**ASSIGNMENT – 4**

**COURIER MANAGEMENT SYSTEM**

Task 1: Control Flow Statements

**CODE:**

# Check Order Delivery Status

status = "Delivered"

if status == "Delivered":

print("The order has been delivered.")

elif status == "Processing":

print("The order is still being processed.")

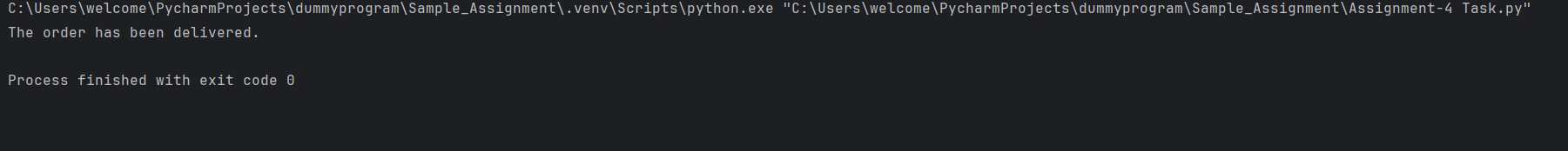
elif status == "Cancelled":

print("The order has been cancelled.")

else:

print("Invalid order status.")

**OUTPUT:**



**CODE:**

# Categorize Parcels Based on Weight

def categorize\_parcel(weight):

weight\_categories = {

0: "Light",

1: "Light",

2: "Light",

3: "Light",

4: "Medium",

5: "Medium",

6: "Medium",

7: "Heavy",

8: "Heavy",

9: "Heavy"

}

category = weight\_categories.get(weight, "Heavy")

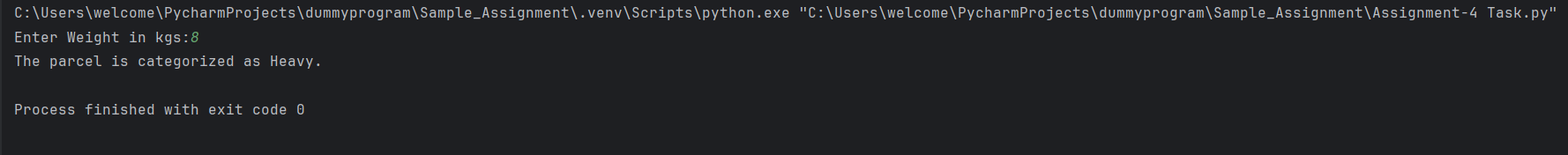
return category

parcel\_weight = int(input("Enter Weight:"))

result = categorize\_parcel(parcel\_weight)

print(f"The parcel is categorized as {result}.")

**OUTPUT:**



**CODE:**

# User Authentication

def authenticate\_user(username, password):

valid\_username = "user123"

valid\_password = "pass456"

if username == valid\_username and password == valid\_password:

return True

else:

return False

entered\_username = input("Enter username: ")

entered\_password = input("Enter password: ")

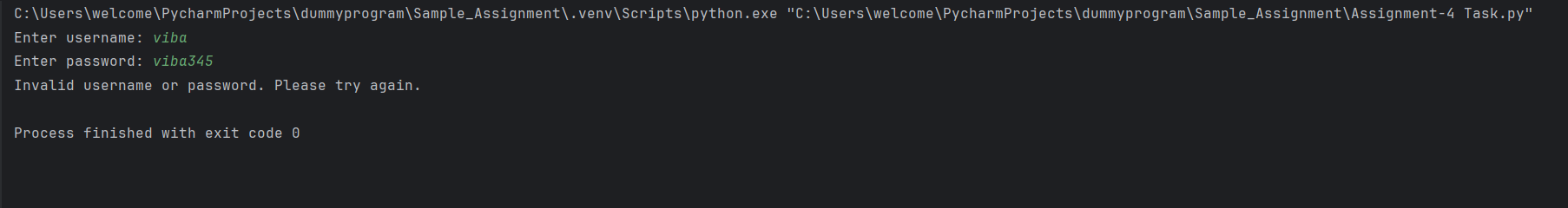
if authenticate\_user(entered\_username, entered\_password):

print("Login successful!")

else:

print("Invalid username or password. Please try again.")

