# Project: Developing a E-commerce platform

In [1]:

```
users = [{'username': 'U1', 'password': 'P1'}, {'username': 'U2', 'passwo
categories = {'footwear': [], 'clothing': [], 'Electronics': []}
carts = {}
payments = ['Credit_cards', 'Debit_Card', "Cash", "UPI"]
```

In [2]:

```
print(type(users))
print(type(categories))
print(type(carts))
print(type(payments))
```

```
<class 'list'>
<class 'dict'>
<class 'dict'>
<class 'list'>
```

In [3]:

```
def add_user(username, password):
    # Check if the username already exists, Username will be unique
    for user in users:
        if user['username'] == username:
            print("Username already exists. Please choose a different use
            return False
    # If the username is unique, add the new user
    users.append({'username': username, 'password': password})
    # prefix a string with the letter 'f' to create an fstring. The lette
    print(f"User {username} added successfully.")
    return True
```

In [4]:

```
# User Authentication
def login(username, password):
    for user in users:
        if user['username'] == username and user['password'] == password:
            print("Access Granted")
            return True
    print("Access Denied, Either Incorrect Username or Incorrect Password
    return False
```

In [5]:

```
# Category Management
def add_category(name):
    if name not in categories:
        categories[name] = []
```

In [6]:

```
# Update / Modify Category Name
def update_category(old_name, new_name):
    if old_name in categories:
        categories[new_name] = categories.pop(old_name)
```

In [7]:

```
def add_to_cart(username, category, item):

    user_exists = any(user['username'] == username for user in users)
    if not user_exists:
        print(f"User {username} does not exist.")
        return False

    if username not in carts:
        carts[username] = []
    carts[username].append({'category': category, 'item': item})
    print(f"Item {item} added to {username}'s cart.")
    return True
```

In [8]:

```
def remove_from_cart(username, category, item):
    if username in carts:
        # Filter out the item to be removed based on category and item na
        carts[username] = [cart_item for cart_item in carts[username] if
        print(f"Item {item} removed from {username}'s cart.")
    else:
        print(f"No cart found for user {username}.")
```

In [9]:

```
def process_payment(username, amount, payment_method):
    # Check if the payment method is supported
    if payment_method in payments:
        print(f"Payment of {amount} using {payment_method} for user {user
    else:
        print("Unsupported payment method.")
```

# Program output checking

In [10]:

```
users
```

Out[10]:

```
[{'username': 'U1', 'password': 'P1'}, {'username': 'U2', 'password': 'P
2'}]
```

In [11]:

```
# A new User Comes and enter the Username : U1
add_user("U1","Password")
```

```
Username already exists. Please choose a different username.
```

Out[11]:

```
False
```

In [12]:

```
users
```

Out[12]:

```
[{'username': 'U1', 'password': 'P1'}, {'username': 'U2', 'password': 'P
2'}]
```

In [13]:

```
add_user("Georgy","123")
```

```
User Georgy added successfully.
```

Out[13]:

```
True
```

In [14]:

```
users
```

Out[14]:

```
[{'username': 'U1', 'password': 'P1'},
 {'username': 'U2', 'password': 'P2'},
 {'username': 'Georgy', 'password': '123'}]
```

In [15]:
```
# Now checking for the login of a new user: Authentication

login("Georgy", "123")

# This function will return me True when the user exists, and the credent
```
```
Access Granted
```
Out[15]:
```
True
```

In [16]:
```
# If not then, it will throw the error

login("Georgy", "1234")
```
```
Access Denied, Either Incorrect Username or Incorrect Password or Both
```
Out[16]:
```
False
```

In [17]:
```
# Checking existing category

categories
```
Out[17]:
```
{'footwear': [], 'clothing': [], 'Electronics': []}
```

In [18]:
```
# Adding new category : Kitchen appliances

# add_category("")

add_category("Groceries")
add_category("Toys") # Babies
```

In [19]:
```
categories
```
Out[19]:
```
{'footwear': [],
 'clothing': [],
 'Electronics': [],
 'Groceries': [],
 'Toys': []}
```

In [20]:
```
# Changing Category name

update_category("Toys", "Baby toys")
```

In [21]:
```
categories
```
Out[21]:
```
{'footwear': [],
 'clothing': [],
 'Electronics': [],
 'Groceries': [],
 'Baby toys': []}
```

In [22]:
```
add_to_cart("Georgy", "clothing", "Hoodie")
add_to_cart("U1", "Baby toys", "Building blocks")
```
```
Item Hoodie added to Georgy's cart.
Item Building blocks added to U1's cart.
```
Out[22]:
```
True
```

In [23]:
```
carts
```

```
Out[23]: {'Georgy': [{'category': 'clothing', 'item': 'Hoodie'}],
          'U1': [{'category': 'Baby toys', 'item': 'Building blocks'}]}
```

```
In [24]: # Now removal of some item

         remove_from_cart("U1", "Baby toys", "Building blocks")
```

```
Item Building blocks removed from U1's cart.
```

```
In [25]: carts
```

```
Out[25]: {'Georgy': [{'category': 'clothing', 'item': 'Hoodie'}], 'U1': []}
```

```
In [26]: # Process the payment.
         process_payment("Georgy", 1000, "Phone Pe")

         # Paytm is not a valid payment method in our database
```

```
Unsupported payment method.
```

```
In [27]: process_payment("GEORGY", 1000, "UPI")
```

```
Payment of 1000 using UPI for user GEORGY processed successfully. Expect
the Order within today
```