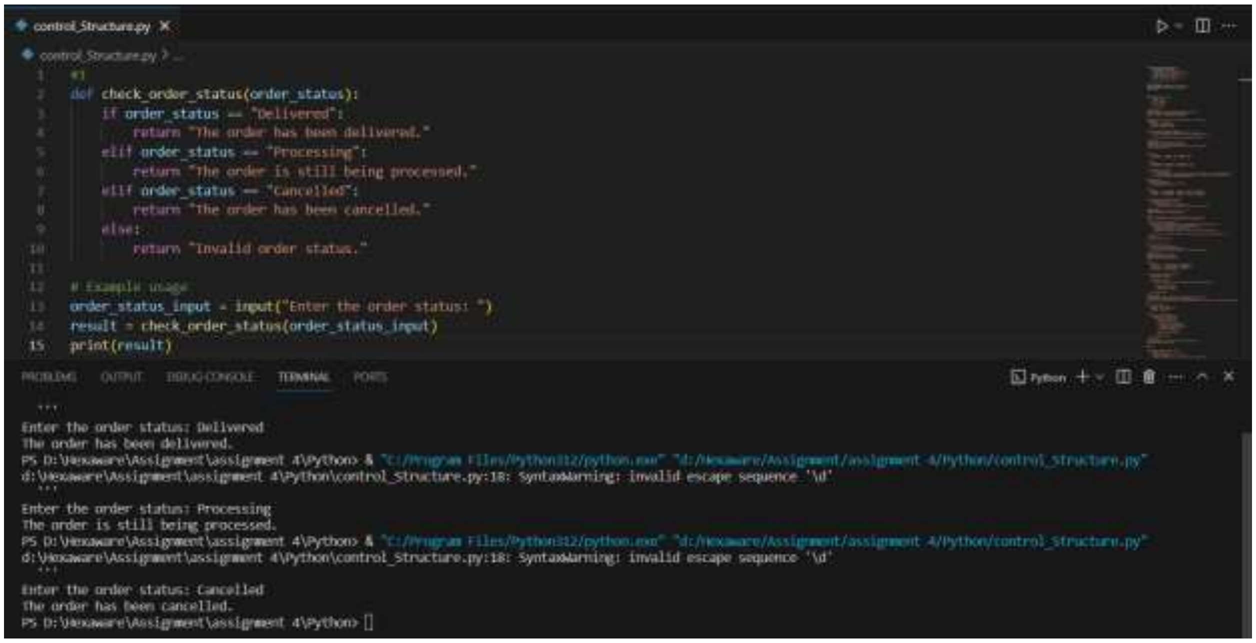
Assignment 4

Courier Management System

Control Structure(Task1,2,3,4)(With Outputs):