**OUTPUT:**



**CODE:**

# Courier assignment Logic

def assign\_courier(proximity, load\_capacity):

while proximity > 0 and load\_capacity > 0:

if 0 < proximity < 10 and load\_capacity > 5:

print("Assigning courier...")

else:

print("No available courier meeting the criteria.")

proximity = float(input("Enter shipment proximity (in km): "))

load\_capacity = float(input("Enter courier load capacity (in kg): "))

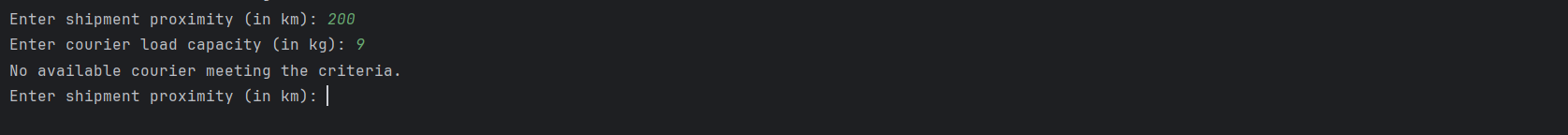
shipment\_proximity = float(input("Enter shipment proximity (in km): "))

courier\_load\_capacity = float(input("Enter courier load capacity (in kg): "))

assign\_courier(shipment\_proximity, courier\_load\_capacity)

print("Exiting courier assignment program.")

**OUTPUT:**



**TASK:2**

**CODE:**

#Display All Orders for a Specific Customer

def display\_customer\_orders(customer\_id, orders):

for order in orders:

if order["customer\_id"] == customer\_id:

print(f"Order ID: {order['order\_id']}, Status: {order['status']}")

customer\_id = 123

sample\_orders = [

{"order\_id": 1, "customer\_id": 123, "status": "Delivered"},

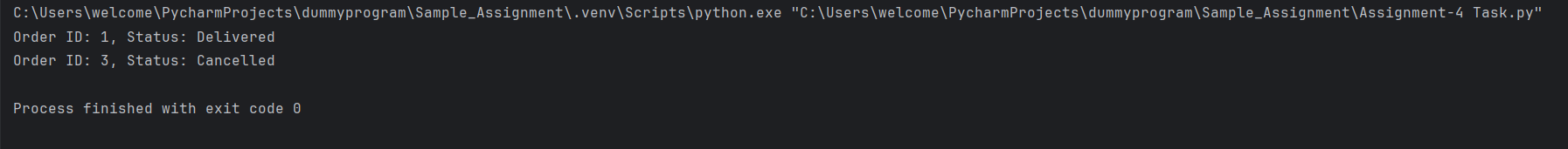
{"order\_id": 2, "customer\_id": 456, "status": "Processing"},

{"order\_id": 3, "customer\_id": 123, "status": "Cancelled"},

]

display\_customer\_orders(customer\_id, sample\_orders)

**OUTPUT:**



**CODE:**

# Track Real-Time Location of a Courier

def track\_courier\_location(destination, current\_location):

while current\_location != destination:

print(f"Courier is currently at {current\_location}.")

