storesipython

December 5, 2023

[3]:
```python
# 1.Lets start a class for Users
# 1.

# users atributs
#
class User:
  def __init__(self, username, password):
    self.username = username
    self.password = password
    self.cart = []
# add products to user cart
#
  def add_to_cart(self, product):
    self.cart.append(product)
# remove products from user cart
#
  def remove_from_cart(self, product):
    if product in self.cart:
      self.cart.remove(product)
# to see self cart
#
  def view_cart(self):
    return self.cart
```

[4]:
```python
# 2. Lets make a class Admin for to control the online store
# 2.            Admin


class Admin:
  # admin atributs
  #
  def __init__(self, username, password):
    self.username = username
    self.password = password
  # function for add new category of products
  #
  def add_category(self, categories, new_category):
```

```
    self.categories.append(new_category)
  # function to delete category of products
  #
  def del_category(self, categories, category):
    if category in categories:
      self.categories.remove(category)
  # function to add new product to catalog
  #
  def add_product(self, catalog, product):
    self.catalog.append(product)
  # function to remove product to catalog
  #
  def del_product(self, catalog, product_id):
    catalog = [product for product in catalog if product['id'] != product_id]
    return catalog
  #function to modify product information
  #
  def modify_product(self, catalog, product_id, new_price):
    for product in catalog:
      if product['id'] == product_id:
        product['price'] = new_price
```

