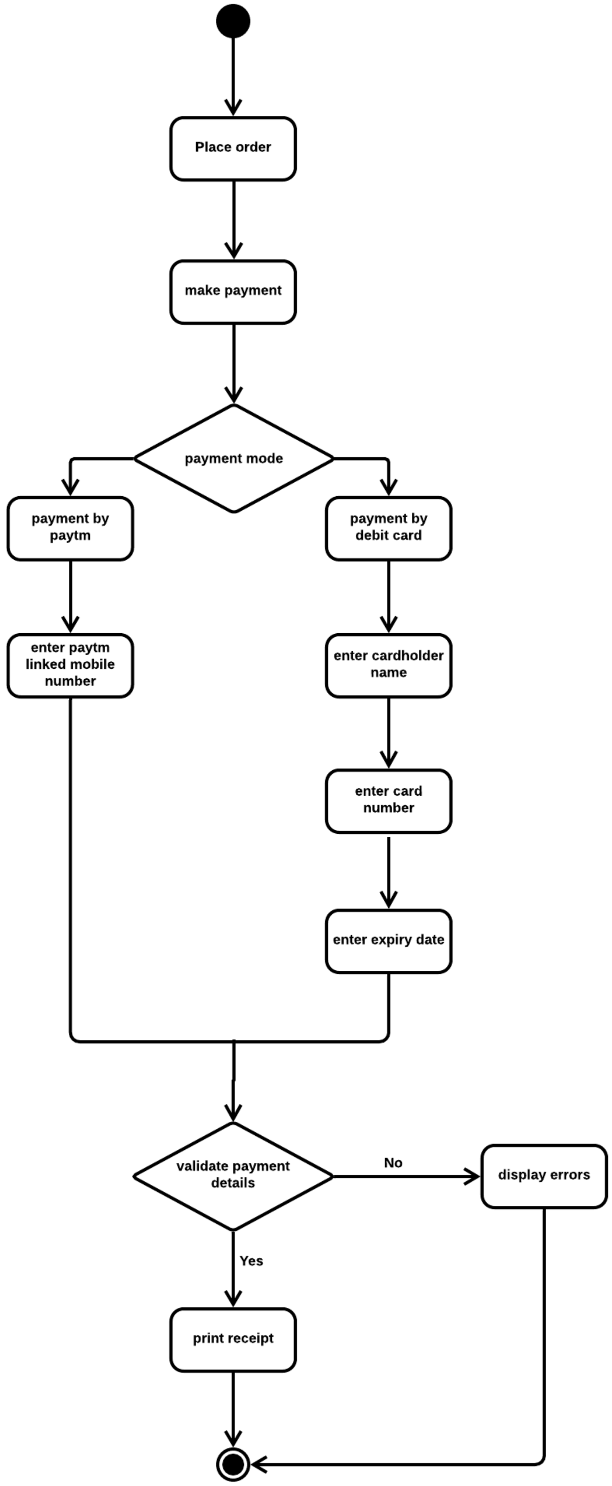


For “make payment” use case

4.3 Activity Diagram

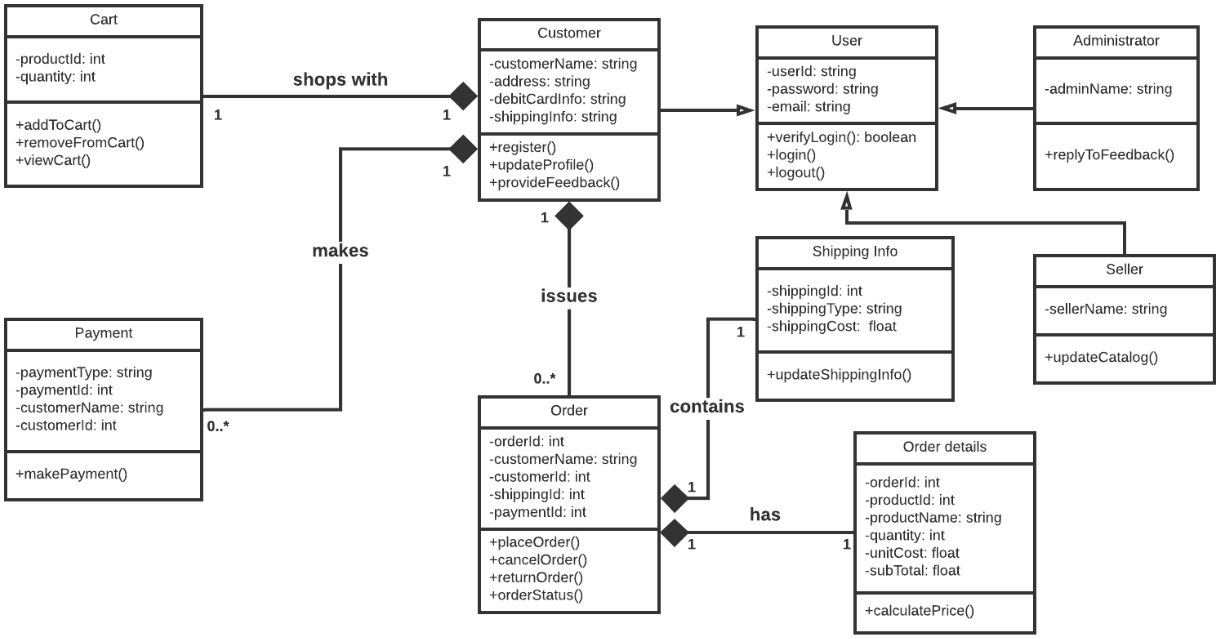


For “place order” use case



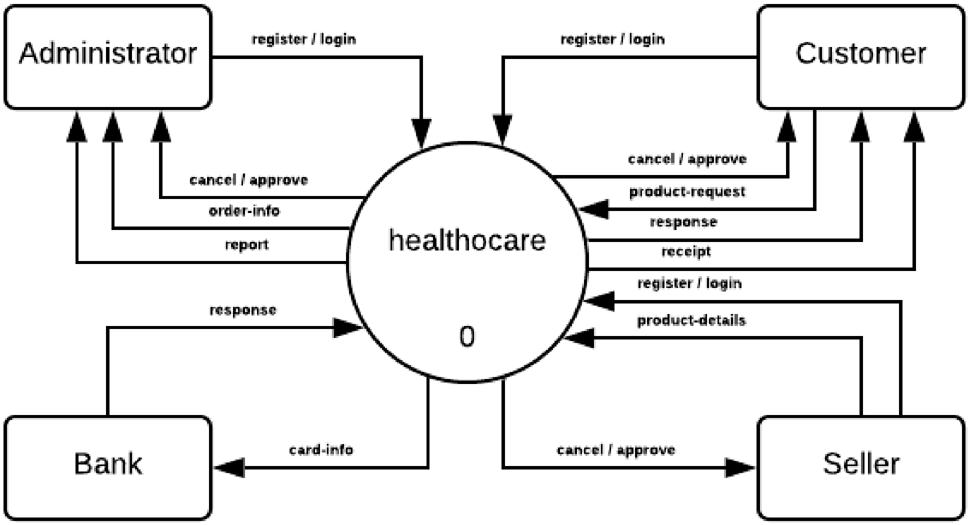
For "make payment" use case

4.4 Class Diagram

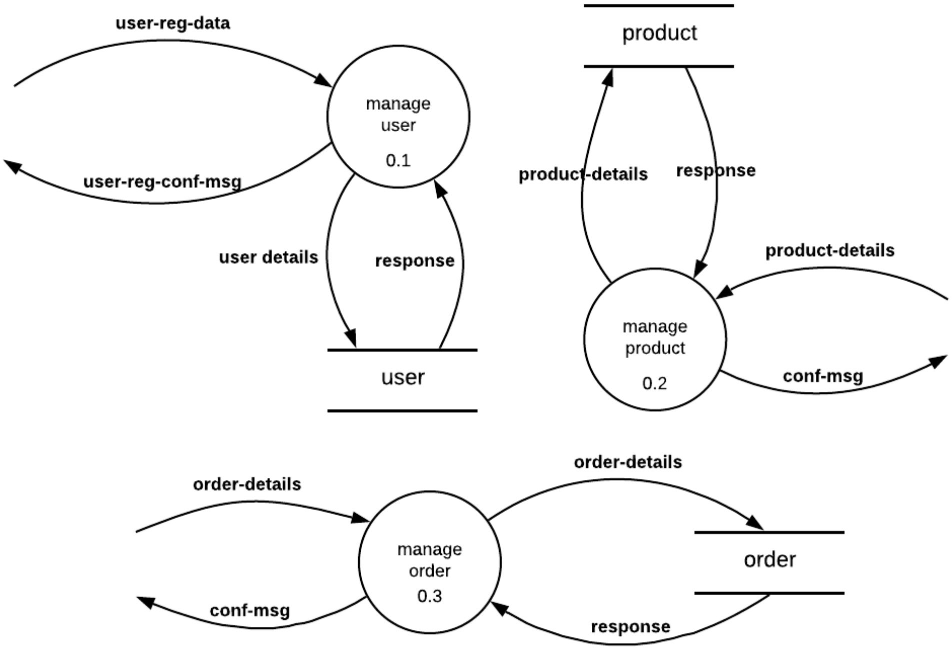


Class diagram

4.5 DFD Model

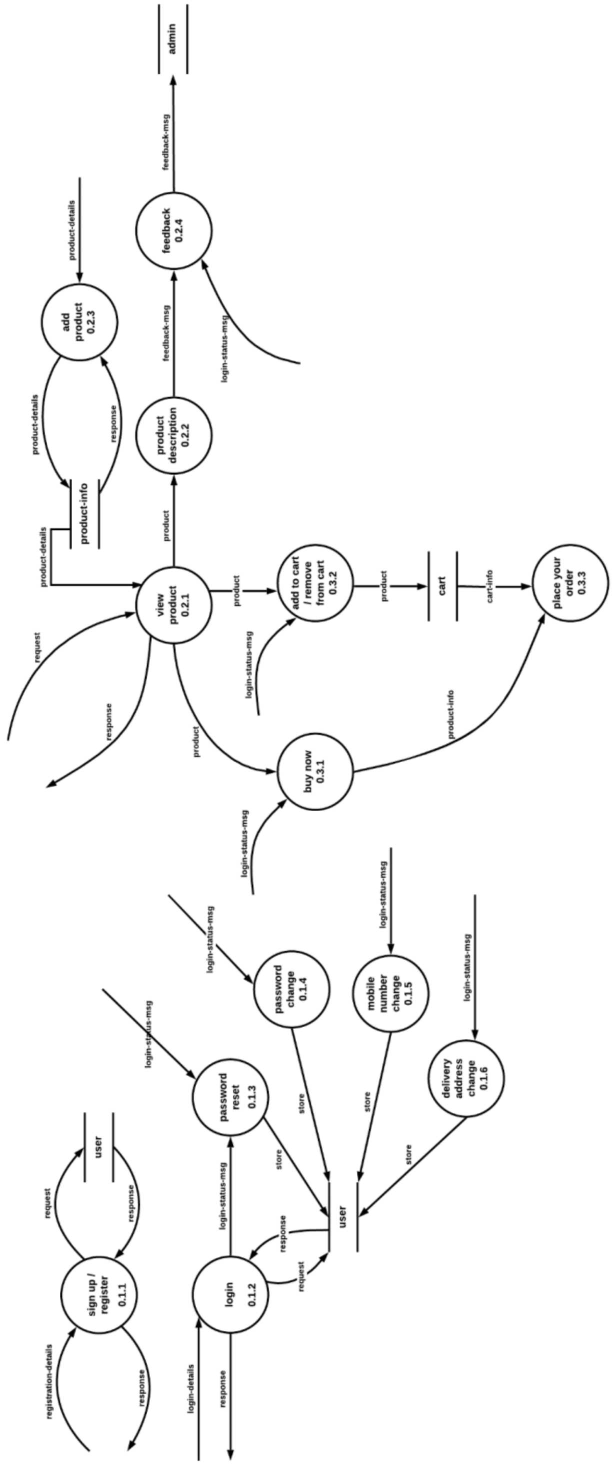


Level 0 DFD / Context Diagram



Level 1 DFD

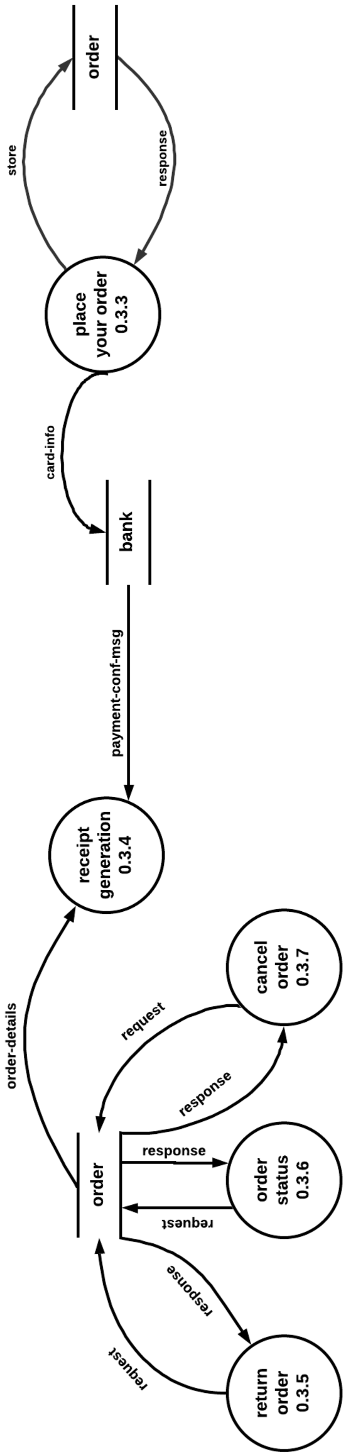
Level 2 DFD



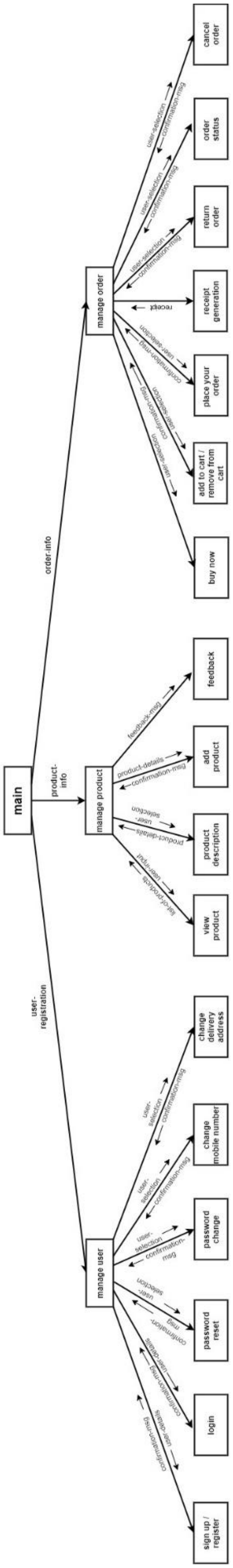
admin
feedback-msg
feedback
0.2.4
product-details
add
product
0.2.3
feedback-msg
login-status-msg
product-details
response
product
description
0.2.2
product-info
product-details
product
view
product
0.2.1
product
add to cart
/ remove
from cart
0.3.2
product
cart
cart-info
place your
order
0.3.3
request
login-status-msg
response
product
product-info
buy now
0.3.1
login-status-msg
login-status-msg
login-status-msg
login-status-msg
login-status-msg
password
change
0.1.4
mobile
number
change
0.1.5
delivery
address
change
0.1.6
password
reset
0.1.3
store
store
store
store
user
request
response
login-status-msg
response
request
user
sign up /
register
0.1.1
login
0.1.2
registration-details
response
login-details
response

