



**Submitted by- Khushi Kashyap**

**Section- C**

**Course-Computer Science**

**Fundamentals**

**and Career Pathways**

**Assignment Title-Design And Stimulate**

**a Real-World Process Using Flowcharts**

**And Pseudocode.**

**Problem Covered-Online Shopping**

**checkout**

# **Problem Analysis-**

## **1. Abstraction**

- Abstraction is a fundamental concept that involves reducing complexity by focusing on the main idea and ignoring specific details.
- Show only key details like Price , Delivery Address and Payment Options.
- User Addresses and payment details are saved for future checkouts.
- Users see direct message in error rather than Technical error Code.

## **2. Decomposition**

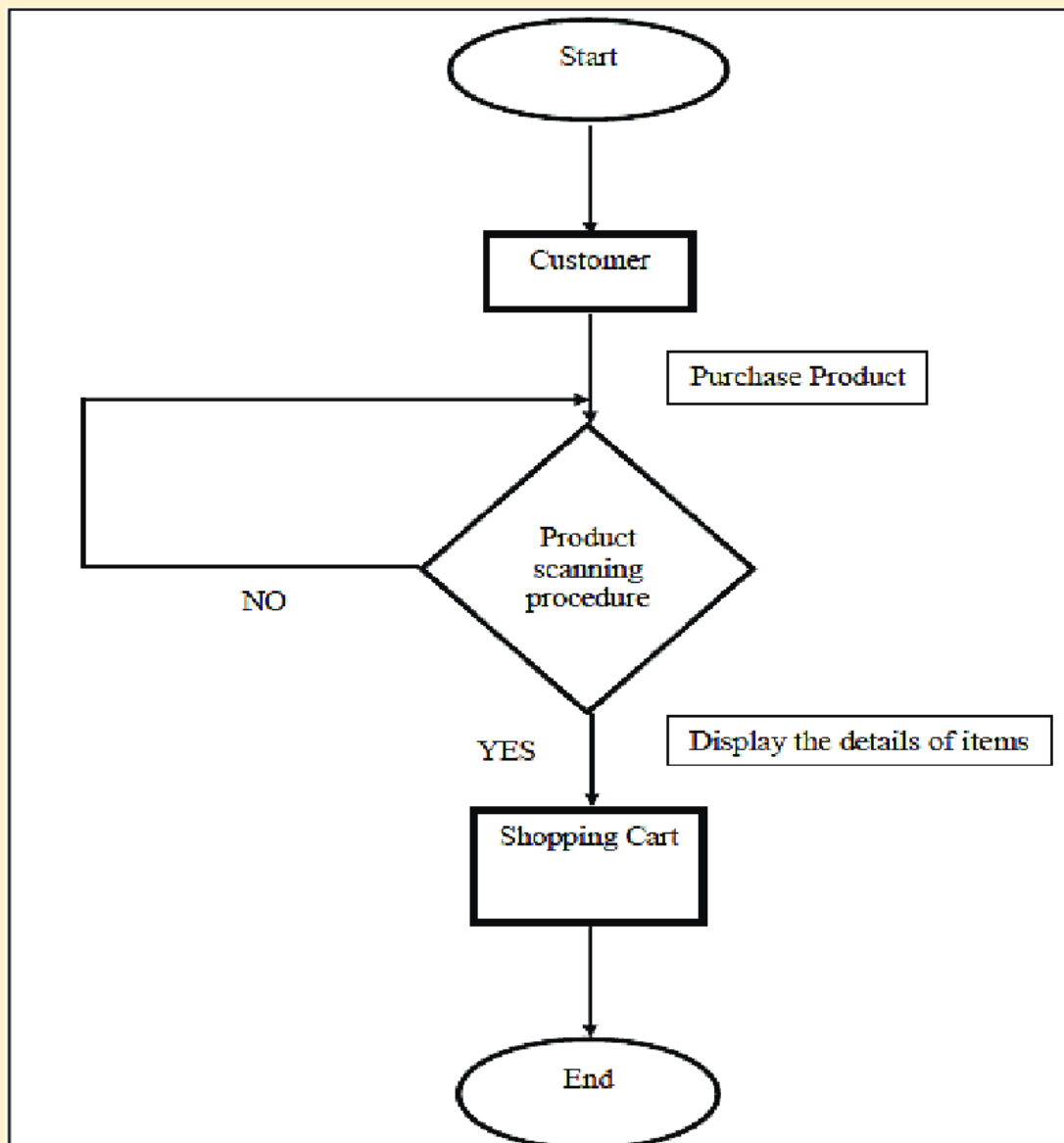
- It is the Process of Breaking down complex problems into smaller and manageable parts.
- Break entire checkout process into smaller parts like cart review ,secure payment ,order confirmation process.
- Break complex Backend processes into data validation , payment gateway connection etc.
- Manage different type of errors like invalid payment ,out of stock item ,etc.

## **3. Pattern Recognition**

- It is the Process of identifying patterns , trends ,regularities in a data.
- Observe preferred payment method ,deliver address.
- Recognize Purchase Patterns.
- Identify repeated user errors.

# Solution Design-

## 1.FLOWCHART



## 2. Pseudocode

- **START**
- **Add items to cart**
- **Display order summary**
- **Select payment method**

**IF payment\_method == 1**

**PRINT "Enter credit card details"**

**ELIF payment\_method == 2**

**PRINT "Enter debit card details"**

**ELIF payment\_method == 3**

**PRINT "Enter UPI ID"**

**ELSE**

**PRINT "Invalid payment method selected"**

- **Confirm payment**
- **END**

# Implementation-

Payment Method to python code

```
pay=input("Enter the payment method:")
```

```
if(pay=="credit card"):
```

```
    print("enter credit card details")
```

```
elif(pay=="debit card"):
```

```
    print("enter debit card detail")
```

```
elif(pay=="UPI"):
```

```
    print("Enter UPI id")
```

```
else:
```

```
    print("Invalid payment method")
```

Done in VS Code

Output- PS C:\Users\kcs\Desktop\python> &  
C:/Users/kcs/AppData/Local/Programs/Python/Python313/python.exe

c:/Users/kcs/Desktop/python/python/proble.py

Enter the payment method:UPI

Enter UPI id

PS C:\Users\kcs\Desktop\python>

# **Report Compilation-**

The development process confirmed that the complexity of the checkout lies not in the sheer volume of steps, but in the criticality of each step's outcome. The greatest challenge is handling error particularly payment failure, which requires a non-linear path (retrying) instead of a simple progression.

The insight gained is that security and validation are paramount, necessitating robust data validation patterns and strict separation of the secure Payment Module (handling sensitive data) from the rest of the application. The modular Decomposition approach proved vital in ensuring that changes to shipping fees, taxes, or discounts would only require modifications within the dedicated Calculation Module, enhancing the system's maintainability.