Nancy's Shopping App

Thank you for viewing my shoppint app! I have defined all required functionality for my shopping app in the first section. You may find my final shopping app in the last section of this document.

Section 1: Defining User and Admin Functions

User and Admin Logins

In [ ]:
```
# user and admin login demo login

user_access = {
    # username : password
    'userA': 'passA',
    'userB': 'passB',
    'userC': 'passC'
}

admin_access = {
    # username : password
    'auserA': 'apassA',
    'auserB': 'apassB',
    'auserC': 'apassC'
}

session_id = {
    'userA': '12345',
    'userB': '12346',
    'userC': '12347',
    'auserA': '12348',
    'auserB': '12349',
    'auserC': '12350'
}
```

In [ ]:
```
# create random number generator for new session IDs
from random import randint
def randNumGen():
    randNum = randint(11111,99999)
    if randNum in session_id.values(): #update use 'while'
        randNum = randint(11111,99999)
        return(randNum)
    else:
        return(randNum)

randNumGen()
```

In [ ]:
```
# admin login function

def admin_login():
    user = input('Enter your username.')
    password = input('Enter your password.')
    global sessionID

    # check if the password is available in the user dictionary
    if user in admin_access and admin_access[user] == password:
        sessionID = session_id[user]
        print("Welcome to Nancy's Marketplace, ", user,'!')
        print("Your session has been verified. Your session ID is:", sessionID, "."
    else:
        newAccount = input("Would you like to create a new user login?")
        if newAccount == 'Yes':
            admin_access[user] = password
            sessionID = randNumGen()
            session_id[user] = sessionID
            print("Congratulations, your new account has been created,", user,"!")
            print("Your new session ID is:", sessionID,".")
        else:
            print("Access Denied: Incorrect username and password. \nPlease review
            sessionID = 0

admin_login()
```

In [ ]:

```
# user logins function
def user_login():
    user = input('Enter your username.')
    password = input('Enter your password.')
    global sessionID

    # check if the password is available in the user dictionary
    if user in user_access and user_access[user] == password:
        sessionID = session_id[user]
        print("Welcome to Nancy's Marketplace, {}! \nYour session has been verified
    else:
        newAccount = input("Account not recognized. Would you like to create a new
        if newAccount == 'Yes':
            user_access[user] = password
            sessionID = randNumGen()
            session_id[user] = sessionID
            print("Congratulations, your new account has been created, {}! \nYour n
        else:
            print("Access Denied: Incorrect username and password. \nPlease review
            sessionID = 0

user_login()
```

Define Sample Catalog

In [ ]:

```
# nested dict
sample_catalog = {
    1: {'productID': 1, 'categoryID': 'Soda', 'name': 'Coca Cola', 'price': 4.99},
    2: {'productID': 2, 'categoryID': 'Soda', 'name': 'Diet Coca Cola', 'price': 4.
    3: {'productID': 3, 'categoryID': 'Soda', 'name': 'Sprite', 'price': 4.99},
    4: {'productID': 4, 'categoryID': 'Liquor', 'name': 'Tequila', 'price': 83.99},
    5: {'productID': 5, 'categoryID': 'Liquor', 'name': 'Whiskey', 'price': 79.99},
    6: {'productID': 6, 'categoryID': 'Liquor', 'name': 'Gin', 'price': 54.99},
    7: {'productID': 7, 'categoryID': 'Beer', 'name': 'Stout', 'price': 8},
    8: {'productID': 8, 'categoryID': 'Beer', 'name': 'Porter', 'price': 7},
    9: {'productID': 9, 'categoryID': 'Beer', 'name': 'IPA', 'price': 9},
    10: {'productID': 10, 'categoryID': 'Cider', 'name': 'Raspberry Cider', 'price'
    11: {'productID': 11, 'categoryID': 'Cider', 'name': 'Strawberry Cider', 'price
    12: {'productID': 12, 'categoryID': 'Cider', 'name': 'Blueberry Cider', 'price'
}

sample_catalog
```

In [ ]:

```
# build a  function to display the items in the catalog
def displayCatalog():
    print("Nancy's Marketplace Catalog")
    #loop through each item
    for items in sample_catalog:
        print("productID: {}, name: {}, categoryID: {}, price: {}".format(sample_ca

displayCatalog()
```

