

# ARHA-CONNECTING ARTISANS

## AN E-COMMERCE PLATFORM

### PHASE- 4: DEVELOPMENT PART 2

In this phase we are continuing to build our E-Commerce platform ARHA. In the previous development phase we have done few of the frontend features logo of the app, login page, register page, Home page, product page.

Now let us see about individual product page, shopping cart, orders list ,order summary, managing your account. We have also shown the flask program and the database which are the backend of our app.

#### INDIVIDUAL PRODUCT PAGE:

The individual product page of our app has a short description about the product, product code, the seller details, price, and the quantity the user needs. By default it is 1.The user can add the product to the cart by clicking the add to cart button in the individual products page itself.

The individual page looks something like this:

## Rope Khatli Bajot Stool - Black



Traditional Handicrafts Jute Rope Khatli Bajot Stool - Black

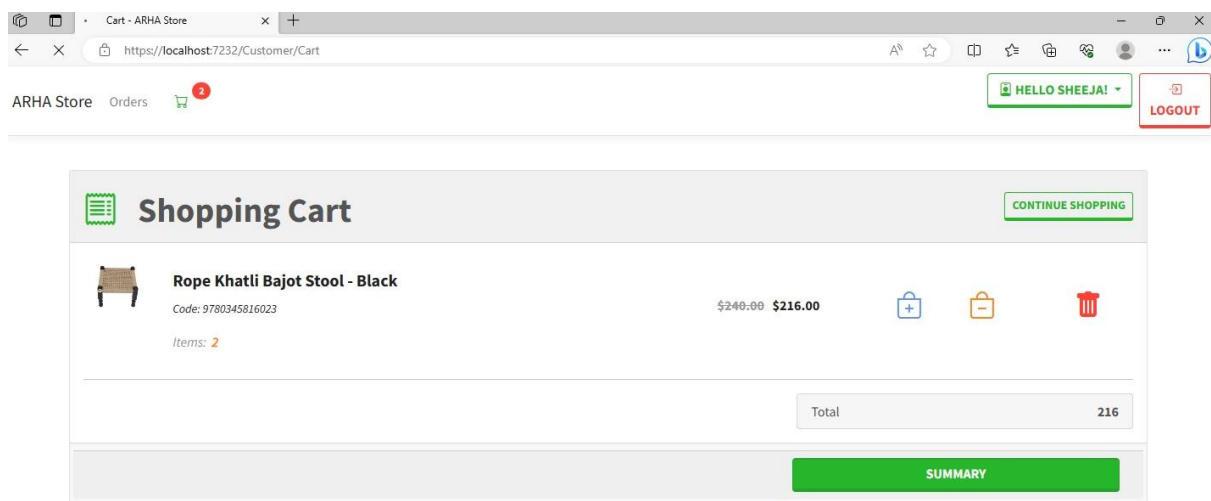
**Short Description:** Rope Khatli Bajot Stool - Black**Product Code :** 9780345816023**Category:** Clothes**Seller:** Jordan B. Peterson**Price:** \$120.00**As low as:** \$108.00**Quantity** [BACK TO LIST](#)[ADD TO CART](#)

The html CSS coding for the individual product page is:

```
1  @{{
2      ViewData["Title"] = "Index";
3      #pragma warning disable
4  }}
5  <div class="container p-3">
6      <div class="row pt-4">
7          <div class="col-6">
8              <h2 class="text-primary">Products</h2>
9          </div>
10         <div class="col-6 text-end">
11             <a class="btn btn-primary" asp-controller="Product" asp-action="Upsert">
12                 <i class="bi bi-plus-circle"></i> &nbsp; New
13             </a>
14         </div>
15     </div>
16     <br>
17     <table id="products" class="table table-striped table-hover stripe w-100">
18         <thead>
19             <tr>
20                 <th scope="col">
21                     @Html.DisplayName("Title")
22                 </th>
23                 <th scope="col">
24                     @Html.DisplayName("Code")
25                 </th>
26                 <th scope="col">
27                     @Html.DisplayName("Seller")
28                 </th>
29                 <th scope="col">
30                     @Html.DisplayName("Category")
31                 </th>
32                 <th scope="col">
33                     @Html.DisplayName("Price")
34                 </th>
35                 <th scope="col"></th>
36             </tr>
37         </thead>
38     </table>
39 </div>
40
41 @section scripts
```

## SHOPPING CART :

The shopping cart enables the user to see all the product that he/she has added to the cart. The shopping cart has the name, price, quantity, total bill amount. The user also has an option to remove the item from the cart .



The html CSS coding that we used for this is :

```
File Edit Selection View Go Run Terminal Help
my_orders.html my_cart.html _init.py 9+
C:\Users> shani> Downloads > OnlineShop-master > OnlineShop-master > OnlineShop > templates > my_cart.html > html
1 <html>
2 <head>
3 <link rel="stylesheet" href="{{url_for('static', filename='main.css')}}">
4 <link href="https://fonts.googleapis.com/css?family=Acme|K2D" rel="stylesheet">
5 <link rel="icon" href="{{url_for('static', filename='logo.png')}}">
6 </head>
7 <body>
8 <div class="sticky">
9 <h1>cart-ARHA Store</h1>
10 </div>
11 <div class="nav">
12 <p><a href="/">Home</a> > <a href="/buy">Buy</a> > <a href="">My Cart</a></p>
13 </div>
14 <h2>My Cart</h2>
15 {% if cart|length==0 %}
16 <p>Your Cart is empty right now. <a href="/buy">Click here to Shop for products</a></p>
17 {% else %}
18 <form action="" method="POST">
19 <table>
20 <tr>
21 <th>SHO.</th><th>PRODUCT</th><th>PRICE (PER UNIT)</th><th>QUANTITY</th><th></th>
22 </tr>
23 {% set cnt=[ 'val':1] %}
24 {% for item in cart %}
25 <tr>
26 <td>{{cnt['val']}}</td><td><a href="/viewproduct/{{item[0]}}">{{item[1]}}</a></td><td>{{item[2]}}</td>
27 <td><input name="qty{{item[0]}}" type="number" value="{{item[3]}}" min="1" max="{{item[4]}}" required</td>
28 <td><input class="press" type="button" value="REMOVE FROM CART"{{item[0]}}></td>
29 </tr>
30 <tr>
31 <td colspan="5">{{item[3]}}</td>
32 </tr>
33 </table>
34 <p class="press"><a href="/buy/cart/delete">EMPTY CART</a></p>
35 <input type="submit" value="PROCEED TO ORDER" style="display:block; margin-left:auto; margin-right:auto; margin-top:1vw;">
36 </form>
37 </body>
```

## ORDERS LIST:

The user can also view all the previous or present orders he/she has made. The page looks something like this.