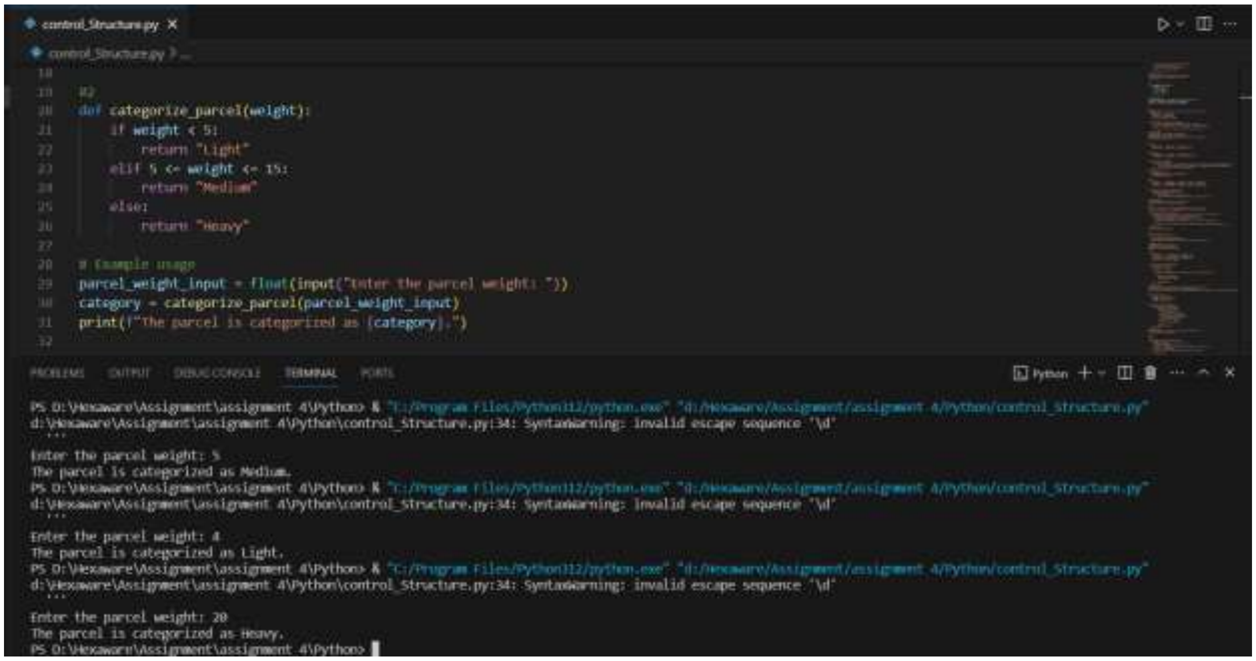
1.



```python
#1
def check_order_status(order_status):
    if order_status == "Delivered":
        return "The order has been delivered."
    elif order_status == "Processing":
        return "The order is still being processed."
    elif order_status == "Cancelled":
        return "The order has been cancelled."
    else:
        return "Invalid order status."

# Example usage
order_status_input = input("Enter the order status: ")
result = check_order_status(order_status_input)
print(result)
```

```
  '''
Enter the order status: Delivered
The order has been delivered.
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
d:\Hexaware\Assignment\assignment 4\Python\control_Structure.py:18: SyntaxWarning: invalid escape sequence '\d'
  '''
Enter the order status: Processing
The order is still being processed.
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
d:\Hexaware\Assignment\assignment 4\Python\control_Structure.py:18: SyntaxWarning: invalid escape sequence '\d'
  '''
Enter the order status: Cancelled
The order has been cancelled.
PS D:\Hexaware\Assignment\assignment 4\Python> []
```