Level 2 DFD (Continue)

Note: From <u>place your order</u> bubble the Level 2 DFD continue



4.6 Structure Chart



4.7 Data Dictionary

auth_user									
Sr no.	Field Name	Data Type	Width	Unique	Null	Default	Primary Key / Foreign Key	Referred Table	Description
1	id	int	11	Yes	No	None	Yes	-	AUTO_INCREMENT
2	password	varchar	128	No	No	None	-	-	-
3	last_login	datetime	6	No	Yes	NULL	-	-	-
4	is_superuser	tinyint	1	No	No	None	-	-	-
5	username	varchar	150	Yes	No	None	-	-	-
6	first_name	varchar	150	No	No	None	-	-	-
7	last_name	varchar	150	No	No	None	-	-	-
8	email	varchar	254	No	No	None	-	-	-
9	is_staff	tinyint	1	No	No	None	-	-	-
10	is_active	tinyint	1	No	No	None	-	-	-
11	date_joined	datetime	6	No	No	None	-	-	-

Best_deals

Sr no.	Field Name	Data Type	Width	Unique	Null	Default	Primary Key / Foreign Key	Referred Table	Description
1	id	int	11	Yes	No	None	Yes	-	AUTO_INCREMENT
2	img	varchar	100	No	No	None	-	-	-
3	name	varchar	1000	No	No	None	-	-	-
4	desc	longtext	-	No	No	None	-	-	-
5	mfr	varchar	5000	No	No	None	-	-	-
6	price	Double	-	No	No	None	-	-	-
7	offer	int	11	No	No	None	-	-	-
8	offer_price	double	-	No	No	None	-	-	-

diagnostic_packages

Sr no.	Field Name	Data Type	Width	Unique	Null	Default	Primary Key / Foreign Key	Referred Table	Description
1	id	int	11	Yes	No	None	Yes	-	AUTO_INCREMENT
2	img	varchar	100	No	No	None	-	-	-
3	name	varchar	1000	No	No	None	-	-	-
4	desc	longtext	-	No	No	None	-	-	-
5	price	double	-	No	No	None	-	-	-

order									
Sr no.	Field Name	Data Type	Width	Unique	Null	Default	Primary Key / Foreign Key	Referred Table	Description
1	id	int	11	Yes	No	None	Yes	-	AUTO_INCREMENT
2	date_ordered	datetime	6	No	No	None	-	-	-
3	complete	tinyint	1	Yes	Yes	NULL	-	-	-
4	transaction_id	varchar	255	Yes	Yes	NULL	-	-	-
5	customer_id	int	11	Yes	Yes	NULL	Yes	auth_user	-

orderitem									
Sr no.	Field Name	Data Type	Width	Unique	Null	Default	Primary Key / Foreign Key	Referred Table	Description
1	id	int	11	Yes	No	None	Yes	-	AUTO_INCREMENT
2	quantity	int	11	No	Yes	NULL	-	-	-
3	date_added	datetime	6	No	No	None	-	-	-
4	order_id	int	11	No	Yes	NULL	Yes	order	-
5	product_id	int	11	No	Yes	NULL	Yes	best_deals	-

shippingaddress									
Sr no.	Field Name	Data Type	Width	Unique	Null	Default	Primary Key / Foreign Key	Referred Table	Description
1	id	int	11	Yes	No	None	Yes	-	AUTO_INCREMENT
2	address	varchar	1000	No	No	None	-	-	-
3	city	varchar	255	No	No	None	-	-	-
4	country	varchar	255	No	No	None	-	-	-
5	zip_code	Varchar	255	No	No	None	-	-	-
6	date_added	datetime	6	No	No	None	-	-	-
7	customer_id	int	11	No	Yes	NULL	Yes	auth_user	-
8	order_id	int	11	No	Yes	NULL	Yes	order	-

5. Implementation Details

The system consists of 3 basic modules namely

1. User Module
2. Product Module
3. Order Module

Each module consists of several methods to implement the required functionality. Implementation is done using Django. Database used in these modules is MySQL.

5.1 User Module

This module is the base for authentication and authorization to ensure the security aspect of the user. It also includes profile creation and update it if required.

5.2 Product Module

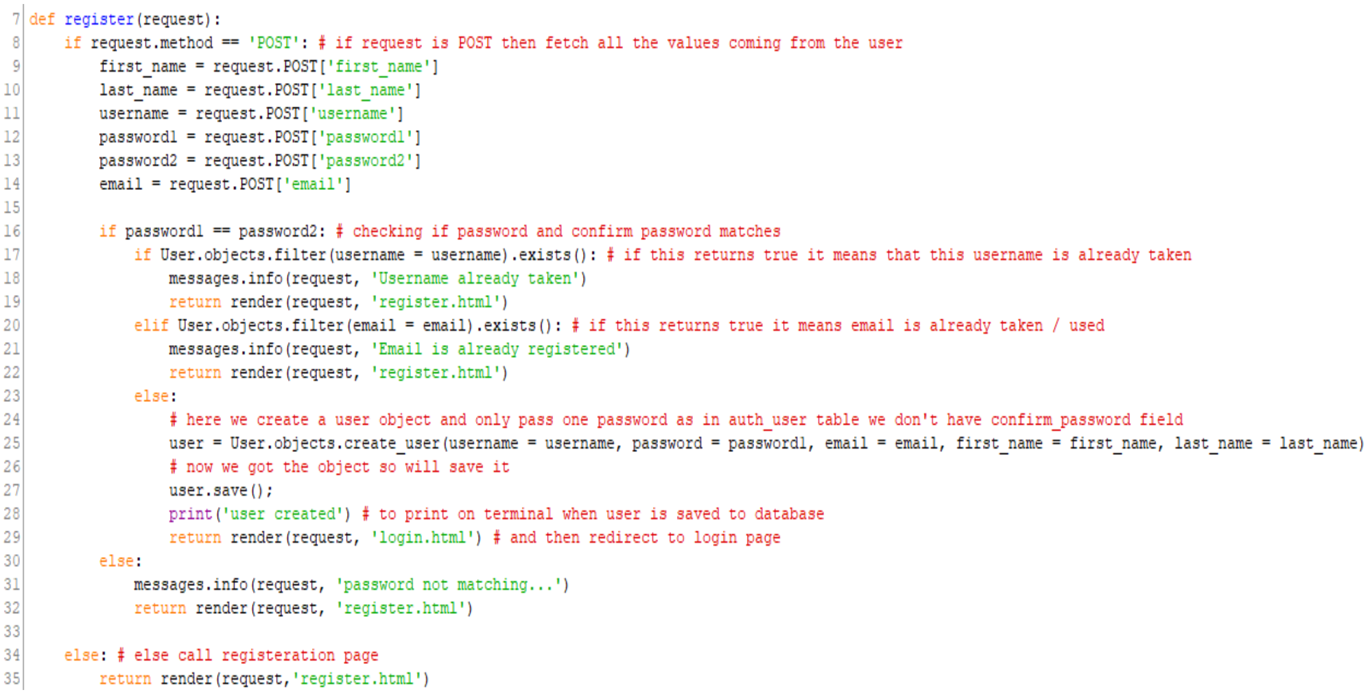
This module handles the functions related to product management. It allows user to search for a given product and also allows user to search for any type of checkups. It also displays product description for which user has searched. It also allows Admin / Seller to add product which are out of stock / unavailable. It allows user to provide feedback about the product.