[14]:
```
# now lets make the class to store or products
#

class AmazonasStore:
  # make a list for users ,admins, categories and catalog
  #
  def __init__(self):
    self.users = []
    self.admins = []
    self.catalog = [
        {'id': 1, 'name': 'Iphone', 'category': 'Eletronic', 'price': 999}
    ]
    self.categories = ['Eletronic','Footwear', 'Clothing', 'Electronics']

  # function to display Welcome mensage
  #
  def display_welcome(self):
    print("Welcome to Amazonas Store")

  # creating demo user and admin
  #
  def creat_demo_accounts(self):
    user = User("user", "password")
    admin = Admin("admin", "password")
    self.users.append(user)
```

```
    self.admins.append(admin)

# function to user login
#

  def user_login(self, username, password):
    for user in self.users:
      if user.username == username and user.password == password:
        return user

    return None

  # function to admin login
  #

  def admin_login(self, username, password):
    for admin in self.admins:
      if admin.username == username and admin.password == password:
        return admin

    return None

  #function to display catalog
  #
  def display_catalog(self):
    print("Product Catalog:")
    for product in self.catalog:
      print(f"{product['id']}. {product['name']} - {product['category']} -␣
↪${product['price']}")

  # function to display categories
  #
  def display_categories(self):
    print("Categories: ", self.categories)


  # function to start main
  # main
  def main(self):
          self.display_welcome()
          self.creat_demo_accounts()
          # choice User, Admin, and Exit
          #
          while True:
              print("\n1. User Login\n2. Admin Login\n3. Exit")
              choice = int(input("Enter your choice: "))
              # Program for user
```

```
            #
            if choice == 1:
                username = input("Enter username: ")
                password = input("Enter password: ")
                user = self.user_login(username, password)

                if user:
                    while True:
                        print("\n1. View Catalog\n2. View Cart\n3. Add to␣
↪Cart\n4. Remove from Cart\n5. Checkout\n6. Logout")
                        user_choice = int(input("Enter your choice: "))

                        if user_choice == 1:
                            self.display_catalog()
                        elif user_choice == 2:
                            print("Cart:", user.view_cart())
                        elif user_choice == 3:
                            self.display_catalog()
                            product_id = int(input("Enter the product ID to␣
↪add to cart: "))
                            product = next((p for p in self.catalog if␣
↪p['id'] == product_id), None)
                            if product:
                                user.add_to_cart(product)
                                print(f"{product['name']} added to cart.")
                            else:
                                print("Invalid product ID.")
                        elif user_choice == 4:
                            print("Cart:", user.view_cart())
                            product_id = int(input("Enter the product ID to␣
↪remove from cart: "))
                            product = next((p for p in self.catalog if␣
↪p['id'] == product_id), None)
                            if product:
                                user.remove_from_cart(product)
                                print(f"{product['name']} removed from cart.")
                            else:
                                print("Invalid product ID.")
                        elif user_choice == 5:
                            print("Checkout functionality not implemented in␣
↪this demo.")
                        elif user_choice == 6:
                            break
                        else:
                            print("Invalid choice.")
```

```
                else:
                    print("Invalid username or password.")

            elif choice == 2:
                username = input("Enter admin username: ")
                password = input("Enter admin password: ")
                admin = self.admin_login(username, password)
                # program for admin
                #
                if admin:
                    while True:
                        print("\n1. View Catalog\n2. Add Category\n3. Remove␣
↪Category\n4. Add Product\n"
                              "5. Modify Product\n6. Remove Product\n7.␣
↪Logout")
                        admin_choice = int(input("Enter your choice: "))

                        if admin_choice == 1:
                            self.display_catalog()
                        elif admin_choice == 2:
                            new_category = input("Enter the new category: ")
                            self.admin.add_category(self.categories,␣
↪new_category)
                            print("Category added successfully.")
                        elif admin_choice == 3:
                            self.display_categories()
                            category = input("Enter the category to remove: ")
                            self.admin.remove_category(self.categories,␣
↪category)
                            print("Category removed successfully.")
                        elif admin_choice == 4:
                            product = {
                                'id': len(self.catalog) + 1,
                                'name': input("Enter product name: "),
                                'category': input("Enter product category: "),
                                'price': float(input("Enter product price: "))
                            }
                            admin.add_product(self.catalog, product)
                            print("Product added successfully.")
                        elif admin_choice == 5:
                            self.display_catalog()
                            product_id = int(input("Enter the product ID to␣
↪modify: "))
                            new_price = float(input("Enter the new price: "))
                            self.admin.modify_product(self.catalog,␣
↪product_id, new_price)
```

```
                            print("Product modified successfully.")
                        elif admin_choice == 6:
                            self.display_catalog()
                            product_id = int(input("Enter the product ID to␣
 ↪remove: "))
                            self.catalog = self.admin.remove_product(self.
 ↪catalog, product_id)
                            print("Product removed successfully.")
                        elif admin_choice == 7:
                            break
                        else:
                            print("Invalid choice.")

                else:
                    print("Invalid admin credentials.")

            elif choice == 3:
                print("Exiting the application.")
                break

            else:
                print("Invalid choice.")



if __name__ == "__main__":
    App = AmazonasStore()
    App.main()
```

```
Welcome to Amazonas Store

1. User Login
2. Admin Login
3. Exit
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-14-27664d75faf4> in <cell line: 172>()
    172 if __name__ == "__main__":
    173     App = AmazonasStore()
--> 174     App.main()
    175
    176

<ipython-input-14-27664d75faf4> in main(self)
     67             while True:
     68                 print("\n1. User Login\n2. Admin Login\n3. Exit")
```

```
---> 69                 choice = int(input("Enter your choice: "))
     70
     71                 if choice == 1:

/usr/local/lib/python3.10/dist-packages/ipykernel/kernelbase.py in␣
 ↪raw_input(self, prompt)
    849                 "raw_input was called, but this frontend does not␣
 ↪support input requests."
    850             )
--> 851         return self._input_request(str(prompt),
    852             self._parent_ident,
    853             self._parent_header,

/usr/local/lib/python3.10/dist-packages/ipykernel/kernelbase.py in␣
 ↪_input_request(self, prompt, ident, parent, password)
    893             except KeyboardInterrupt:
    894                 # re-raise KeyboardInterrupt, to truncate traceback
--> 895                 raise KeyboardInterrupt("Interrupted by user") from Non
    896             except Exception as e:
    897                 self.log.warning("Invalid Message:", exc_info=True)

KeyboardInterrupt: Interrupted by user
```