current\_location += 1

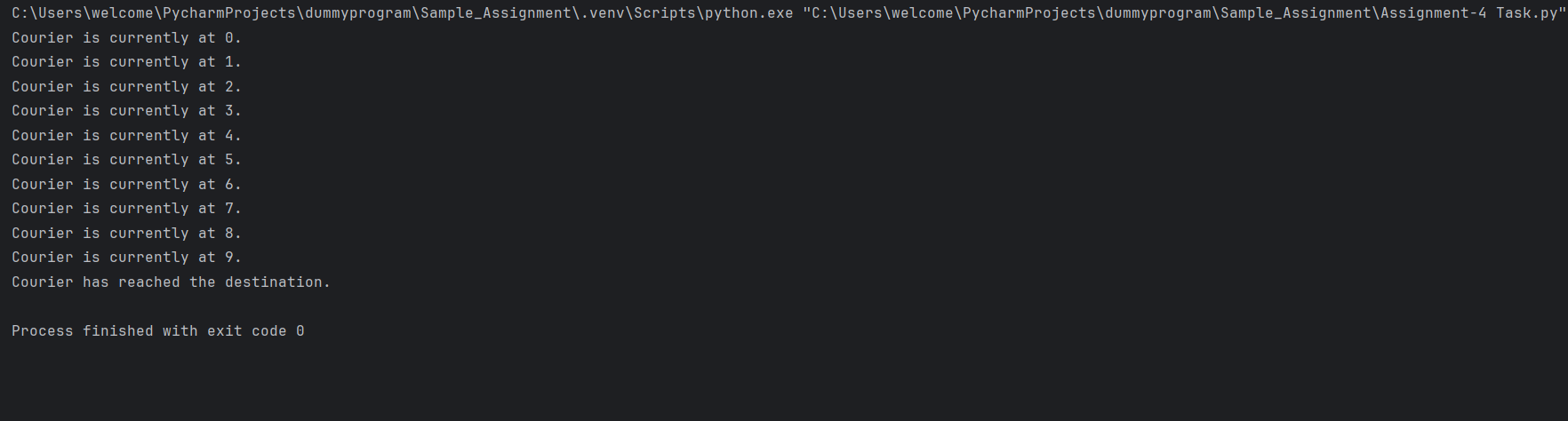
print("Courier has reached the destination.")

courier\_destination = 10

courier\_current\_location = 0

track\_courier\_location(courier\_destination, courier\_current\_location)

**OUTPUT:**



**TASK 3**

**CODE:**

# Store Parcel Tracking History

def store\_parcel\_tracking(parcel\_id, location, tracking\_history):

entry = {"parcel\_id": parcel\_id, "location": location}

tracking\_history.append(entry)

parcel\_tracking\_history = []

store\_parcel\_tracking(1, "Warehouse A", parcel\_tracking\_history)

store\_parcel\_tracking(1, "Distribution Center", parcel\_tracking\_history)

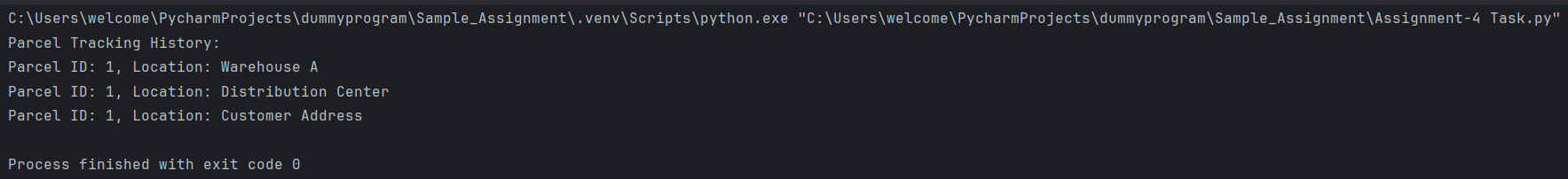
store\_parcel\_tracking(1, "Customer Address", parcel\_tracking\_history)

print("Parcel Tracking History:")

for entry in parcel\_tracking\_history:

print(f"Parcel ID: {entry['parcel\_id']}, Location: {entry['location']}")