User cart functionality Functions

In [ ]:

```
#%pip install pandas
import pandas as pd
from pandas import DataFrame

# build an empty cart
cart_columns = ["SessionID", "ProductID", "Quantity", "Price", "Category", "Name"]
cart = pd.DataFrame(columns=cart_columns)

#create funtion for adding to cart
def add_to_cart(productID, quantity, sessionID = sessionID):
    global cart

    if productID not in list(sample_catalog.keys()):
        print("You are attempting to add a product that is not in the catalog. Plea
    else:
        category = sample_catalog[productID]["categoryID"]
        name = sample_catalog[productID]["name"]
        priceByQuantity = float(sample_catalog[productID]["price"])*quantity

        #check if product already exists for existing sessionID
        if ((cart['ProductID'] == productID) & (cart["SessionID"] == sessionID)).an
            # Update quantity for the existing product
            cart.loc[(cart['ProductID'] == productID) & (cart["SessionID"] == sessi
            cart.loc[(cart['ProductID'] == productID) & (cart["SessionID"] == sessi
            sessionCart = cart[cart["SessionID"] == sessionID]
        else:
            # Create a new entry for the product
            newProduct = pd.DataFrame([[sessionID, productID, quantity, priceByQuan
            if cart.empty:
                cart = newProduct
                # define what got added
                sessionCart = cart[cart["SessionID"] == sessionID]
            else:
                cart = pd.concat([cart, newProduct], ignore_index=True)
                # define what got added
                sessionCart = cart[cart["SessionID"] == sessionID]
            #print(newProduct)
        print("You have added: {} {} {}(s).\n Your cart contains: \n{}".format(quan
```

In [ ]:
```
add_to_cart(16,4, sessionID)
```

In [ ]:
```
#create funtion for removing items from cart
def remove_from_cart(productID, quantity, sessionID = sessionID):
    global cart
    category = sample_catalog[productID]["categoryID"]
    name = sample_catalog[productID]["name"]
    priceByQuantity = float(sample_catalog[productID]["price"])*quantity

    #check if product already exists for existing sessionID
    if ((cart['ProductID'] == productID) & (cart["SessionID"] == sessionID) & (cart
        # Update quantity for the existing product
        cart.loc[(cart['ProductID'] == productID) & (cart["SessionID"] == sessionID
        cart.loc[(cart['ProductID'] == productID) & (cart["SessionID"] == sessionID
        sessionCart = cart[(cart["SessionID"] == sessionID) & (cart["Quantity"] !=
        print("You have removed: {} {} {}(s).".format(quantity, category, name))
        if sessionCart.empty == True:
            print("Your cart is now empty.")
        else:
            print("Your cart contains: \n{}".format(sessionCart))
    else:
        # Error message
        print("Unable to remove desired product from your cart. \n Please make sure
```

In [ ]:
```
remove_from_cart(1,2, sessionID)
```

In [ ]:
```
##Display session cart
def displayCart(sessionID=sessionID):
    sessionCart = cart[(cart["SessionID"] == sessionID) & (cart["Quantity"] != 0)]
    return(sessionCart)
```

In [ ]:
```
displayCart(sessionID)
```

In [ ]:
```
def checkout(sessionID = sessionID):
    # sum total cost of session cart
    totalPrice = round(sum(displayCart(sessionID)["Price"]),2)

    if totalPrice in (0,"0"):
        print("Your cart is empty. Please make sure there are items in your cart be
    else:
        print('''Your total is ${}. How would you like to pay?
            1) Credit Card
            2) PayPal
            3) UPI'''.format(totalPrice))



        pay_meth = input("Please select your payment method from the available opti
        if pay_meth in ('Credit Card', 1, "1"):
            print("You will shortly be redirected to enter your credit card informa
            # reset cart once payment is complete
            cart.drop(index=cart.index, inplace=True)
        elif pay_meth in ('PayPal', 2, "2"):
            print("Your oder went through using PayPal. Thank you for shopping at N
            # reset cart once payment is complete
            cart.drop(index=cart.index, inplace=True)
        elif pay_meth in ('UPI', 3, "3"):
            print("You will be shortly redirected to the portal for Unified Payment
            # reset cart once payment is complete
            cart.drop(index=cart.index, inplace=True)
        else:
            print("Please try again. Make sure you are selecting between the availa
```

In [ ]:

```
checkout(sessionID)
```

In [ ]:

```
# user functionality -- I will be using this functionality in my final shopping App

print('''
        1. View Nancy's Marketplace shopping catalog.
        2. Add item(s) to yout cart.
        3. Remove item(s) from your cart.
        4. View your shopping cart.
        5. Checkout.
''')
choice = input('What would you like to do next?')
if choice in ("1",1):
    displayCatalog()
elif choice in ("2", 2):
    productID = int(input("What is the productID for the item you would like to add
    quantity = int(input("How many of that item would you like to add?"))
    add_to_cart(productID,quantity, sessionID)
elif choice in ("3", 3):
    productID = int(input("What is the productID for the item you would like to rem
    quantity = int(input("How many of that item would you like to remove?"))
    remove_from_cart(productID,quantity, sessionID)
elif choice in ("4",4):
    print(displayCart(sessionID))
elif choice in ("5",5):
    checkout(sessionID)
else:
    print("Action not supported. Please review the options carefully and select bet
```

Admin Functionality Functions

In [ ]:

```
##Admin add product to catalog

def add_product(sample_catalog=sample_catalog):
    print("Please edit carefully as your modifications will permanently alter the c

    prodCat = input("What is the category of the item you would like to add?")
    prodName = input("What is the name of the item you would like to add?")
    prodPrice = input("What is the unit price for the item you would like to add?")

    # check to see if product category and name already exists within catalog
    matching_entries = [product_id for product_id, product_info in sample_catalog.i
                    if product_info['categoryID'] == prodCat and product_info['name

    # if it exists, ask admin to double check new item, or consider modifying the c
    if matching_entries:
        print("This product category and name already exists in your catalog. Revie
    else:
        prodNum = list(sample_catalog.keys())[-1] + 1
        newproduct = {'productID': prodNum, 'categoryID': prodCat, 'name': prodName
        sample_catalog[prodNum] = newproduct
        print("You have successfully added a new product! New product added: \n{}".
```

In [ ]:

```
add_product()
```

In [ ]:

```
## admin remove product from catalog

def remove_product(sample_catalog=sample_catalog):
    print("Please edit carefully as your modifications will permanently alter the c

    prodID = input("What is the productID of the item you would like to delete?")

    # check to see if product category and name already exists within catalog
    matching_entries = [product_id for product_id, product_info in sample_catalog.i
                    if product_info['productID'] == int(prodID)]

    # if it exists, ask admin to double check new item, or consider modifying the c
    if not matching_entries:
        # productID does not exist
        print("ProductID odes not exist in the catalog. Review the product details
    else:
        removedproduct = matching_entries[0]
        product = sample_catalog[int(removedproduct)]
        sample_catalog.pop(removedproduct)
        print("You have successfully removed a product. Product removed: \n{}".form
```

In [ ]:

```
remove_product()
```

In [ ]:

```
## modify product from catalog

def modify_product(sample_catalog=sample_catalog):
    print("Please edit carefully as your modifications will permanently alter the c

    # identify the productID
    prodID = input("What is the productID of the item you would like to modify?")

    matching_entries = [product_id for product_id, product_info in sample_catalog.i
                    if product_info['productID'] == int(prodID)]

    if not matching_entries:
        # productID does not exist
        print("ProductID odes not exist. Review your new product details and try ag
    else:
        # save original product info for comparison after successful product modifi
        originalproduct = sample_catalog[int(prodID)]
        print("You have selected to modify product: \n{}".format(originalproduct))
        # identify what the admin would like to edit
        prodNameEdit = input("Would you like to modify the product name? (Yes/No)")
        prodCatEdit = input("Would you like to modify the product category? (Yes/No
        prodPriceEdit = input("Would you like to modify the product price? (Yes/No)

        # modify product details
        if prodNameEdit in ("Yes", "yes"):
            newprodName = input("What would you like to update the product name to?
            sample_catalog[int(prodID)]["name"] = newprodName
        if prodCatEdit in ("Yes", "yes"):
            newprodCat = input("What would you like to update the product category
            sample_catalog[int(prodID)]["categoryID"] = newprodCat
        if prodPriceEdit in ("Yes", "yes"):
            newprodPrice = input("What would you like to update the product price t
            sample_catalog[int(prodID)]["price"] = newprodPrice

        # save new product details for comparison after successful product modifica
        modifiedproduct = sample_catalog[int(prodID)]

        print("You have successfully modified this product to: \n{}".format(modifie
```