5.3 Order Module

This module handles various functionality related to order management. It allows user to buy the product either through buy now option or by adding it to cart. It also allows user to track their order.

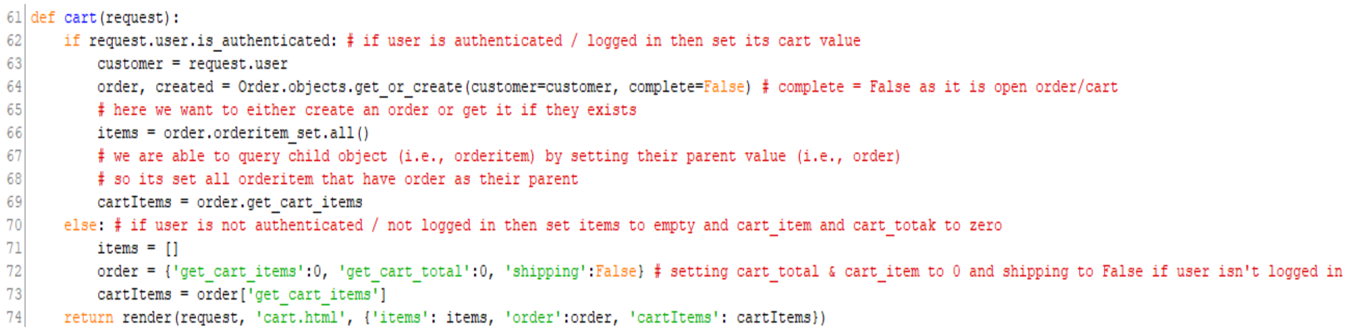
It allows user to cancel their order and can also return their order.

5.4 Function prototypes



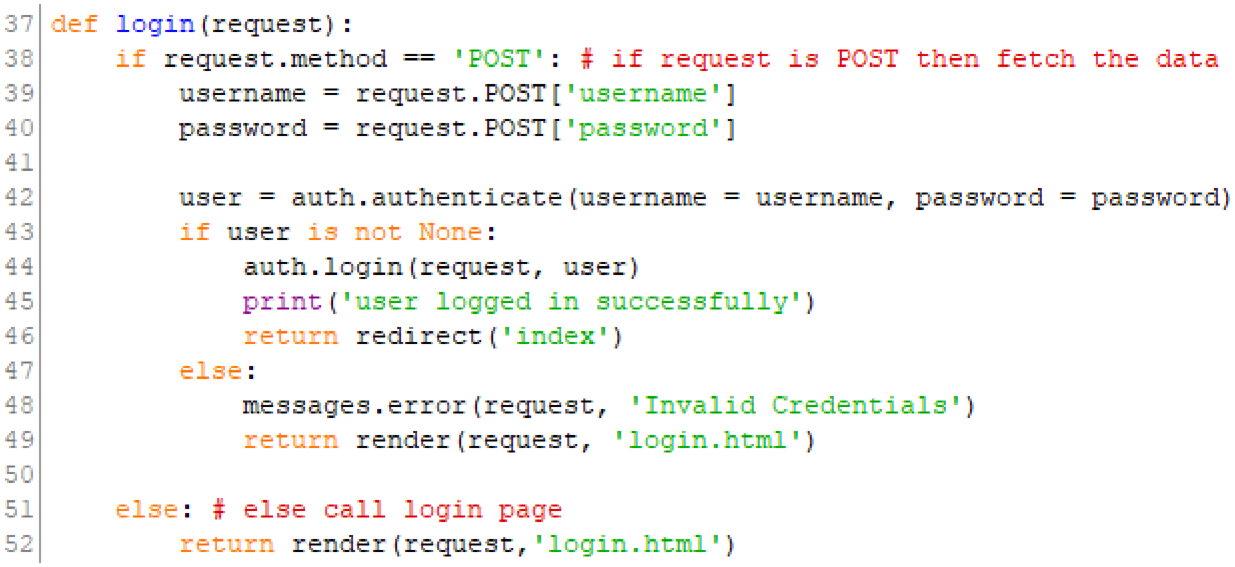
```python
def register(request):
    if request.method == 'POST': # if request is POST then fetch all the values coming from the user
        first_name = request.POST['first_name']
        last_name = request.POST['last_name']
        username = request.POST['username']
        password1 = request.POST['password1']
        password2 = request.POST['password2']
        email = request.POST['email']

        if password1 == password2: # checking if password and confirm password matches
            if User.objects.filter(username = username).exists(): # if this returns true it means that this username is already taken
                messages.info(request, 'Username already taken')
                return render(request, 'register.html')
            elif User.objects.filter(email = email).exists(): # if this returns true it means email is already taken / used
                messages.info(request, 'Email is already registered')
                return render(request, 'register.html')
            else:
                # here we create a user object and only pass one password as in auth_user table we don't have confirm_password field
                user = User.objects.create_user(username = username, password = password1, email = email, first_name = first_name, last_name = last_name)
                # now we got the object so will save it
                user.save();
                print('user created') # to print on terminal when user is saved to database
                return render(request, 'login.html') # and then redirect to login page
        else:
            messages.info(request, 'password not matching...')
            return render(request, 'register.html')

    else: # else call registeration page
        return render(request,'register.html')
```

Registration



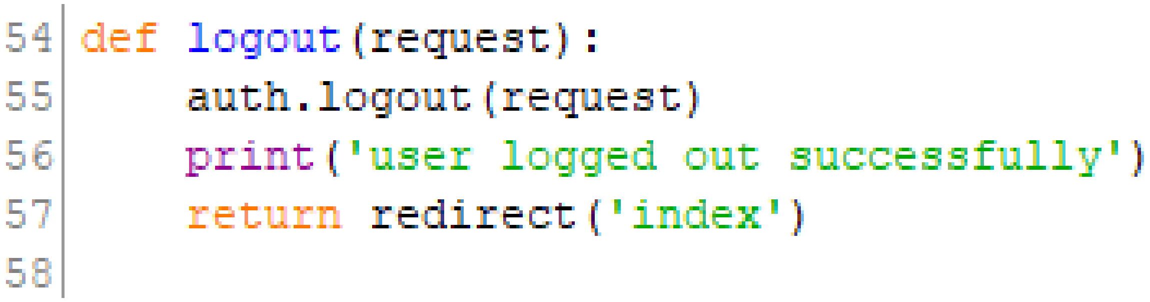
```python
def cart(request):
    if request.user.is_authenticated: # if user is authenticated / logged in then set its cart value
        customer = request.user
        order, created = Order.objects.get_or_create(customer=customer, complete=False) # complete = False as it is open order/cart
        # here we want to either create an order or get it if they exists
        items = order.orderitem_set.all()
        # we are able to query child object (i.e., orderitem) by setting their parent value (i.e., order)
        # so its set all orderitem that have order as their parent
        cartItems = order.get_cart_items
    else: # if user is not authenticated / not logged in then set items to empty and cart_item and cart_totak to zero
        items = []
        order = {'get_cart_items':0, 'get_cart_total':0, 'shipping':False} # setting cart_total & cart_item to 0 and shipping to False if user isn't logged in
        cartItems = order['get_cart_items']
    return render(request, 'cart.html', {'items': items, 'order':order, 'cartItems': cartItems})
```

Cart



```python
def login(request):
    if request.method == 'POST': # if request is POST then fetch the data
        username = request.POST['username']
        password = request.POST['password']

        user = auth.authenticate(username = username, password = password)
        if user is not None:
            auth.login(request, user)
            print('user logged in successfully')
            return redirect('index')
        else:
            messages.error(request, 'Invalid Credentials')
            return render(request, 'login.html')

    else: # else call login page
        return render(request,'login.html')
```

Login



```python
def logout(request):
    auth.logout(request)
    print('user logged out successfully')
    return redirect('index')

```

Logout

6. Testing

Manual testing was performed in order to find and fix the bugs in development process.

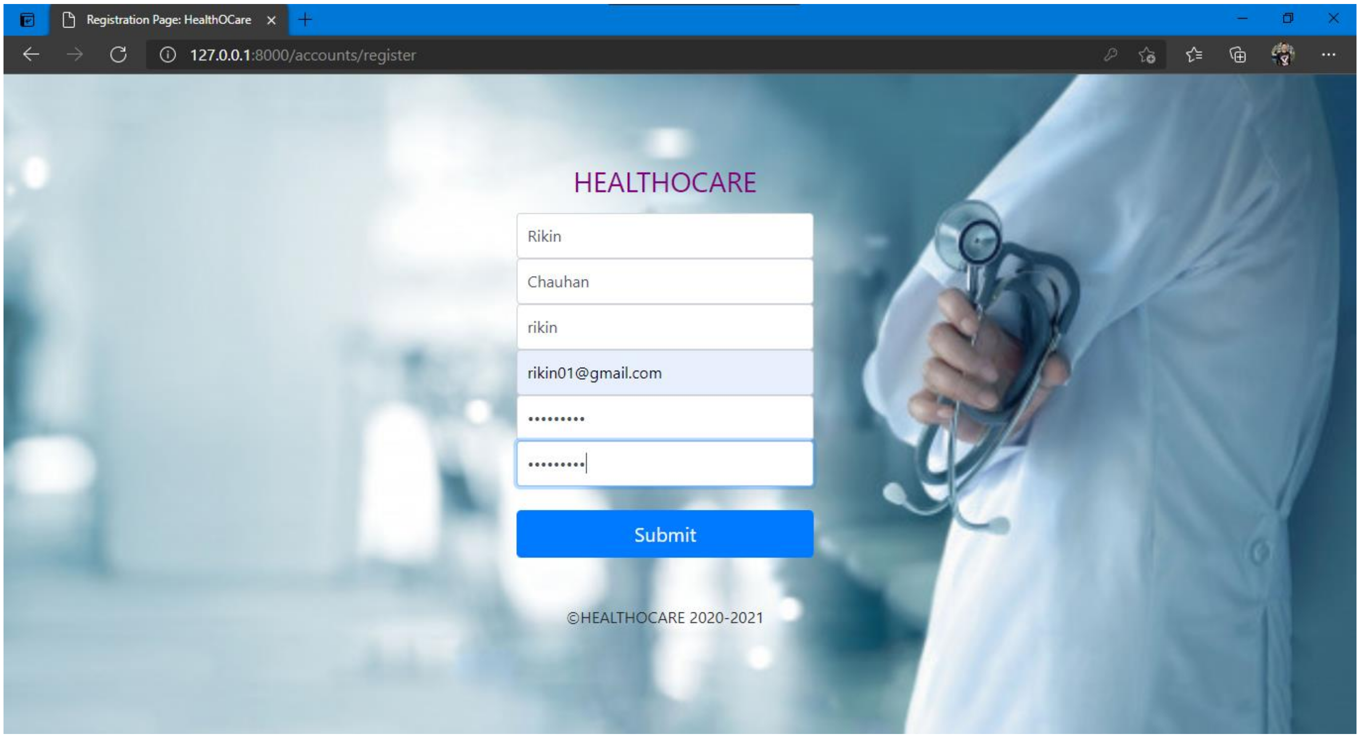
Testing Method: Manual Testing

Sr. No.	Test Scenario	Expected Result	Actual Result	Status
1	Login with incorrect credentials	User should not be able to log-in and shown an error message	The system shows the message ‘Invalid Credentials’ and redirected to login page	Success
2	Login with correct credentials	User should be able to log-in and redirected to home page	User is logged in and redirected to home page	Success
3	Registering with entering different password and confirm password	User should not be able to register and shown an error message	The system shows the message ‘password not matching...’ and redirected to registration page	Success
4	Registering with an existing username	User should not be able to register and shown an error message	The system shows the message ‘Username already taken’ and redirected to registration page	Success

5	Registering with an existing email-id	User should not be able to register and shown an error message	The system shows the message ‘Email is already registered’ and redirected to registration page	Success
6	Logout	User should be logged out and restricted from the system until next login	User is successfully logged out of the system	Success
7	Add to Cart	The product selected must be added to cart / The product against which up arrow is clicked must be added to cart and the total cart cost must be adjusted accordingly	Product is added to cart and cart cost is adjusted	Success
8	Remove from cart	The product against which down arrow is clicked must be removed from cart and the total cart cost must be adjusted accordingly	Product is removed from cart and cart cost is adjusted	Success

9	Add products	Admin / Seller must be able to add products into the system through admin page	Product is added and displayed on the home page	Success
10	Cart	The cart page is displayed	Cart page for a user is displayed	Success
11	Checkout	The checkout page is displayed along with cart details and shipping address form	Checkout page is displayed for a user	Success
12	Make payment	Payment is done and system shows an alert message	The system shows the message ‘Transaction completed’ and redirected to home page	Success