**OUTPUT:**



**CODE**

def find\_nearest\_courier(customer\_location, courier\_locations):

nearest\_courier = None

min\_distance = float('inf')

for courier\_location in courier\_locations:

distance = abs(customer\_location - courier\_location)

if distance < min\_distance:

min\_distance = distance

nearest\_courier = courier\_location

return nearest\_courier

customer\_address = 5

courier\_locations = [8, 3, 1, 4]

nearest\_courier = find\_nearest\_courier(customer\_address, courier\_locations)

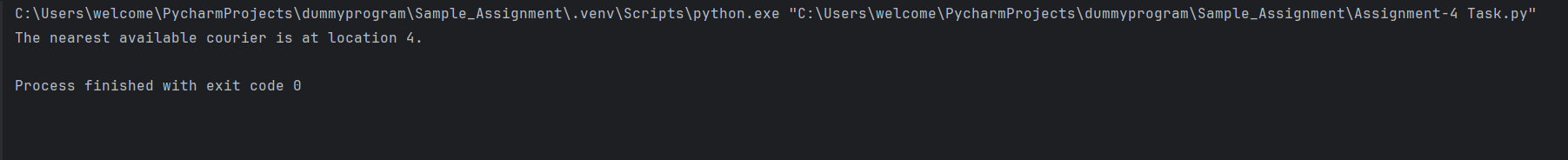
if nearest\_courier is not None:

print(f"The nearest available courier is at location {nearest\_courier}.")

else:

print("No available couriers.")

**OUTPUT**



**COURIER INFORMATION SYSTEM**

**CODE:**

# create database connection

import mysql.connector as con

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

# Get user details

def user():

id = input("Enter User Id: ")

data = (id,)

sql = " select \* from user where userid = %s; "

c = connect.cursor()

c.execute(sql, data)

list = c.fetchall()

for i in list:

print("Name: ", i[1])

print("Email: ", i[2])

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

print("Contact: ", i[4])

print("Address: ", i[5])

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

menu()

# updating courier status

def update\_courier\_status():

Id = input("Enter Courier Id:")

Status = input("Courier status:")

data = (Status, Id)

q1 = "update courier set Status = %s where courier\_id = %s;"

c = connect.cursor()

c.execute(q1, data)

connect.commit()

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

menu()

# revenue report

def revenue\_report():

payment\_id = input("Enter Payment Id: ")

data = (payment\_id,)

sql = " select \* from payment where Payment\_ID = %s; "

c = connect.cursor()

c.execute(sql, data)

list = c.fetchall()

for i in list:

print("Courier Id is: ", i[1])

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

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

print("Payment Date: ", i[4])

print("Payment Date: ", i[5])

print("Employee Id:", i[6])

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

menu()

# shippment status report

def shippment\_status\_report():

package\_id = input("Enter Package Id: ")

data = (package\_id,)

sql = " select \* from packages where Package\_ID = %s; "

c = connect.cursor()

c.execute(sql, data)

list = c.fetchall()

for i in list:

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

print("Package\_Details: ", i[2])

print("Shipping\_Status: ", i[3])

print("Courier Id :", i[4])

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

menu()

# courier services

def courier\_services():

id = input("Enter Service Id: ")

data = (id,)

sql = " select \* from courier\_services where service\_id = %s; "

c = connect.cursor()

c.execute(sql, data)

list = c.fetchall()

for i in list:

print("Service Name : ", i[1])

print("Cost: ", i[2])

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

menu()

def menu():

print("Select an option:")

print("1.Shipment Status Report")

print("2.Update Courier Status")

print("3.Revenue Report")

print("4.Get Courier Services")

print("5.Get User Information")

print("6.Exit")

option = input("Enter option: ")

if option == '1':

shippment\_status\_report()

elif option == '2':

update\_courier\_status()

elif option == '3':

revenue\_report()

elif option == '4':

courier\_services()

elif option == '5':

user()

elif option == '6':

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

else:

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

menu()

menu()

OUTPUT:



