class User:

def \_\_init\_\_(self, username, password):

self.username = username

self.password = password

self.cart = []

class Admin:

def \_\_init\_\_(self, username, password):

self.username = username

self.password = password

class Product:

def \_\_init\_\_(self, name, product\_id, price):

self.name = name

self.product\_id = product\_id

self.price = price

class OnlineMarketplace:

def \_\_init\_\_ (self):

self.users = []

self.logged\_in\_users = None

self.admins = []

self.products = []

self.nextproductid = 1

self.categories = []

def new\_user (self, username, password):

user = User(username, password)

self.users.append(user)

def new\_admin(self, username, password):

admin = Admin(username, password)

self.admins.append(admin)

def addproduct(self, name, price):

if not isinstance(self.logged\_in\_users, Admin):

return "Need Admin priviledges"

product = Product(self.nextproductid, name, price)

self.products.append(product)

self.nextproductid += 1

return "Product added successfully"

def modifyproduct(self, product\_id, name, price):

if not self.logged\_in\_users or not isinstance(self.logged\_in\_users, Admin):

return "Need Admin privileges"

index = next((index for index, p in enumerate(self.products) if p.product\_id == product\_id), None)

if index is not None:

self.products[index].name = name

self.products[index].price = price

return "Product modified"

else:

return "No product found"

def removeproduct(self, product\_id):

if not self.logged\_in\_users or not isinstance(self.logged\_in\_users, Admin):

return "Need Admin priviledges"

product = next((p for p in self.products if p.product\_id == product\_id), None)

if product:

print(f"Product Found: {product.product\_id}")

return "Product removed"

else:

print(f"No product found with that ID: {product.product\_id}")

return "No Product Found"

def newproduct(self, name, price):

product = Product(self.nextproductid, name, price)

self.products.append(product)

self.nextproductid += 1

def view\_products(self):

print("List of Products: ")

for product in self.products:

print(f"Product ID: {product.name}, Name: {product.product\_id}, Price: {product.price} ")

def login(self, username, password, role):

if role == 'user':

for user in self.users:

if user.username == username and user.password == password:

self.loggedusers = user

return True

elif role == 'admin':

for admin in self.admins:

if admin.username == username and admin.password == password:

self.loggedusrs = admin

return True

return False

def adminlogin(self, username, password):

admin = next((a for a in self.admins if a.username == username and a.password == password), None)

if admin:

self.logged\_in\_users = admin

return True

else:

return False

def userinput(self):

username = input("Please Enter Your Username: ")

password = input("Please Enter Your Password: ")

role = input("Please enter your role (User or Admin): ").lower()

if role == 'user':

if self.login(username, password, 'user'):

print("User Login Successful")

else:

print("User Log In Failed. Try Again")

elif role == 'admin':

if self.adminlogin(username, password):

print("Admin Login Successful")

else:

print("Admin Log In Failed. Try Again")

else:

print("Role Invalid. Try Again")

def adminops(self):

if not self.loggedusers or not isinstance(self.loggedusers, Admin):

return "Need Admin priviledges "

def cart(self, product\_id, quantity):

if not isinstance(self.loggedusers, User):

return "Only users can add items to the cart"

product = next((p for p in self.products if p.product\_id == product\_id), None)

if product:

self.loggedusers.cart.append((product, quantity))

return f"{quantity} {product.product\_id}(s) have been added to the cart."

else:

return "Product is not available."

def usercart(self):

if not isinstance(self.loggedusers, User):

print("First log in and then try again")

return

while True:

market.access\_cart()

selfinput = input("Enter 'Add' to add items to the cart or enter 'Remove' to remove items from the cart (type 'Done' to exit) ")

if selfinput == 'Done':

break

if selfinput == 'Add':

market.view\_products()

product\_id = (input("Enter the ID of the product you wish to add to the cart (Type 'Done' to exit): "))

if product\_id == 'Done':

break

if any(product.product\_id == product\_id for product in market.products):

quantity = int(input("Enter the quantity of the product: "))

result = self.cart(product\_id, quantity)

print (result)

else:

print("Incorrect Product ID - please try again")

elif selfinput == 'Remove':

market.access\_cart()

product\_id = (input("Enter the ID of the product you wish to remove from the cart (Type Done to exit): "))

if any ((product.product\_id == product\_id) for product, \_ in self.loggedusers.cart):

result = self.lesscart(product\_id)

print(result)

else:

print("No product found within your cart")

else:

print("Incorrect input - Please type either 'Add' 'Remove' or 'Done' ")