ARHA Store Orders

HELLO SHEEJA! LOGOUT

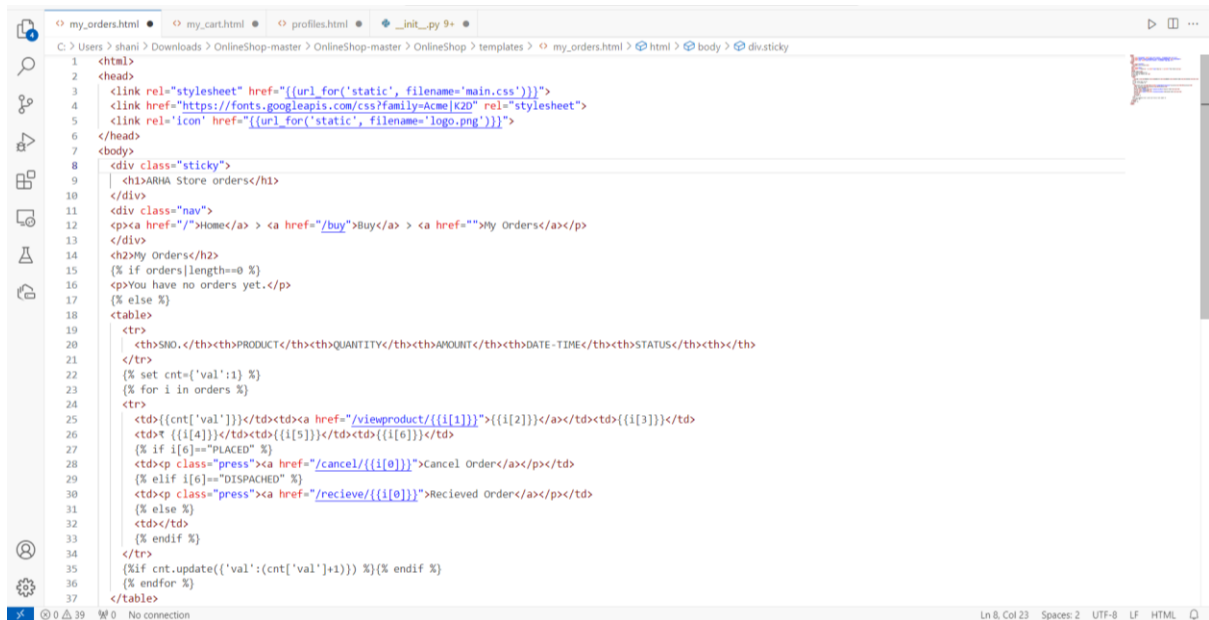
### Orders

Show 10 entries Search:

ID	First Name	Last Name	Phone Number	Email	Status	Total	
3	Sheeja	J S	9600171802	sheeja@gmail.com	Pending	216	
4	Sheeja	J S	9600171802	sheeja@gmail.com	Pending	216	
5	Sheeja	J S	9600171802	sheeja@gmail.com	Pending	216	
6	Sheeja	J S	9600171802	sheeja@gmail.com	Pending	216	

Showing 1 to 4 of 4 entries Previous 1 Next

The html CSS code has been shown:



```
1 <html>
2 <head>
3   <link rel="stylesheet" href="{url_for('static', filename='main.css')}">
4   <link href="https://fonts.googleapis.com/css?family=Acme|K2D" rel="stylesheet">
5   <link rel="icon" href="{url_for('static', filename='logo.png')}">
6 </head>
7 <body>
8   <div class="sticky">
9     <h1>ARWA Store orders</h1>
10  </div>
11  <div class="nav">
12    <p><a href="/">Home</a> > <a href="/buy">Buy</a> > <a href="">My Orders</a></p>
13  </div>
14  <h2>My Orders</h2>
15  {% if orders|length==0 %}
16  <p>You have no orders yet.</p>
17  {% else %}
18  <table>
19    <tr>
20      <th>SNO.</th><th>PRODUCT</th><th>QUANTITY</th><th>AMOUNT</th><th>DATE-TIME</th><th>STATUS</th><th></th>
21    </tr>
22    {% set cnt=['val':1] %}
23    {% for i in orders %}
24    <tr>
25      <td>{{cnt['val']}}</td><td><a href="/viewproduct/{{i[1]}}">{{i[2]}}</a></td><td>{{i[3]}}</td>
26      <td>{{i[4]}}</td><td>{{i[5]}}</td><td>{{i[6]}}</td>
27      {% if i[6]=="PLACED" %}
28      <td><button class="press"><a href="/cancel/{{i[0]}}">Cancel Order</a></td>
29      {% elif i[6]=="DISPATCHED" %}
30      <td><button class="press"><a href="/recieve/{{i[0]}}">Recieved Order</a></td>
31      {% else %}
32      <td></td>
33      {% endif %}
34    </tr>
35    {% if cnt.update({'val':(cnt['val']+1)}) %}
36    {% endif %}
37  </table>
```

## ORDER SUMMARY:

Before the user wishes to place an order, he/she will be able to view the order summary. The order summary has the details about the product, shipping details, estimated arrival date and place order button. Once the user fills up all the shipping details, he/she will be able to place the order.

The order summary page looks something like this:

ARHA Store
Orders

HELLO SHEEJA!
LOGOUT

## Order Summary

[BACK TO CART](#)

### Shipping Details

Name

Sheeja

JS

Phone Number

9600171802

Street Address

Keezhkulam

City

Marthandam

State

Tamil Nadu

Postal Code

629172

Estimate Arrival Date: 10/22/2023

PLACE ORDER

### Order Summary

Rope Khatli Bajot Stool - Black	\$216.00
Price:\$240.00	
Quantity: 2	
<b>Total</b>	<b>\$216.00</b>

The html CSS code for this is:

```

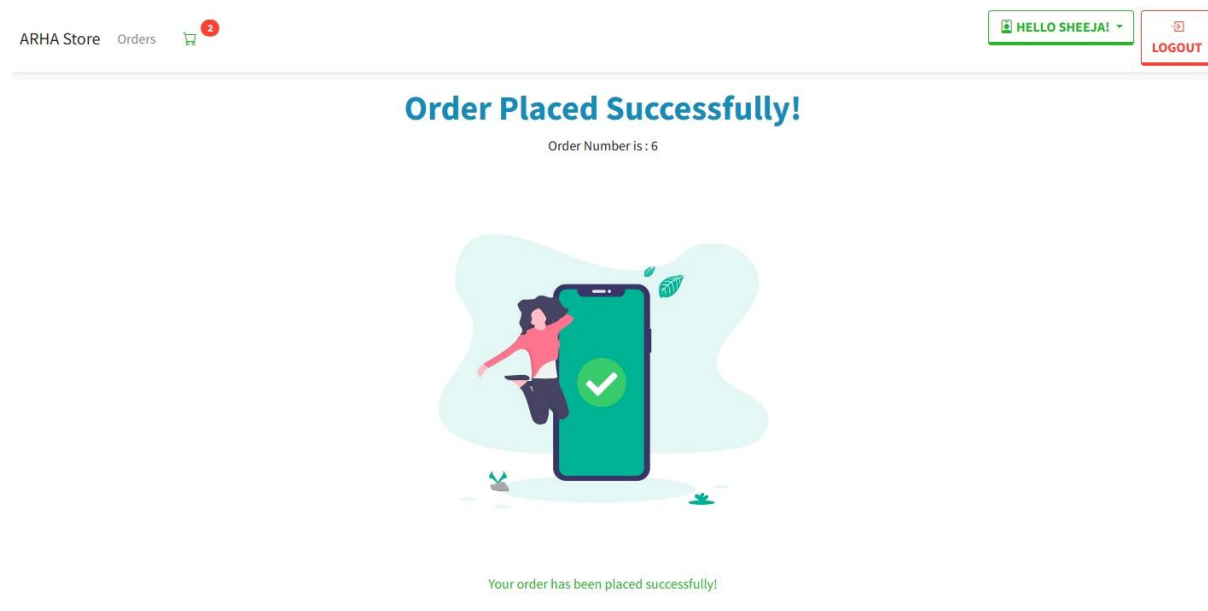
1  @using Books.Domain.Utilities
2  @model Books.Domain.ViewModels.OrderViewModel
3
4  @{
5      ViewData["Title"] = "Order";
6      #pragma warning disable
7  }
8
9  <br />
10 <div class="container-fluid">
11     <form method="post" enctype="multipart/form-data">
12         <div asp-validation-summary="All" class="text-danger"></div>
13         <input asp-for="OrderHeader.Id" hidden />
14         <div class="card">
15             <div class="card-header bg-secondary text-dark fw-bold fs-1 ml-0">
16                 <div class="row">
17                     <div class="col-12 d-none d-md-block col-md-6 pb-1">
18                         <i class="fa fa-shopping-cart"></i><i class="bi bi-tags-fill text-success"></i> &nbsp; Order Summary
19                     </div>
20                     <div class="col-6 text-end">
21                         <a asp-action="Index" class="btn btn-outline-info btn-sm">Back to Orders</a>
22                     </div>
23                 </div>
24             </div>
25             <div class="card-body">
26                 <div class="row">
27                     <div class="col-sm-12 col-lg-6 pb-4">
28                         <div class="row">
29                             <h4 class="d-flex justify-content-between align-items-center mb-3">
30                                 <span class="text-dark fw-bold">Pick Up Details</span>
31                             </h4>
32                         </div>
33                         <div class="row my-1">
34                             <div class="col-3">
35                                 <label>@Html.DisplayName("Name")</label>
36                             </div>
37                             @if (User.IsInRole(Roles.RoleType.Admin.ToString()) || User.IsInRole(Roles.RoleType.SuperAdmin.ToString()))
38                             {
39                                 <div class="col-4">
40                                     <input asp-for="OrderHeader.FirstName" readonly class="form-control" />
41                                     <span asp-validation-for="OrderHeader.FirstName" class="text-danger"></span>
42                                 </div>
43                             }
44                         </div>
45                     </div>
46                 </div>
47             </div>
48         </div>
49     </form>
50 </div>

```

## ORDER PLACED SUCCESSFULLY:

Once the user has placed the order by fulfilling the requirements in the order summary, he/she will be prompted with an order placed successfully page.

The page looks like this:

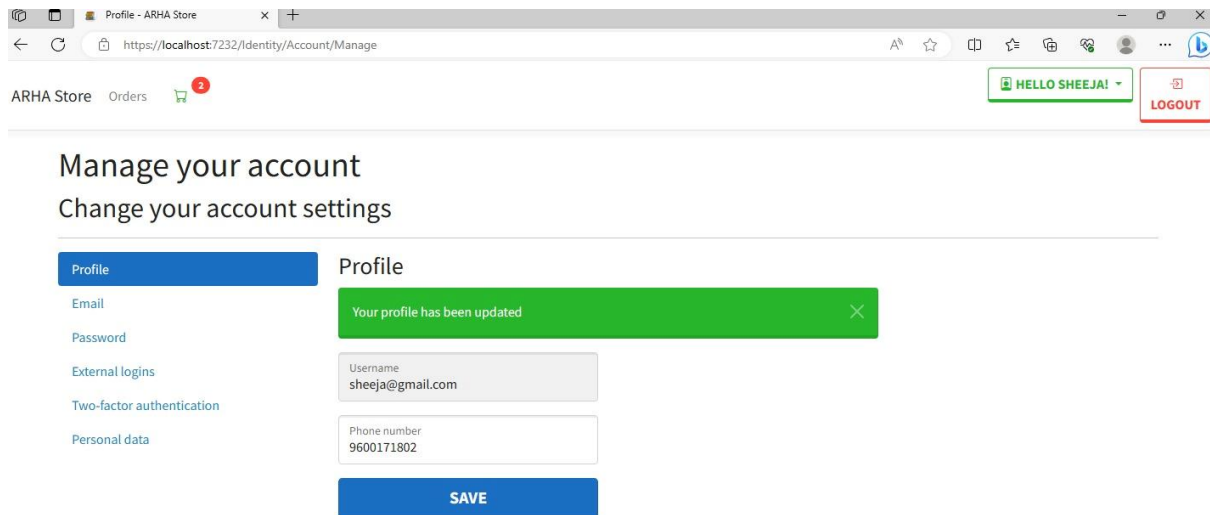


## MANAGE YOUR ACCOUNTS:

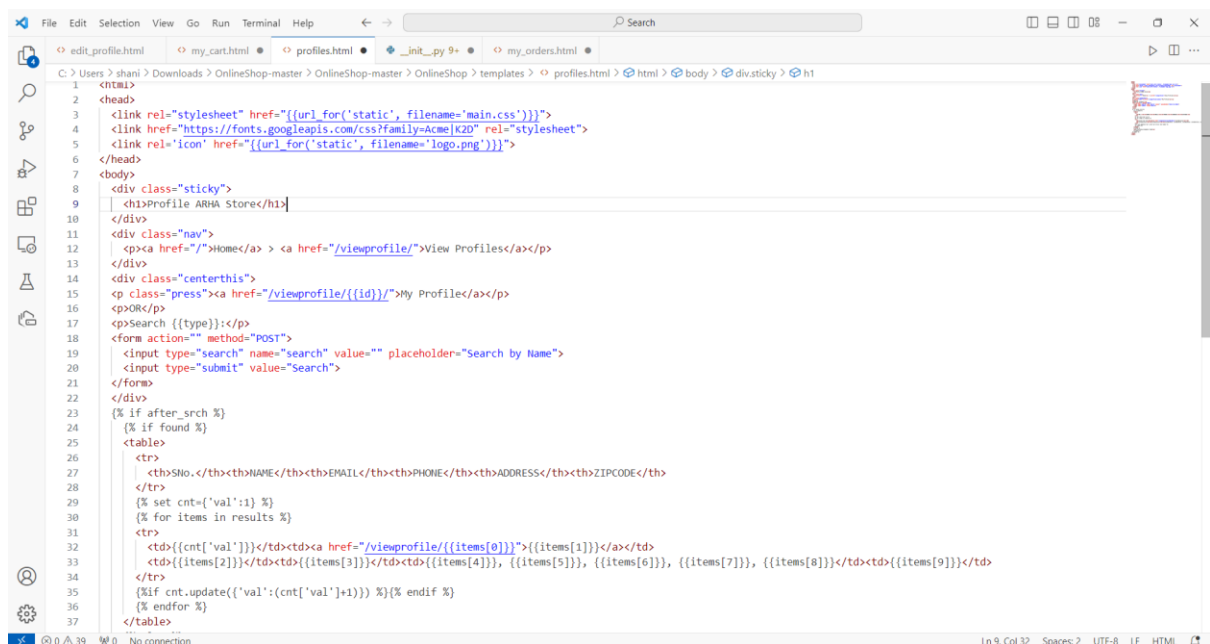
The user has an option of changing his/her account details after registration. This includes changing username, mobile number,

password in case of forgetting and other personal data.

The page looks something like this:



The HTML CSS code for this page is:





These are the front end features available in our app at present. Soon many more features will be added.

We hope that the pages will be visually pleasing and enhance user experience in our app.

Next, we are going to give an outline about the backend of our app.

## BACKEND:

To develop any app, we need backend. In our app ARHA we use Flask as our backend. The entire app runs in Flask Framework.

Flask is an API of Python that allows us to build up web-applications. **Flask** is often referred to as a microframework. It is designed to keep the core of the application simple and scalable.

**Flask** is a web framework that allows developers

to build lightweight web applications quickly and easily with **Flask** Libraries.

HTML CSS templates are used in this as a frontend.

SQLite is used for databases.

User authentication :

User authentication in our app is performed by using bcrypt library in flask.

This can be done by using the following coding which will be incorporated in our app soon.

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_bcrypt import Bcrypt
from flask_migrate import Migrate

from flask_login import (
    UserMixin,
    login_user,
```

```
LoginManager,  
current_user,  
logout_user,  
login_required,  
)
```

The flask program used to run our app is:

```
from flask import Flask, render_template, request, url_for,  
redirect, abort, session
```

```
from flask session import Session
```

```
from ARHA.dbaccess import *
```

```
import os
```

```
app = Flask(__name__)
```

```
sess = Session()
```

```
@app.route("/")
```

```
def home():
```

```
    if "userid" in session:
```

```
        return render_template("home.html", signedin=True,  
id=session['userid'], name=session['name'],  
type=session['type'])
```

```
return render_template("home.html", signedin=False)
```

```
@app.route("/signup/", methods = ["POST", "GET"])
```

```
def signup():
```

```
    if request.method == "POST":
```

```
        data = request.form
```

```
        ok = add_user(data)
```

```
        if ok:
```

```
            return render_template("success_signup.html")
```

```
            return render_template("signup.html", ok=ok)
```

```
        return render_template("signup.html", ok=True)
```

```
@app.route("/login/", methods=["POST", "GET"])
```

```
def login():
```

```
    if request.method == "POST":
```

```
        data = request.form
```

```
        userdat = auth_user(data)
```

```
        if userdat:
```

```
            session["userid"] = userdat[0]
```

```
            session["name"] = userdat[1]
```

```
            session["type"] = data["type"]
```

```
            return redirect(url_for('home'))
```

```
        return render_template("login.html", err=True)
```

```
return render_template("login.html", err=False)
```

```
@app.route("/logout/")
```

```
def logout():
```

```
    session.pop('userid')
```

```
    session.pop('name')
```

```
    session.pop('type')
```

```
    return redirect(url_for('home'))
```

```
@app.route("/viewprofile/<id>/")
```

```
def view_profile(id):
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    userid = session["userid"]
```

```
    type = session["type"]
```

```
    my = True if userid==id else False
```

```
    if not my: profile_type = "Customer" if type=="Seller" else  
    "Seller"
```

```
    else: profile_type = type
```

```
    det, categories = fetch_details(id, profile_type) #details
```

```
    if len(det)==0:
```

```
        abort(404)
```

```

det = det[0]
return render_template("view_profile.html",
                        type=profile_type,
                        name=det[1],
                        email=det[2],
                        phone=det[3],
                        area=det[4],
                        locality=det[5],
                        city=det[6],
                        state=det[7],
                        country=det[8],
                        zip=det[9],
                        category=(None if
profile_type=="Customer" else categories),
                        my=my)

@app.route("/viewprofile/", methods=["POST", "GET"])
def profile():
    if 'userid' not in session:
        return redirect(url_for('home'))
    type = "Seller" if session['type']=="Customer" else
"Customer"
    if request.method=="POST":

```

```

        search = request.form['search']
        results = search_users(search, type)
        found = len(results)

        return render_template('profiles.html',
            id=session['userid'], type=type, after_srch=True,
            found=found, results=results)

    return render_template('profiles.html', id=session['userid'],
        type=type, after_srch=False)

@app.route("/viewprofile/<id>/sellerproducts/")
def seller_products(id):
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session["type"]=="Seller":
        abort(403)
    det, categories = fetch_details(id, "Seller") #details
    if len(det)==0:
        abort(404)
    det = det[0]
    name=det[1]
    res = get_seller_products(id)

    return render_template('seller_products.html', name=name,
        id=id, results=res)

```

```
@app.route("/editprofile/", methods=["POST", "GET"])
def edit_profile():
    if 'userid' not in session:
        return redirect(url_for('home'))

    if request.method=="POST":
        data = request.form
        update_details(data, session['userid'], session['type'])
        return redirect(url_for('view_profile',
id=session['userid']))

    if request.method=="GET":
        userid = session["userid"]
        type = session["type"]
        det, _ = fetch_details(userid, type)
        det = det[0]
        return render_template("edit_profile.html",
                                type=type,
                                name=det[1],
                                email=det[2],
                                phone=det[3],
                                area=det[4],
```



```
locality=det[5],  
city=det[6],  
state=det[7],  
country=det[8],  
zip=det[9])
```

```
@app.route("/changepassword/", methods=["POST", "GET"])
```

```
def change_password():
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    check = True
```

```
    equal = True
```

```
    if request.method=="POST":
```

```
        userid = session["userid"]
```

```
        type = session["type"]
```

```
        old_psswd = request.form["old_psswd"]
```

```
        new_psswd = request.form["new_psswd"]
```

```
        cnfrm_psswd = request.form["cnfrm_psswd"]
```

```
        check = check_psswd(old_psswd, userid, type)
```

```
        if check:
```

```
            equal = (new_psswd == cnfrm_psswd)
```

```
            if equal:
```

```
                set_psswd(new_psswd, userid, type)
```

```
        return redirect(url_for('home'))

    return render_template("change_password.html",
check=check, equal=equal)

@app.route("/sell/", methods=["POST", "GET"])
def my_products():
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session["type"]=="Customer":
        abort(403)
    categories = get_categories(session["userid"])
    if request.method=="POST":
        data = request.form
        srchBy = data["search method"]
        category = None if srchBy=='by keyword' else
data["category"]
        keyword = data["keyword"]
        results = search_myproduct(session['userid'], srchBy,
category, keyword)

        return render_template('my_products.html',
categories=categories, after_srch=True, results=results)

    return render_template("my_products.html",
categories=categories, after_srch=False)
```

```
@app.route("/sell/addproducts/", methods=["POST", "GET"])
def add_products():
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session["type"]=="Customer":
        abort(403)
    if request.method=="POST":
        data = request.form
        add_prod(session['userid'],data)
        return redirect(url_for('my_products'))
    return render_template("add_products.html")
```

```
@app.route("/viewproduct/")
def view_prod():
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session['type']=="Seller":
        return redirect(url_for('my_products'))
    if session['type']=="Customer":
        return redirect(url_for('buy'))
```

```
@app.route("/viewproduct/<id>/")
def view_product(id):
```

```
if 'userid' not in session:
    return redirect(url_for('home'))
type = session["type"]
ispresent, tup = get_product_info(id)
if not ispresent:
    abort(404)

(name, quantity, category, cost_price, sell_price, sellID,
 desp, sell_name) = tup
if type=="Seller" and sellID!=session['userid']:
    abort(403)

return render_template('view_product.html', type=type,
name=name, quantity=quantity, category=category,
cost_price=cost_price, sell_price=sell_price, sell_id=sellID,
sell_name=sell_name, desp=desp, prod_id=id)

@app.route("/viewproduct/<id>/edit/", methods=["POST",
"GET"])
def edit_product(id):
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session['type']=="Customer":
        abort(403)
    ispresent, tup = get_product_info(id)
    if not ispresent:
```

```

        abort(404)

    (name, quantity, category, cost_price, sell_price, sellID,
    desp, sell_name) = tup

    if sellID!=session['userid']:
        abort(403)

    if request.method=="POST":
        data = request.form
        update_product(data, id)
        return redirect(url_for('view_product', id=id))

    return render_template('edit_product.html', prodID=id,
    name=name, qty=quantity, category=category,
    price=cost_price, desp=desp)

@app.route("/buy/", methods=["POST", "GET"])
def buy():
    if 'userid' not in session:
        return redirect(url_for('home'))

    if session['type']=="Seller":
        abort(403)

    if request.method=="POST":
        data = request.form
        srchBy = data["search method"]
        category = None if srchBy=='by keyword' else
data["category"]

```

```
keyword = data["keyword"]
results = search_products(srchBy, category, keyword)
return render_template('search_products.html',
after_srch=True, results=results)

return render_template('search_products.html',
after_srch=False)
```

```
@app.route("/buy/<id>/", methods=['POST', 'GET'])
def buy_product(id):
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session['type']=="Seller":
        abort(403)
    ispresent, tup = get_product_info(id)
    if not ispresent:
        abort(404)
    (name, quantity, category, cost_price, sell_price, sellID,
desp, sell_name) = tup
    if request.method=="POST":
        data = request.form
        total = int(data['qty'])*float(sell_price)
        return redirect(url_for('buy_confirm', total=total,
quantity=data['qty'], id=id))
```

```
    return render_template('buy_product.html', name=name,
category=category, desp=desp, quantity=quantity,
price=sell_price)
```

```
@app.route("/buy/<id>/confirm/", methods=["POST",
"GET"])
```

```
def buy_confirm(id):
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    if session['type']=="Seller":
```

```
        abort(403)
```

```
    ispresent, tup = get_product_info(id)
```

```
    if not ispresent:
```

```
        abort(404)
```

```
    (name, quantity, category, cost_price, sell_price, sellID,
desp, sell_name) = tup
```

```
    if 'total' not in request.args or 'quantity' not in request.args:
```

```
        abort(404)
```

```
    total = request.args['total']
```

```
    qty = request.args['quantity']
```

```
    if request.method=="POST":
```

```
        choice = request.form['choice']
```

```
        if choice=="PLACE ORDER":
```

```
            place_order(id, session['userid'], qty)
```

```
        return redirect(url_for('my_orders'))
    elif choice=="CANCEL":
        return redirect(url_for('buy_product', id=id))
    items = ((name, qty, total),)
    return render_template('buy_confirm.html', items=items,
total=total)
```

```
@app.route("/buy/myorders/")
```

```
def my_orders():
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    if session['type']=="Seller":
```

```
        abort(403)
```

```
    res = cust_orders(session['userid'])
```

```
    return render_template('my_orders.html', orders=res)
```

```
@app.route("/cancel/<orderID>/")
```

```
def cancel_order(orderID):
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    res = get_order_details(orderID)
```

```
    if len(res)==0:
```

```
        abort(404)
```



```
    custID = res[0][0]
    sellID = res[0][1]
    status = res[0][2]
    if session['type']=="Seller" and sellID!=session['userid']:
        abort(403)
    if session['type']=="Customer" and
custID!=session['userid']:
        abort(403)
    if status!="PLACED":
        abort(404)
    change_order_status(orderID, "CANCELLED")
    return redirect(url_for('my_orders')) if
session['type']=="Customer" else
redirect(url_for('new_orders'))

@app.route("/dispatch/<orderID>/")
def dispatch_order(orderID):
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session['type']=="Customer":
        abort(403)
    res = get_order_details(orderID)
    if len(res)==0:
        abort(404)
```

```
custID = res[0][0]
sellID = res[0][1]
status = res[0][2]
if session['userid']!=sellID:
    abort(403)
if status!="PLACED":
    abort(404)
change_order_status(orderID, "DISPATCHED")
return redirect(url_for('new_orders'))
```

```
@app.route("/recieve/<orderID>/")
```

```
def recieve_order(orderID):
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session['type']=="Seller":
        abort(403)
    res = get_order_details(orderID)
    if len(res)==0:
        abort(404)
    custID = res[0][0]
    sellID = res[0][1]
    status = res[0][2]
    if session['userid']!=custID:
```

```
        abort(403)
    if status!="DISPATCHED":
        abort(404)
    change_order_status(orderID, "RECIEVED")
    return redirect(url_for('my_purchases'))
```

```
@app.route("/buy/purchases/")
def my_purchases():
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session['type']=="Seller":
        abort(403)
    res = cust_purchases(session['userid'])
    return render_template('my_purchases.html',
purchases=res)
```

```
@app.route("/sell/neworders/")
def new_orders():
    if 'userid' not in session:
        return redirect(url_for('home'))
    if session['type']=="Customer":
        abort(403)
    res = sell_orders(session['userid'])
```

```
return render_template('new_orders.html', orders=res)
```

```
@app.route("/sell/sales/")
```

```
def my_sales():
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    if session['type']=="Customer":
```

```
        abort(403)
```

```
    res = sell_sales(session['userid'])
```

```
    return render_template('my_sales.html', sales=res)
```

```
@app.route("/buy/cart/", methods=["POST", "GET"])
```

```
def my_cart():
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    if session['type']=="Seller":
```

```
        abort(403)
```

```
    cart = get_cart(session['userid'])
```

```
    if request.method=="POST":
```

```
        data = request.form
```

```
        qty = {}
```

```
        for i in data:
```

```
            if i.startswith("qty"):
```

```
        qty[i[3:]] = data[i]    #qty[prodID]=quantity
    update_cart(session['userid'], qty)
    return redirect("/buy/cart/confirm/")
return render_template('my_cart.html', cart=cart)
```

```
@app.route("/buy/cart/confirm/", methods=["POST", "GET"])
```

```
def cart_purchase_confirm():
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    if session['type'] == "Seller":
```

```
        abort(403)
```

```
    if request.method == "POST":
```

```
        choice = request.form['choice']
```

```
        if choice == "PLACE ORDER":
```

```
            cart_purchase(session['userid'])
```

```
            return redirect(url_for('my_orders'))
```

```
        elif choice == "CANCEL":
```

```
            return redirect(url_for('my_cart'))
```

```
    cart = get_cart(session['userid'])
```

```
    items = [(i[1], i[3], float(i[2])*float(i[3])) for i in cart]
```

```
    total = 0
```

```
    for i in cart:
```

```
        total += float(i[2])*int(i[3])
```

```
    return render_template('buy_confirm.html', items=items,
total=total)
```

```
@app.route("/buy/cart/<prodID>/")
```

```
def add_to_cart(prodID):
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    if session['type']=="Seller":
```

```
        abort(403)
```

```
    add_product_to_cart(prodID, session['userid'])
```

```
    return redirect(url_for('view_product', id=prodID))
```

```
@app.route("/buy/cart/delete/")
```

```
def delete_cart():
```

```
    if 'userid' not in session:
```

```
        return redirect(url_for('home'))
```

```
    if session['userid']=="Seller":
```

```
        abort(403)
```

```
    empty_cart(session['userid'])
```

```
    return redirect(url_for('my_cart'))
```

```
@app.route("/buy/cart/delete/<prodID>/")
```

```
def delete_prod_cart(prodID):
```

```
if 'userid' not in session:
    return redirect(url_for('home'))
if session['userid']=="Seller":
    abort(403)
remove_from_cart(session['userid'], prodID)
return redirect(url_for('my_cart'))
```

```
app.config['SECRET_KEY'] = os.urandom(17)
app.config['SESSION_TYPE'] = 'filesystem'
app.config['TEMPLATES_AUTO_RELOAD'] = True
sess.init_app(app)
```

```
if __name__=="__main__":
    app.run()
```

## DATABASE:

Now we have used SQLite for our convenience as beginners.

The coding we have done for this is:

```

1 import sqlite3
2
3 def gen_custID():
4     conn = sqlite3.connect("ARHA/ARHA.db")
5     cur = conn.cursor()
6     cur.execute("UPDATE metadata SET custnum = custnum + 1")
7     conn.commit()
8     custnum = str([i for i in cur.execute("SELECT custnum FROM metadata")][0][0])
9     conn.close()
10    id = "CID"+0*(7-len(custnum))+custnum
11    return id
12
13 def gen_sellID():
14     conn = sqlite3.connect("ARHA/ARHA.db")
15     cur = conn.cursor()
16     cur.execute("UPDATE metadata SET sellnum = sellnum + 1")
17     conn.commit()
18     sellnum = str([i for i in cur.execute("SELECT sellnum FROM metadata")][0][0])
19     conn.close()
20    id = "SID"+0*(7-len(sellnum))+sellnum
21    return id
22
23 def gen_prodID():
24     conn = sqlite3.connect("ARHA/ARHA.db")
25     cur = conn.cursor()
26     cur.execute("UPDATE metadata SET prodnum = prodnum + 1")
27     conn.commit()
28     prodnum = str([i for i in cur.execute("SELECT prodnum FROM metadata")][0][0])
29     conn.close()
30    id = "PID"+0*(7-len(prodnum))+prodnum
31    return id
32
33 def gen_orderID():
34     conn = sqlite3.connect("ARHA/ARHA.db")
35     cur = conn.cursor()
36     cur.execute("UPDATE metadata SET ordernum = ordernum + 1")
37     conn.commit()

```

Ln 39, Col 1 No connection Ln 402, Col 41 Spaces: 4 UTF-8 LF Python 3

```

38    ordernum = str([i for i in cur.execute("SELECT ordernum FROM metadata")][0][0])
39    conn.close()
40    id = "OID"+0*(7-len(ordernum))+ordernum
41    return id
42
43 def add_user(data):
44     conn = sqlite3.connect("ARHA/ARHA.db")
45     cur = conn.cursor()
46     email = data["email"]
47     if data["type"]=="customer":
48         a = cur.execute("SELECT * FROM customer WHERE email=?", (email,))
49     elif data["type"]=="seller":
50         a = cur.execute("SELECT * FROM seller WHERE email=?", (email,))
51     if len(list(a))!=0:
52         return False
53     tup = ( data["name"],
54            data["email"],
55            data["phone"],
56            data["area"],
57            data["locality"],
58            data["city"],
59            data["state"],
60            data["country"],
61            data["zip"],
62            data["password"])
63     if data["type"]=="customer":
64         cur.execute("INSERT INTO customer VALUES (?, ?, ?, ?, ?, ?, ?, ?)", (gen_custID(), *tup))
65     elif data["type"]=="seller":
66         cur.execute("INSERT INTO seller VALUES (?, ?, ?, ?, ?, ?, ?, ?)", (gen_sellID(), *tup))
67     conn.commit()
68     conn.close()
69     return True
70
71 def auth_user(data):
72     conn = sqlite3.connect("ARHA/ARHA.db")
73     cur = conn.cursor()
74     type = data["type"]

```

Ln 39, Col 1 No connection Ln 402, Col 41 Spaces: 4 UTF-8 LF Python 3.11.6 64-bit



```

74 type = data["type"]
75 email = data["email"]
76 password = data["password"]
77 if type=="customer":
78     a = cur.execute("SELECT custID, name FROM customer WHERE email=? AND password=?", (email, password))
79 elif type=="seller":
80     a = cur.execute("SELECT sellID, name FROM seller WHERE email=? AND password=?", (email, password))
81 a = list(a)
82 conn.close()
83 if len(a)==0:
84     return False
85 return a[0]
86
87 def fetch_details(userid, type):
88     conn = sqlite3.connect("ARHA/ARHA.db")
89     cur = conn.cursor()
90     if type=="customer":
91         a = cur.execute("SELECT * FROM customer WHERE custID=?", (userid,))
92         a = list(a)
93         b = []
94     elif type=="seller":
95         a = cur.execute("SELECT * FROM seller WHERE sellID=?", (userid,))
96         a = list(a)
97         b = cur.execute("SELECT DISTINCT(category) from product WHERE sellID=?", (userid,))
98         b = [i[0] for i in b]
99     conn.close()
100     return a, b
101
102 def search_users(search, srch_type):
103     conn = sqlite3.connect("ARHA/ARHA.db")
104     cur = conn.cursor()
105     search = "%"+search+"%"
106     if srch_type=="customer":
107         res = cur.execute("SELECT custID, name, email, phone, area, locality, city, state, country, zipcode FROM customer WHERE LOWER(name) like ?", (search,))
108     elif srch_type=="seller":
109         res = cur.execute("SELECT sellID, name, email, phone, area, locality, city, state, country, zipcode FROM seller WHERE LOWER(name) like ?", (search,))
110     res = [i for i in res]

```

```

108     res = cur.execute("SELECT custID, name, email, phone, area, locality, city, state, country, zipcode FROM customer WHERE LOWER(name) like ?", (search,))
109     elif srch_type=="seller":
110         res = cur.execute("SELECT sellID, name, email, phone, area, locality, city, state, country, zipcode FROM seller WHERE LOWER(name) like ?", (search,))
111     res = [i for i in res]
112     conn.close()
113     return res
114
115 def update_details(data, userid, type):
116     conn = sqlite3.connect("ARHA/ARHA.db")
117     cur = conn.cursor()
118     if type=="customer":
119         cur.execute("UPDATE customer SET phone=?, area=?, locality=?, city=?, state=?, country=?, zipcode=? where custID=?", (data["phone"],
120             data["area"],
121             data["locality"],
122             data["city"],
123             data["state"],
124             data["country"],
125             data["zip"],
126             userid))
127     elif type=="seller":
128         cur.execute("UPDATE seller SET phone=?, area=?, locality=?, city=?, state=?, country=?, zipcode=? where sellID=?", (data["phone"],
129             data["area"],
130             data["locality"],
131             data["city"],
132             data["state"],
133             data["country"],
134             data["zip"],
135             userid))
136     conn.commit()
137     conn.close()
138
139 def check_psswd(psswd, userid, type):
140     conn = sqlite3.connect("ARHA/ARHA.db")
141     cur = conn.cursor()
142     if type=="customer":
143         a = cur.execute("SELECT password FROM customer WHERE custID=?", (userid,))
144     elif type=="seller":
145         a = cur.execute("SELECT password FROM seller WHERE sellID=?", (userid,))

```

```

143     elif type=="Seller":
144         a = cur.execute("SELECT password FROM seller WHERE sellID=?", (userid,))
145         real_psswd = list(a)[0][0]
146         conn.close()
147         return psswd==real_psswd
148
149     def set_psswd(psswd, userid, type):
150         conn = sqlite3.connect("ARHA/ARHA.db")
151         cur = conn.cursor()
152         if type=="Customer":
153             a = cur.execute("UPDATE customer SET password=? WHERE custID=?", (psswd, userid))
154         elif type=="Seller":
155             a = cur.execute("UPDATE seller SET password=? WHERE sellID=?", (psswd, userid))
156         conn.commit()
157         conn.close()
158
159     def add_prod(sellID, data):
160         conn = sqlite3.connect("ARHA/ARHA.db")
161         cur = conn.cursor()
162         prodID = gen_prodID()
163         tup = (prodID,
164               data["name"],
165               data["qty"],
166               data["category"],
167               data["price"],
168               data["price"],
169               data["desp"],
170               sellID)
171         cur.execute("INSERT INTO product VALUES (?, ?, ?, ?, (SELECT profit_rate from metadata)*?, ?, ?)", tup)
172         conn.commit()
173         conn.close()
174
175     def get_categories(sellID):
176         conn = sqlite3.connect("ARHA/ARHA.db")
177         cur = conn.cursor()
178         a = cur.execute("SELECT DISTINCT(category) from product where sellID=?", (sellID,))
179         categories = [i[0] for i in a]

```

```

177     cur = conn.cursor()
178     a = cur.execute("SELECT DISTINCT(category) from product where sellID=?", (sellID,))
179     categories = [i[0] for i in a]
180     conn.close()
181     return categories
182
183     def search_myproduct(sellID, srchBy, category, keyword):
184         conn = sqlite3.connect("ARHA/ARHA.db")
185         cur = conn.cursor()
186         keyword = ['%'+i+'%' for i in keyword.split()]
187         if len(keyword)==0: keyword.append('%')
188         if srchBy=="by category":
189             a = cur.execute("""SELECT prodID, name, quantity, category, cost_price
190                             FROM product WHERE category=? AND sellID=? """, (category, sellID))
191             res = [i for i in a]
192         elif srchBy=="by keyword":
193             res = []
194             for word in keyword:
195                 a = cur.execute("""SELECT prodID, name, quantity, category, cost_price
196                                 FROM product
197                                 WHERE (name LIKE ? OR description LIKE ? OR category LIKE ?) AND sellID=? """,
198                                   (word, word, word, sellID))
199                 res += list(a)
200             res = list(set(res))
201         elif srchBy=="both":
202             res = []
203             for word in keyword:
204                 a = cur.execute("""SELECT prodID, name, quantity, category, cost_price
205                                 FROM product
206                                 WHERE (name LIKE ? OR description LIKE ?) AND sellID=? AND category=? """,
207                                   (word, word, sellID, category))
208                 res += list(a)
209             res = list(set(res))
210         conn.close()
211         return res
212
213     def get_product_info(id):

```

```

213 def get_product_info(id):
214     conn = sqlite3.connect('ARHA/ARHA.db')
215     cur = conn.cursor()
216     a = cur.execute("""SELECT p.name, p.quantity, p.category, p.cost_price, p.sell_price,
217                       p.sellID, p.description, s.name FROM product p JOIN seller s
218                       WHERE p.sellID=s.sellID AND p.prodID=? """, (id,))
219     res = [i for i in a]
220     conn.close()
221     if len(res)==0:
222         return False, res
223     return True, res[0]
224
225 def update_product(data, id):
226     conn = sqlite3.connect('ARHA/ARHA.db')
227     cur = conn.cursor()
228     cur.execute("""UPDATE product
229 SET name=?, quantity=?, category=?, cost_price=?,
230 sell_price=(SELECT profit_rate from metadata)*?, description=?
231 where prodID=?""", (data['name'],
232                     data['qty'],
233                     data['category'],
234                     data['price'],
235                     data['price'],
236                     data['desp'],
237                     id))
238     conn.commit()
239     conn.close()
240
241 def search_products(srchBy, category, keyword):
242     conn = sqlite3.connect("ARHA/ARHA.db")
243     cur = conn.cursor()
244     keyword = ['%'+i+'%' for i in keyword.split()]
245     if len(keyword)==0: keyword.append('%')
246     if srchBy=="by category":
247         a = cur.execute("""SELECT prodID, name, category, sell_price
248 FROM product WHERE category=? AND quantity!=0 """, (category,))

```

```

241 def search_products(srchBy, category, keyword):
242     conn = sqlite3.connect("ARHA/ARHA.db")
243     cur = conn.cursor()
244     keyword = ['%'+i+'%' for i in keyword.split()]
245     if len(keyword)==0: keyword.append('%')
246     if srchBy=="by category":
247         a = cur.execute("""SELECT prodID, name, category, sell_price
248 FROM product WHERE category=? AND quantity!=0 """, (category,))
249     res = [i for i in a]
250     elif srchBy=="by keyword":
251         res = []
252         for word in keyword:
253             a = cur.execute("""SELECT prodID, name, category, sell_price
254 FROM product
255 WHERE (name LIKE ? OR description LIKE ? OR category LIKE ?) AND quantity!=0 """,
256 (word, word, word))
257         res += list(a)
258     res = list(set(res))
259     elif srchBy=="both":
260         res = []
261         for word in keyword:
262             a = cur.execute("""SELECT prodID, name, category, sell_price
263 FROM product
264 WHERE (name LIKE ? OR description LIKE ?) AND quantity!=0 AND category=? """,
265 (word, word, category))
266         res += list(a)
267     res = list(set(res))
268     conn.close()
269     return res
270
271 def get_seller_products(sellID):
272     conn = sqlite3.connect('ARHA/ARHA.db')
273     cur = conn.cursor()
274     a = cur.execute("SELECT prodID, name, category, sell_price FROM product WHERE sellID=? AND quantity!=0", (sellID,))
275     res = [i for i in a]
276     conn.close()
277     return res

```

```

270
271 def get_seller_products(sellID):
272     conn = sqlite3.connect('ARHA/ARHA.db')
273     cur = conn.cursor()
274     a = cur.execute("SELECT prodID, name, category, sell_price FROM product WHERE sellID=? AND quantity!=0", (sellID,))
275     res = [i for i in a]
276     conn.close()
277     return res
278
279 def place_order(prodID, custID, qty):
280     conn = sqlite3.connect('ARHA/ARHA.db')
281     cur = conn.cursor()
282     orderID = gen_orderID()
283     cur.execute("""INSERT INTO orders
284                 SELECT ?,?,?,?,datetime('now'), cost_price*, sell_price*, 'PLACED'
285                 FROM product WHERE prodID=? """, (orderID, custID, prodID, qty, qty, qty, prodID))
286     conn.commit()
287     conn.close()
288
289 def cust_orders(custID):
290     conn = sqlite3.connect('ARHA/ARHA.db')
291     cur = conn.cursor()
292     a = cur.execute("""SELECT o.orderID, o.prodID, p.name, o.quantity, o.sell_price, o.date, o.status
293                     FROM orders o JOIN product p
294                     WHERE o.prodID=p.prodID AND o.custID=? AND o.status!= 'RECIEVED'
295                     ORDER BY o.date DESC """, (custID,))
296     res = [i for i in a]
297     conn.close()
298     return res
299
300 def sell_orders(sellID):
301     conn = sqlite3.connect('ARHA/ARHA.db')
302     cur = conn.cursor()
303     a = cur.execute("""SELECT o.orderID, o.prodID, p.name, o.quantity, p.quantity, o.cost_price, o.date, o.status
304                     FROM orders o JOIN product p
305                     WHERE o.prodID=p.prodID AND p.sellID=? AND o.status!= 'RECIEVED'
306                     ORDER BY o.date DESC """, (sellID,))

```

```

307     res = [i for i in a]
308     conn.close()
309     return res
310
311 def get_order_details(orderID):
312     conn = sqlite3.connect('ARHA/ARHA.db')
313     cur = conn.cursor()
314     a = cur.execute("""SELECT o.custID, p.sellID, o.status FROM orders o JOIN product p
315                     WHERE o.orderID=? AND o.prodID=p.prodID """, (orderID,))
316     res = [i for i in a]
317     conn.close()
318     return res
319
320 def change_order_status(orderID, new_status):
321     conn = sqlite3.connect('ARHA/ARHA.db')
322     cur = conn.cursor()
323     cur.execute("UPDATE orders SET status=? WHERE orderID=? ", (new_status, orderID))
324     if new_status=='DISPATCHED':
325         cur.execute("""UPDATE product SET
326                     quantity=quantity-(SELECT quantity FROM orders WHERE orderID=? )
327                     WHERE prodID=(SELECT prodID FROM orders WHERE orderID=? )""", (orderID, orderID))
328     conn.commit()
329     conn.close()
330
331 def cust_purchases(custID):
332     conn = sqlite3.connect('ARHA/ARHA.db')
333     cur = conn.cursor()
334     a = cur.execute("""SELECT o.prodID, p.name, o.quantity, o.sell_price, o.date
335                     FROM orders o JOIN product p
336                     WHERE o.prodID=p.prodID AND o.custID=? AND o.status='RECIEVED'
337                     ORDER BY o.date DESC """, (custID,))
338     res = [i for i in a]
339     conn.close()
340     return res
341
342 def sell_sales(sellID):

```

```

344     cur = conn.cursor()
345     a = cur.execute("""SELECT o.prodID, p.name, o.quantity, o.sell_price, o.date, o.custID, c.name
346                       FROM orders o JOIN product p JOIN customer c
347                       WHERE o.prodID=p.prodID AND o.custID=c.custID AND p.sellID=? AND o.status='RECEIVED'
348                       ORDER BY o.date DESC """, (sellID,))
349     res = [i for i in a]
350     conn.close()
351     return res
352
353 def add_product_to_cart(prodID, custID):
354     conn = sqlite3.connect('ARHA/ARHA.db')
355     cur = conn.cursor()
356     cur.execute("""INSERT INTO cart VALUES (?,?,1) """, (custID, prodID))
357     conn.commit()
358     conn.close()
359
360 def get_cart(custID):
361     conn = sqlite3.connect('ARHA/ARHA.db')
362     cur = conn.cursor()
363     a = cur.execute("""SELECT p.prodID, p.name, p.sell_price, c.sum_qty, p.quantity
364                       FROM (SELECT custID, prodID, SUM(quantity) AS sum_qty FROM cart
365                             GROUP BY custID, prodID) c JOIN product p
366                       WHERE p.prodID=c.prodID AND c.custID=?""", (custID,))
367     res = [i for i in a]
368     conn.close()
369     return res
370
371 def update_cart(custID, qty):
372     conn = sqlite3.connect('ARHA/ARHA.db')
373     cur = conn.cursor()
374     for prodID in qty:
375         cur.execute("DELETE FROM cart WHERE prodID=? AND custID=?", (prodID, custID))
376         cur.execute("INSERT INTO cart VALUES (?,?,?)", (custID, prodID, qty[prodID]))
377     conn.commit()
378     conn.close()
379
380 def cart_purchase(custID):

```

```

369     return res
370
371 def update_cart(custID, qty):
372     conn = sqlite3.connect('ARHA/ARHA.db')
373     cur = conn.cursor()
374     for prodID in qty:
375         cur.execute("DELETE FROM cart WHERE prodID=? AND custID=?", (prodID, custID))
376         cur.execute("INSERT INTO cart VALUES (?,?,?)", (custID, prodID, qty[prodID]))
377     conn.commit()
378     conn.close()
379
380 def cart_purchase(custID):
381     conn = sqlite3.connect('ARHA/ARHA.db')
382     cur = conn.cursor()
383     cart = get_cart(custID)
384     for item in cart:
385         orderID = gen_orderID()
386         prodID = item[0]
387         qty = item[3]
388         cur.execute("""INSERT INTO orders
389                       SELECT ?,?,?,?,datetime('now'), cost_price*?, sell_price*?, 'PLACED'
390                       FROM product WHERE prodID=? """, (orderID, custID, prodID, qty, qty, qty, prodID))
391         cur.execute("DELETE FROM cart WHERE custID=? AND prodID=?", (custID, prodID))
392     conn.commit()
393     conn.close()
394
395 def empty_cart(custID):
396     conn = sqlite3.connect('ARHA/ARHA.db')
397     cur = conn.cursor()
398     cur.execute("DELETE FROM cart WHERE custID=?", (custID,))
399     conn.commit()
400
401 def remove_from_cart(custID, prodID):
402     conn = sqlite3.connect('ARHA/ARHA.db')
403     cur = conn.cursor()
404     cur.execute("DELETE FROM cart WHERE custID=? AND prodID=?", (custID, prodID))
405     conn.commit()

```

The database to store customer information looks something like this:

100 %

Results Messages

	Id	Name	DisplayOrder	CreatedDateTime
1	1	Tops	2	2022-01-15 00:00:00.0000000
2	2	Disposable Plates	2	2022-01-15 00:00:00.0000000
3	3	Wall Decor	2	2022-01-16 00:00:00.0000000
4	4	Toys	1	2022-01-17 00:00:00.0000000
5	5	Cookware	6	2022-01-18 00:00:00.0000000
6	6	Clothes	1	2022-01-19 00:00:00.0000000

ARHADB 1 X

SELECT \* FROM ARHADB *Enter a SQL expression to filter results (use Ctrl+Space)*

	abc id	abc name	123 mobile	abc gender	abc location
1	lala@gmail.com	MANJU	24,561	F	CHENNAI
2	mamy@gmail.com	Srimathi	24,561	F	Hyderabad
3	bob@gmail.com	Divya	24,561	F	Kanyakumari
4	run@gmail.com	Balaji	24,561	M	odisha
5	dada@gmail.com	Ajay	24,561	M	jammu
6	aruna@gmail.com	Mohan	12,345	M	kerala
7	whywhy@gmail.com	Karthik	12,345	M	goa