7. Screenshots


Registration Page: HealthOCare
127.0.0.1:8000/accounts/register
HEALTHOCARE

Rikin

Chauhan

rikin

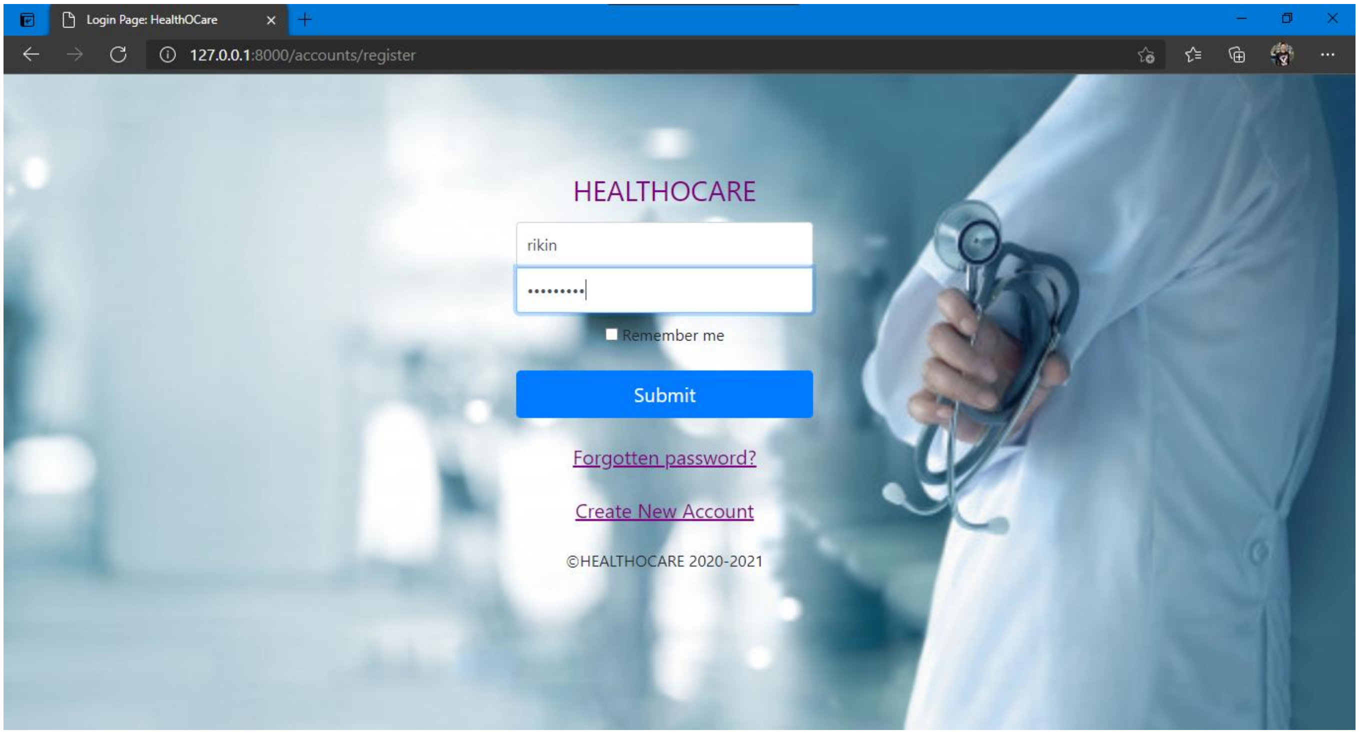
rikin01@gmail.com

•••••••••

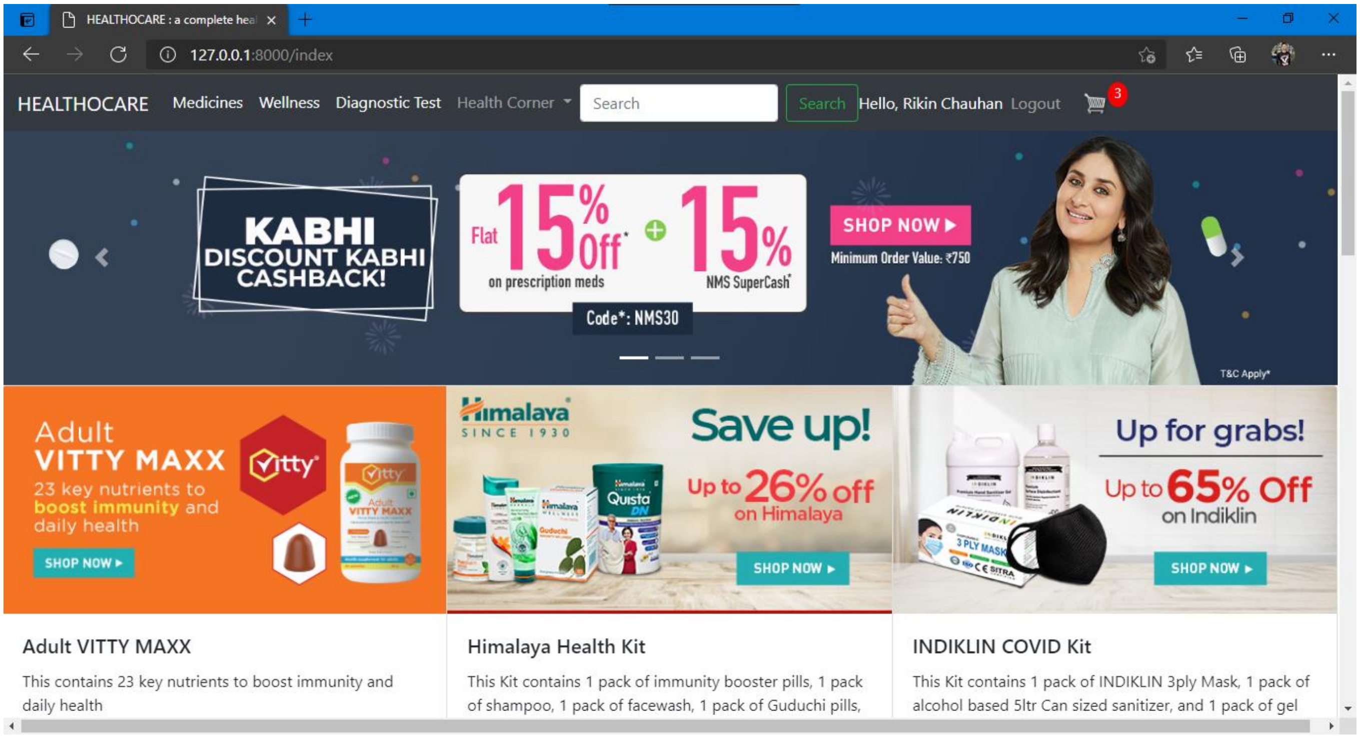
•••••••••

Submit
©HEALTHOCARE 2020-2021

Registration Page



Login page



Adult VITTY MAXX

This contains 23 key nutrients to boost immunity and daily health

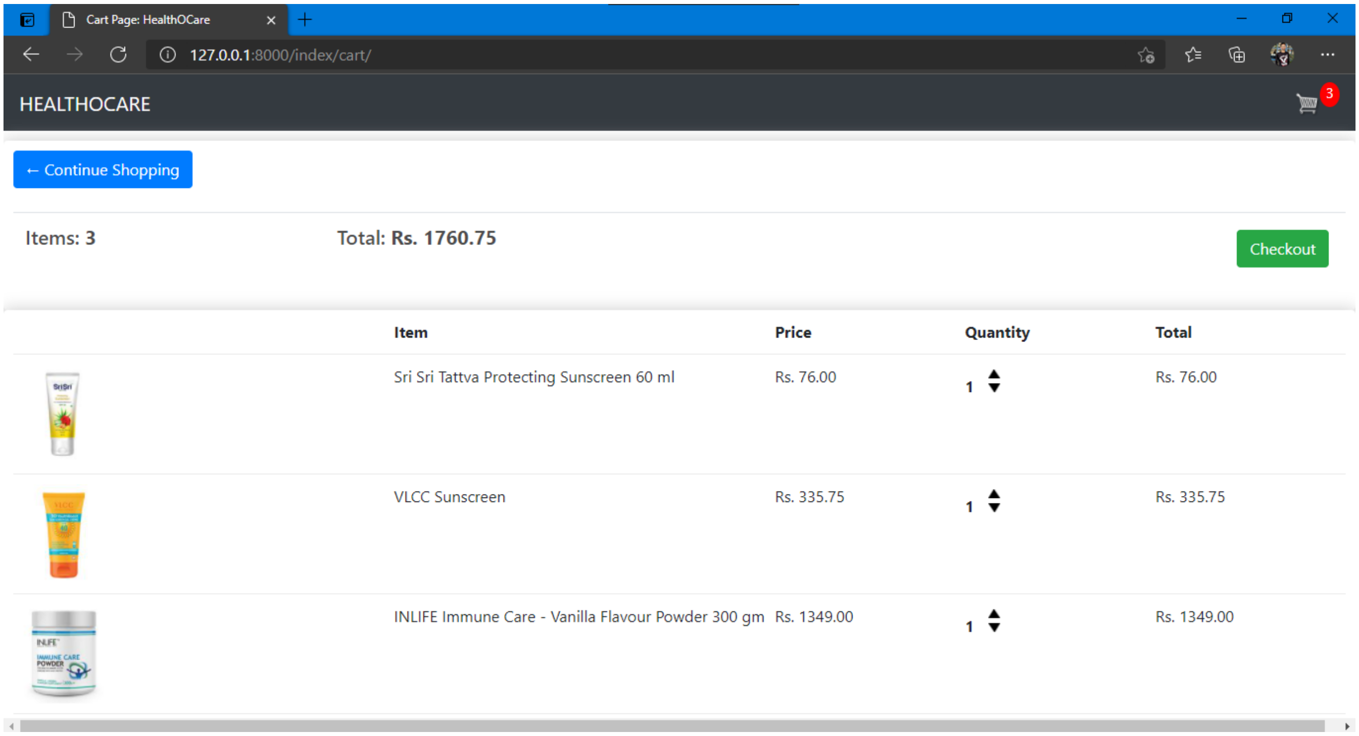
Himalaya Health Kit

This Kit contains 1 pack of immunity booster pills, 1 pack of shampoo, 1 pack of facewash, 1 pack of Guduchi pills,