def access\_cart(self):

if not isinstance(self.loggedusers, User):

print("First log in as a user and then try again:")

return

if not self.loggedusers.cart:

print("The cart is currently empty")

return

print("Items currently in cart: ")

for product, quantity in self.loggedusers.cart:

print(f"{product.product\_id}, Quantity: {quantity}, Price per each: ${product.price}")

def removeusercart(self):

if not isinstance(self.loggedusers, User):

print("First log in and then try again: ")

return

product\_id = (input("Enter the ID of the product you wish to remove from the cart: "))

result = self.lesscart(product\_id)

print(result)

def lesscart(self, product\_id):

if not isinstance(self.loggedusers, User):

return "First log in and then try again"

removed\_item = None

for item in self.loggedusers.cart:

if item[0].product\_id == product\_id:

removed\_item = item

break

if removed\_item:

self.loggedusers.cart.remove(removed\_item)

return f"{removed\_item[0].product\_id} has been removed from the cart."

return "No Product found within the cart."

def checkout(self):

if not isinstance(self.loggedusers, User):

return "Only users are able to checkout"

if not self.cart:

return "Cart currently empty. Please add items"

print("Payment Options")

print("1. Card")

print("2. PayPal")

print("3. UPI")

print("4. Cancel")

payoptions = input("Select a payment option: ")

if payoptions == '1':

totalprice = sum(product.price \* quantity for product,quantity in self.loggedusers.cart)

print("Redirecting to card payment..")

print (f"Order Placed Successfully, Your Total reciept is ${totalprice}")

elif payoptions == '2':

totalprice = sum(product.price \* quantity for product,quantity in self.loggedusers.cart)

print("Redirecting to PayPal portal..")

print (f"Order Placed Successfully, Your Total reciept is ${totalprice}")

elif payoptions == '3':

totalprice = sum(product.price \* quantity for product,quantity in self.loggedusers.cart)

print("Redirecting to UPI Portal..")

print (f"Order Placed Successfully, Your Total reciept is ${totalprice}")

elif payoptions == '4':

print("Checkout cancelled")

else:

print("Invalid selection. Try Again")

self.loggedusers.cart = []

def showusermenu(self):

while True:

print("Here is the User Menu:")

print("1. View Products")

print("2. Add Product to Cart")

print("3. Remove Products from the Cart:")

print("4. View Cart")

print("5. Checkout")

print("6. Exit")

select = input("Enter your selection: ")

if select == '1':

self.view\_products()

elif select == '2':

self.usercart()

elif select == '3':

self.removeusercart()

elif select == '4':

self.access\_cart()

elif select == '5':

checkoutresult = self.checkout()

print (checkoutresult)

elif select == '6':

print ("Leaving Menu")

break

else:

print("Choice invalid. Try Again")

def showadminmenu(self):

while True:

print("Here is the Admin Menu:")

print("1. Add Products")

print("2. Modify Products")

print("3. Remove Products")

print("4. Exit")

adminselect = input("Enter your selection: ")

if adminselect == '1':

name = input("Enter the Product Name: ")

price = float(input("Enter the Price: "))

print(self.addproduct(name, price))

elif adminselect == '2':

product\_id = input("Enter the Product ID you want to modify: ")

name = input("Enter the New Name: ")

price = float(input("Enter the New Price: "))

print(self.modifyproduct(product\_id, name, price))

elif adminselect == '3':

product\_id = input("Enter the Product ID you wish to Remove: ")

print(self.removeproduct(product\_id))

elif adminselect == '4':

print ("Leaving Menu")

break

else:

print("Choice invalid. Try Again")

def mainmenu(self):

while True:

print("Welcome to the Online Marketplace!")

print("1. User Login")

print("2. Admin Login")

print("3. Exit")

select = input("Enter a selection: ")

if select == '1':

self.userinput()

if isinstance(self.loggedusers, User):

self.showusermenu()

else:

print("Login Failed")

elif select == '2':

self.userinput()

if isinstance(self.logged\_in\_users, Admin):

self.showadminmenu()

else:

print("Login Failed")

elif select == '3':

print("Exiting Menu")

break

else:

print("Selection Invalid. Try Again")

market = OnlineMarketplace()

market.new\_user("NewUser", "Shopping")

market.new\_admin("NewAdmin", "Manager")

market.newproduct("Sports Jersey", 125)

market.newproduct("Sunglasses", 85)

market.newproduct("Watch", 120)

market.newproduct("Running Shoes", 110)

market.mainmenu()