**PYTHON ASSIGNMENT -1**

**Techshop**

1. Customer Registration Description: When a new customer registers on the TechShop website, their information (e.g., name, email, phone) needs to be stored in the database.

Task: Implement a registration form and database connectivity to insert new customer records. Ensure proper data validation and error handling for duplicate email addresses.

2: Product Catalog Management Description: TechShop regularly updates its product catalog with new items and changes in product details (e.g., price, description). These changes need to be reflected in the database.

Task: Create an interface to manage the product catalog. Implement database connectivity to update product information. Handle changes in product details and ensure data consistency.

3: Placing Customer Orders Description: Customers browse the product catalog and place orders for products they want to purchase. The orders need to be stored in the database.

Task: Implement an order processing system. Use database connectivity to record customer orders, update product quantities in inventory, and calculate order totals.

4: Tracking Order Status Description: Customers and employees need to track the status of their orders. The order status information is stored in the database.

Task: Develop a feature that allows users to view the status of their orders. Implement database connectivity to retrieve and display order status information.

5: Inventory Management Description: TechShop needs to manage product inventory, including adding new products, updating stock levels, and removing discontinued items.

Task: Create an inventory management system with database connectivity. Implement features for adding new products, updating quantities, and handling discontinued products

6: Customer Account Updates Description: Customers may need to update their account information, such as changing their email address or phone number.

Task: Implement a user profile management feature with database connectivity to allow customers to update their account details. Ensure data validation and integrity.

7: Payment Processing Description: When customers make payments for their orders, the payment details (e.g., payment method, amount) must be recorded in the database.

Task: Develop a payment processing system that interacts with the database to record payment transactions, validate payment information, and handle errors.

8: Product Search and Recommendations Description: Customers should be able to search for products based on various criteria (e.g., name, category) and receive product recommendations.

Task: Implement a product search and recommendation engine that uses database connectivity to retrieve relevant product information

**CODE:**

# create database connection

import mysql.connector as con

connect = con.connect(host="localhost", user="root", password="root", database="techshop")

# customer registration

def customer\_registration():

cusid = input("Enter customer id:")

fname = input("Enter First Name:")

lname = input("Enter Last Name:")

email = input("Enter Email:")

if ("@" not in email) and ("." not in email):

print("Invalid Email...Enter valid email id....")

else:

phno = input("Enter Phone Number:")

if (len(phno)!=10):

print("Invalid Phone number...Enter valid number...")

else:

address = input("Enter Address:")

data = (cusid, fname, lname, email, phno, address)

sql = "insert into customer values(%s,%s,%s,%s,%s,%s);"

c = connect.cursor()

c.execute(sql, data)

connect.commit()

print("Customer registration successful....")

menu()

# product catalog

def product\_catalog\_management():

id = input("Enter Id: ")

Quantity = input("Enter Quantity:")

Price = input("Enter Price: ")

data = (Quantity, id)

data1 = (Price,id)

q1 = "update product set Quantity = %s where Pdt\_id = %s;"

q2 = "update product set price = %s where pdt\_id = %s;"

c = connect.cursor()

c.execute(q1, data)

c.execute(q2, data1)

connect.commit()

print("Product details updated....")

# placing customer orders

def customer\_orders():

q1 = "select sum(total\_amount) from orders; "

c = connect.cursor()

c.execute(q1)

list1 = c.fetchone()

print("Total Amount for Orders made:", list1[0])

# Tracking order status

def tracking\_order\_status():

enter = input(" Want to know Order Status...\n Enter Yes'y' or No'n' ")

if enter.lower() == "y":

q1 = " select \* from order\_details;"

c = connect.cursor()

c.execute(q1)

list = c.fetchall()

for i in list:

print("Order Detail Id:", i[0])

print("Order Id:", i[1])

print("product Id:", i[2])

print("Quantity:", i[3])

print("--------------------------------------")

menu()

else:

menu()

# Inventory Management

def add\_inventory():

InvId = input("Enter Inventory Id: ")

PdtId = input("Enter Product Id: ")

qis = input("Enter Quantity in Stock: ")

lsu = input("Last Stock Update: ")

data = (InvId, PdtId, qis, lsu)

q1 = " insert into inventory values(%s, %s, %s, %s);"

c = connect.cursor()

c.execute(q1, data)

connect.commit()

print("Inventory added successfully....")

# update quantities in inventory

def update\_quantity():

Id = input("Enter Inventory Id:")

Quantity = input("Quantity in stock:")

data = (Quantity, Id)

q1 = "update inventory set Quantity\_in\_stock = %s where Inv\_id = %s;"

c = connect.cursor()

c.execute(q1, data)

connect.commit()

print("....Updation done successfully....")

menu()

# update customer accounts

def update\_customer\_account():

cusid = input("Enter Customer Id:")

Enter = input("Enter Update email 'e',address 'd',phone 'p': ")

if Enter == "e":

email = input("Enter email: ")

if ("@" not in email) and ("." not in email):

print("Invalid Email...Enter valid email id....")

menu()

else:

data = (email, cusid)

q1 = "update customer set email = %s where cus\_id = %s; "

c = connect.cursor()

c.execute(q1, data)

connect.commit()

print(".......Email updated successfully....")

menu()

elif Enter == "a":

address = input("Enter address: ")

data1 = (address, cusid)

q2 = "update customer set address = %s where cus\_id = %s;"

c1 = connect.cursor()

c1.execute(q2, data1)

connect.commit()

print("........Address updated successfully....")

menu()

elif Enter == "p":

phno = input("Enter Phone Number: ")

if (len(phno)!=10):

print("Invalid Phone number...Enter valid number...")

menu()

else:

data2 = (phno, cusid)

q3 = "update customer set phno = %s where cus\_id = %s;"

c2 = connect.cursor()

c2.execute(q3, data2)

connect.commit()

print("..........Phone number updated successfully....")

menu()

#Product Search and Recommendations

def product\_search\_recommendations():

Name = input("Enter Product Name:")

data = (Name.lower(),)

q1 = "select \* from product where pdt\_name = %s;"

c = connect.cursor()

c.execute(q1, data)

list = c.fetchall()

for i in list:

print("---------------------------")

print("Enter Product Id:", i[0])

# print("Enter Product Name:", i[1])

print("Enter Quantity:", i[2])

print("Enter Price:", i[3])

print("Enter Category:", i[4])

print("---------------------------------")

print("..........Product Search Successful.........")

menu()

#payment processing

def payment\_processing():

method = input("Enter Payment Method:")

amount = input("Enter Amount:")

data2 = (method, amount)

sql2 = "insert into payment\_details values(%s, %s);"

c1 = connect.cursor()

c1.execute(sql2, data2)

connect.commit()

print("payment details updated successfully....")

menu()

# creating menu

def menu():

print ("Select an option:")

print("1.Customer Registration")

print("2.Product Catalog management")

print("3.Placing Customer Orders")

print("4.Tracking Order Status")

print("5.Add Inventory ")

print("6.Update Quantities in Inventory")

print("7.Customer Account Updates")

print("8.Product Search and Recommendations ")

print("9.Payment Processing ")

print("10.Exit")

option = input("Enter option: ")

if option == '1':

customer\_registration()

elif option == '2':

product\_catalog\_management()

elif option == '3':

customer\_orders()

elif option == '4':

tracking\_order\_status()

elif option == '5':

add\_inventory()

elif option == '6':

update\_quantity()

elif option == '7':

update\_customer\_account()

elif option == '8':

product\_search\_recommendations()

elif option == '9':

payment\_processing()

elif option == '10':

print("....Exit....")

else:

print("Invalid option...\n Try again...")

menu()

print(".........Welcome to Techshop.........")

menu()

**Output:**



