INDIKLIN COVID Kit

This Kit contains 1 pack of INDIKLIN 3ply Mask, 1 pack of alcohol based 5ltr Can sized sanitizer, and 1 pack of gel

Home page

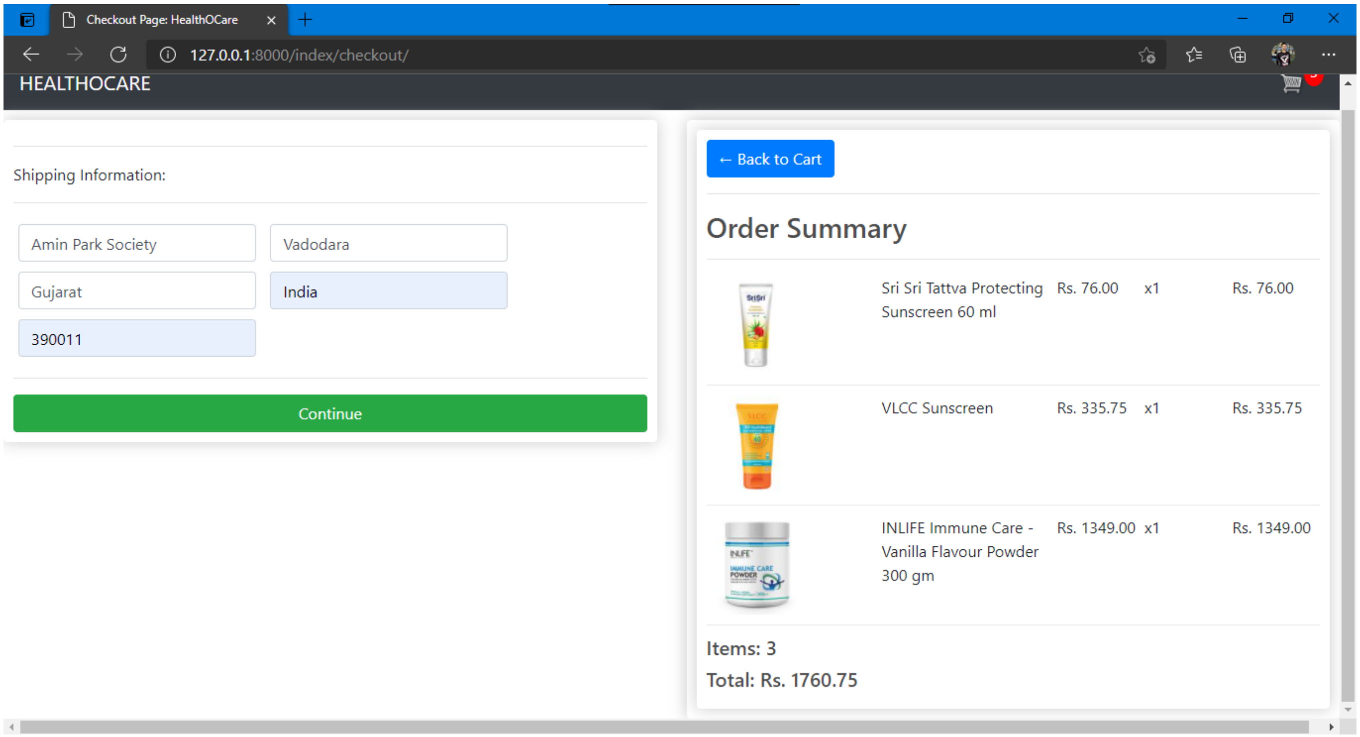


Items: 3

Total: Rs. 1760.75

	Item	Price	Quantity	Total
	Sri Sri Tattva Protecting Sunscreen 60 ml	Rs. 76.00	1	Rs. 76.00
	VLCC Sunscreen	Rs. 335.75	1	Rs. 335.75
	INLIFE Immune Care - Vanilla Flavour Powder 300 gm	Rs. 1349.00	1	Rs. 1349.00

Cart page



Shipping Information:

Amin Park Society

Vadodara

Gujarat

India

390011

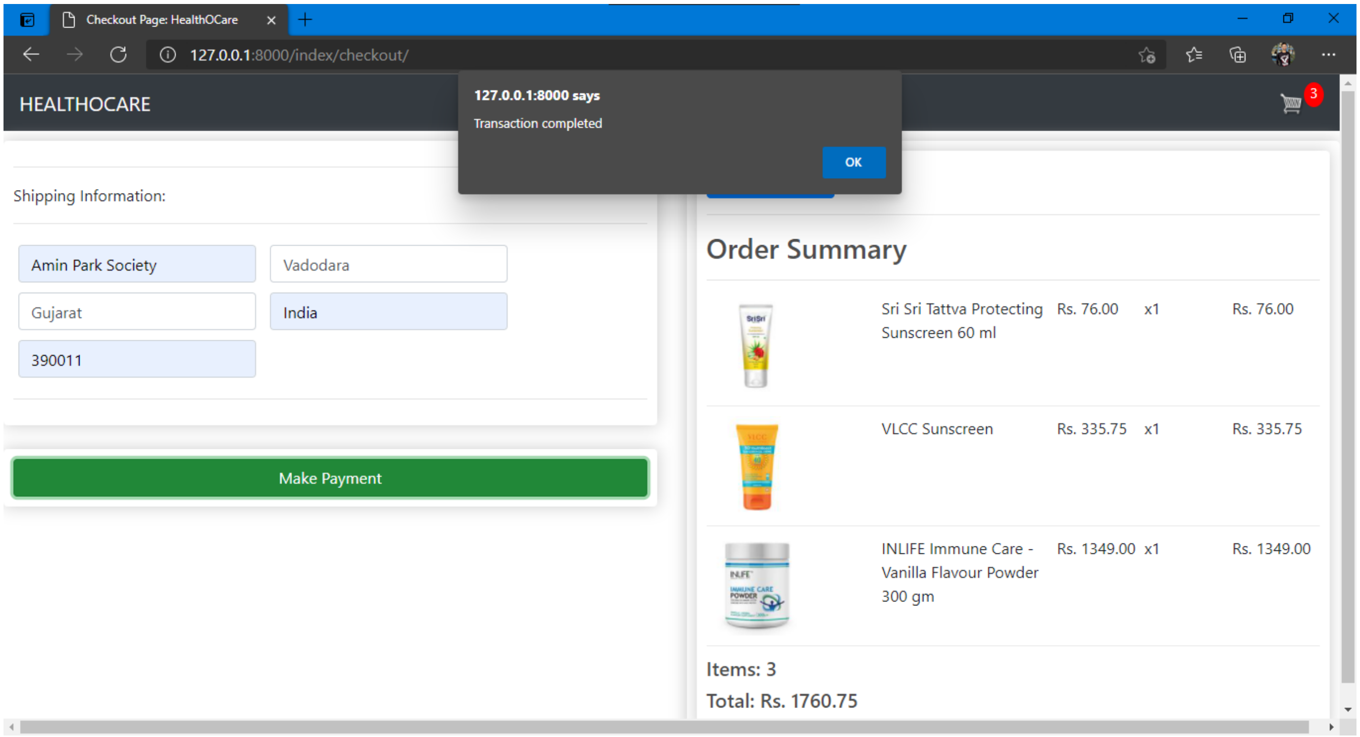
Order Summary

	Sri Sri Tattva Protecting Sunscreen 60 ml	Rs. 76.00	x1	Rs. 76.00
	VLCC Sunscreen	Rs. 335.75	x1	Rs. 335.75
	INLIFE Immune Care - Vanilla Flavour Powder 300 gm	Rs. 1349.00	x1	Rs. 1349.00

Items: 3

Total: Rs. 1760.75

Checkout page



127.0.0.1:8000 says
Transaction completed
OK

Shipping Information:

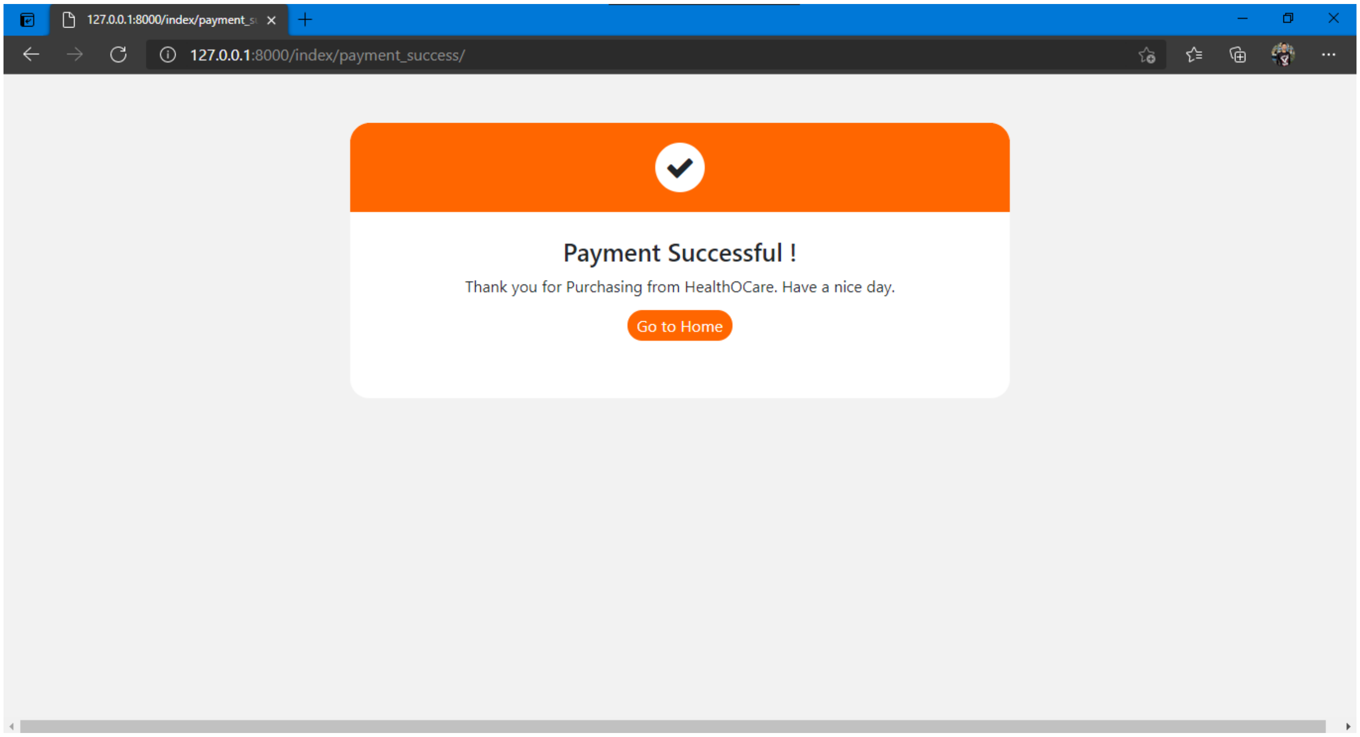
Amin Park Society | Vadodara
Gujarat | India
390011

Make Payment

Order Summary

Sri Sri Tattva Protecting Sunscreen 60 ml	Rs. 76.00	x1	Rs. 76.00
VLCC Sunscreen	Rs. 335.75	x1	Rs. 335.75
INLIFE Immune Care - Vanilla Flavour Powder 300 gm	Rs. 1349.00	x1	Rs. 1349.00

Items: 3
Total: Rs. 1760.75



127.0.0.1:8000/index/payment_s
127.0.0.1:8000/index/payment_success/

Payment Successful !

Thank you for Purchasing from HealthOCare. Have a nice day.

Go to Home

For Payment

8. Conclusion

The functionalities are implemented in system after understanding all the system modules according to the requirements. Functionalities that are successfully implemented in the system are:

- Signup / Registration
- Login
- User authentication
- User validation
- Logout
- Forgotten Password (Reset Password)
- Add Products (by Admin / Seller)
- Remove Products (by Admin / Seller)
- Manage User (by Admin)
- Track Customers Order (by Admin)
- Add to cart (by user)
- Remove from cart (by user)
- Make payment

After the implementation and coding of system, comprehensive testing was performed on the system to determine the errors and possible flaws in the system.

9. Limitations and Future Enhancements

We are able to implement some of the functionality of all modules. We aim to complete all the functionality of all modules and make this product ready to be used practically in all scenarios. Currently, the project runs completely fine if all the inputs / selections are given within proper criteria but it doesn’t cover all the corner cases.

The project can be extended to run robust. The project supports online in-browser view.

We can also extend it to run of mobile devices. Further extensions involve integrating it Machine Learning recommendation models for suggesting what medicine are you looking for. We can also add a Map Feature which tells the user the nearest located hospital for a given disease treatment.

10. Reference / Bibliography

Following links and websites were referred during the development of this project:

getbootstrap.com

stackoverflow.com

docs.djangoproject.com