In [ ]:

```
modify_product()
```

In [ ]:

```
# admin functionality - I will be using this functionality in my final shopping App

print('''
        1. View Nancy's Marketplace shopping catalog.
        2. Add item(s) to the catalog.
        3. Remove item(s) from the catalog.
        4. Modify item(s) in the catalog.
        5. Exit.
''')
choice = input('What would you like to do next? Please select from the available op
if choice in ("1",1):
    displayCatalog()
elif choice in ("2", 2):
    add_product()
elif choice in ("3", 3):
    remove_product()
elif choice in ("4",4):
    modify_product()
elif choice in ("5",5):
    breakpoint
else:
    print("Action not supported. Please review the options carefully and select bet
```

Defining user type: user vs admin Function

In [ ]:

```
# mechanism for user type admin vs user
def UserAccessType():
    login_type = input('Pleasse enter your user type.')

    if login_type == 'user':
        user_login()
        #print("Your session is now active. Happy Shopping!")
        return login_type
    elif login_type =='admin':
        admin_login()
        #print("Your session is now active. Happy Shopping!") ##don't think admin n
        return login_type
    else:
        print("Unsupported user type. Please select between user and admin.")

UserAccessType()
```

Final Shopping App Function

In [ ]:
```
def shoppingApp():
    print('--------------------------------------------------------')
    print("Welcome to Nancy's Marketplace - we are thrilled to have you! \nPlease b
    print('--------------------------------------------------------')

    ##select user access type
    user_type = UserAccessType()

    if sessionID in (0,"0"):
        breakpoint
    else:
        if user_type == 'user':
            print('''
            1. View Nancy's Marketplace shopping catalog.
            2. Add item(s) to yout cart.
            3. Remove item(s) from your cart.
            4. View your shopping cart.
            5. Checkout and exit.
            ''')
            choice = input('What would you like to do first? Please select from the

            while choice in (1,2,3,4,5,"1","2","3","4","5"):
                if choice in ("1",1):
                    displayCatalog()
                    choice = input('What would you like to do next? Please select f
                elif choice in ("2", 2):
                    productID = int(input("What is the productID for the item you w
                    quantity = int(input("How many of that item would you like to a
                    add_to_cart(productID,quantity, sessionID)
                    choice = input('What would you like to do next? Please select f
                elif choice in ("3", 3):
                    productID = int(input("What is the productID for the item you w
                    quantity = int(input("How many of that item would you like to r
                    remove_from_cart(productID,quantity, sessionID)
                    choice = input('What would you like to do next? Please select f
                elif choice in ("4",4):
                    print(displayCart(sessionID))
                    choice = input('What would you like to do next? Please select f
                elif choice in ("5",5):
                    checkout(sessionID)
                    break
            else:
                print("Action not supported. Please review the options carefully an
        elif user_type == 'admin':
            print('''
            1. View Nancy's Marketplace shopping catalog.
            2. Add item(s) to the catalog.
            3. Remove item(s) from the catalog.
            4. Modify item(s) in the catalog.
            5. Sign out and exit.
            ''')

            choice = input('What would you like to do first? Please select from the

            while choice in (1,2,3,4,5,"1","2","3","4","5"):

                if choice in ("1",1):
                    displayCatalog()
                    choice = input('What would you like to do next? Please select f
                elif choice in ("2", 2):
                    add_product()
                    choice = input('What would you like to do next? Please select f
                elif choice in ("3", 3):
                    remove_product()
                    choice = input('What would you like to do next? Please select f
                elif choice in ("4",4):
                    modify_product()
                    choice = input('What would you like to do next? Please select f
                elif choice in ("5",5):
                    print("Your session is now over. Thank you!")
                    break
            else:
                print("Action not supported. Please review the options carefully an

        else:
            print("Unsupported user type. Please select between user and admin.")
            breakpoint
```

In [ ]:

```
shoppingApp()
```