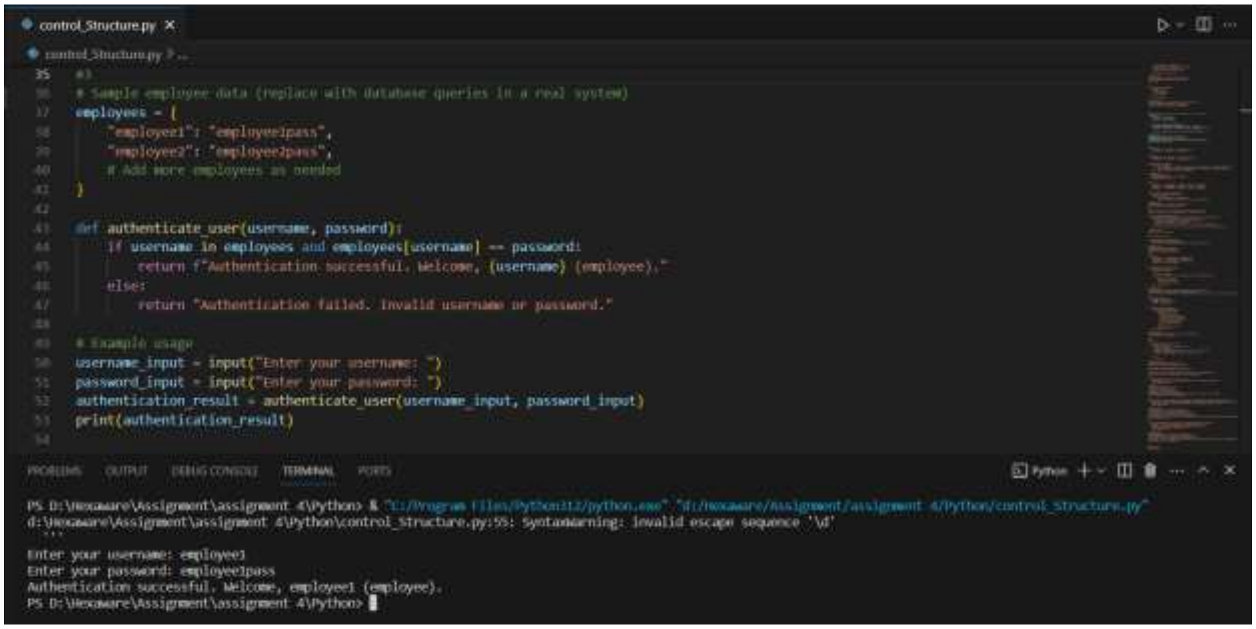
2.



```python
#2
def categorize_parcel(weight):
    if weight < 5:
        return "Light"
    elif 5 <= weight <= 15:
        return "Medium"
    else:
        return "Heavy"

# Example usage
parcel_weight_input = float(input("Enter the parcel weight: "))
category = categorize_parcel(parcel_weight_input)
print(f"The parcel is categorized as {category}.")
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
d:\Hexaware\Assignment\assignment 4\Python\control_Structure.py:34: SyntaxWarning: invalid escape sequence '\d'
  '''
Enter the parcel weight: 5
The parcel is categorized as Medium.
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
d:\Hexaware\Assignment\assignment 4\Python\control_Structure.py:34: SyntaxWarning: invalid escape sequence '\d'
  '''
Enter the parcel weight: 4
The parcel is categorized as Light.
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
d:\Hexaware\Assignment\assignment 4\Python\control_Structure.py:34: SyntaxWarning: invalid escape sequence '\d'
  '''
Enter the parcel weight: 20
The parcel is categorized as Heavy.
PS D:\Hexaware\Assignment\assignment 4\Python> ▌
```

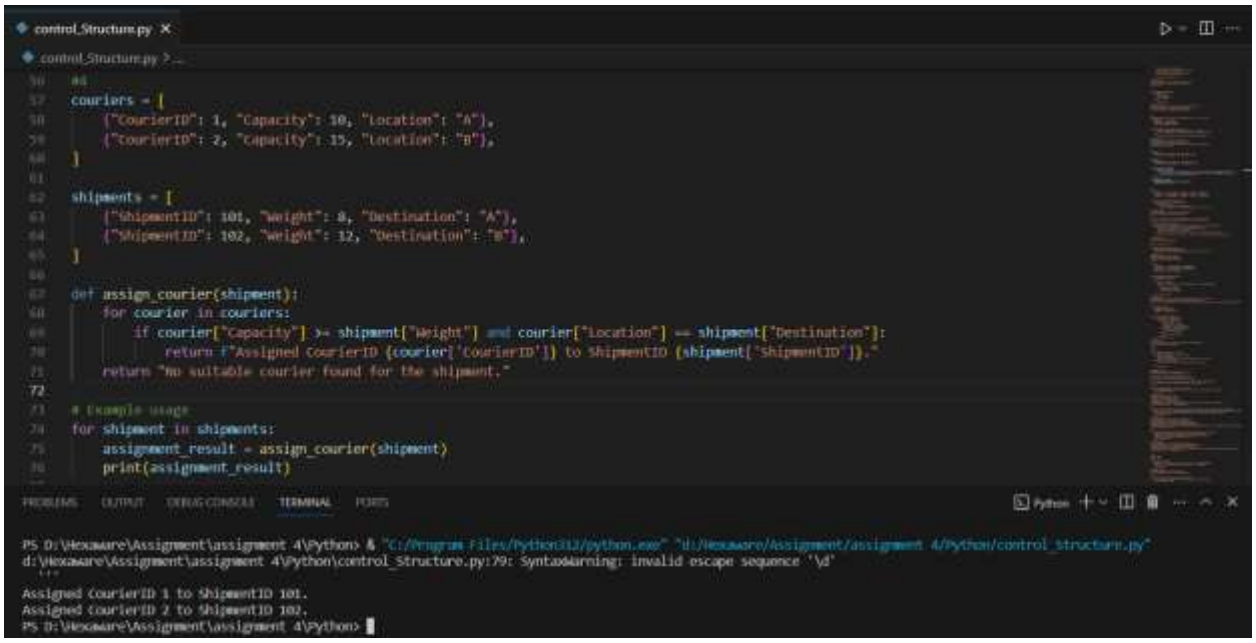
3.



```python
#3
# Sample employee data (replace with database queries in a real system)
employees = {
    "employee1": "employee1pass",
    "employee2": "employee2pass",
    # Add more employees as needed
}

def authenticate_user(username, password):
    if username in employees and employees[username] == password:
        return f"Authentication successful. Welcome, {username} (employee)."
    else:
        return "Authentication failed. Invalid username or password."

# Example usage
username_input = input("Enter your username: ")
password_input = input("Enter your password: ")
authentication_result = authenticate_user(username_input, password_input)
print(authentication_result)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
d:\Hexaware\Assignment\assignment 4\Python\control_Structure.py:55: SyntaxWarning: invalid escape sequence '\d'
  '''
Enter your username: employee1
Enter your password: employee1pass
Authentication successful. Welcome, employee1 (employee).
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

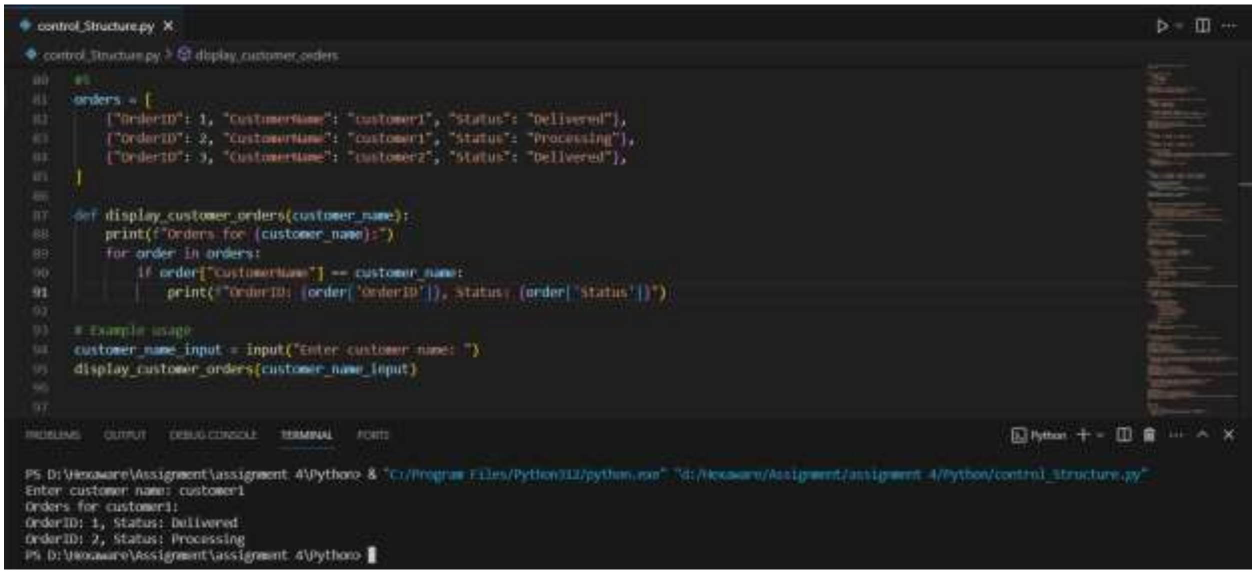
4.



```python
#4
couriers = [
    {"CourierID": 1, "Capacity": 10, "Location": "A"},
    {"CourierID": 2, "Capacity": 15, "Location": "B"},
]

shipments = [
    {"ShipmentID": 101, "Weight": 8, "Destination": "A"},
    {"ShipmentID": 102, "Weight": 12, "Destination": "B"},
]

def assign_courier(shipment):
    for courier in couriers:
        if courier["Capacity"] >= shipment["Weight"] and courier["Location"] == shipment["Destination"]:
            return f"Assigned CourierID {courier['CourierID']} to ShipmentID {shipment['ShipmentID']}."
    return "No suitable courier found for the shipment."

# Example usage
for shipment in shipments:
    assignment_result = assign_courier(shipment)
    print(assignment_result)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
d:\Hexaware\Assignment\assignment 4\Python\control_Structure.py:79: SyntaxWarning: invalid escape sequence '\d'
  '''
Assigned CourierID 1 to ShipmentID 101.
Assigned CourierID 2 to ShipmentID 102.
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

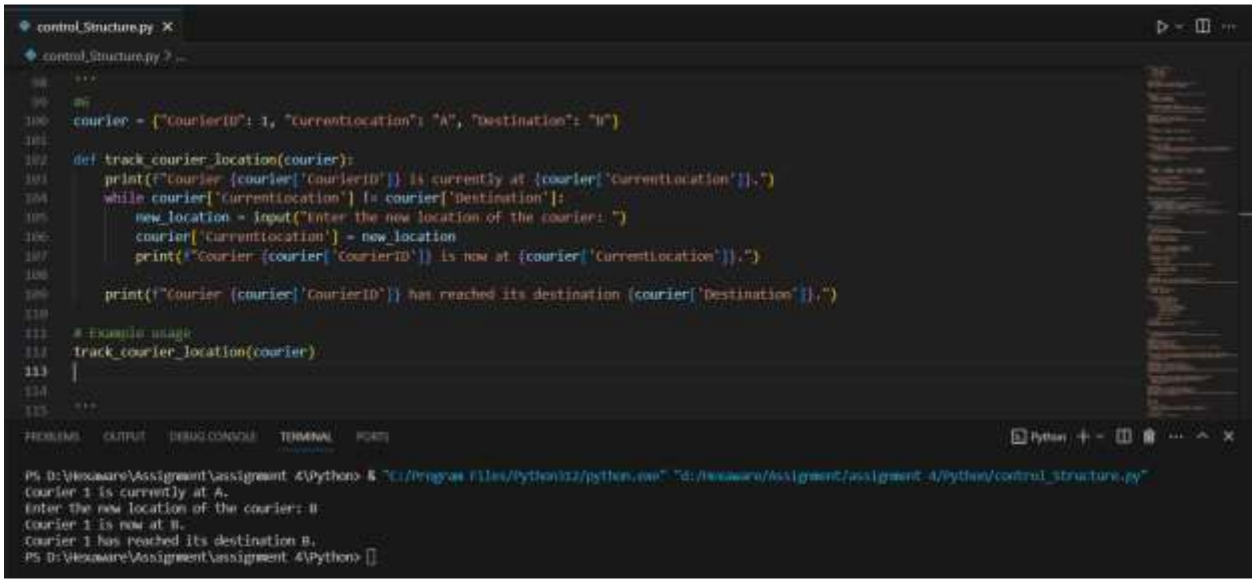
5.



```python
#5
orders = [
    {"OrderID": 1, "CustomerName": "customer1", "Status": "Delivered"},
    {"OrderID": 2, "CustomerName": "customer1", "Status": "Processing"},
    {"OrderID": 3, "CustomerName": "customer2", "Status": "Delivered"},
]

def display_customer_orders(customer_name):
    print(f"Orders for {customer_name}:")
    for order in orders:
        if order["CustomerName"] == customer_name:
            print(f"OrderID: {order['OrderID']}, Status: {order['Status']}")

# Example usage
customer_name_input = input("Enter customer name: ")
display_customer_orders(customer_name_input)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Enter customer name: customer1
Orders for customer1:
OrderID: 1, Status: Delivered
OrderID: 2, Status: Processing
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

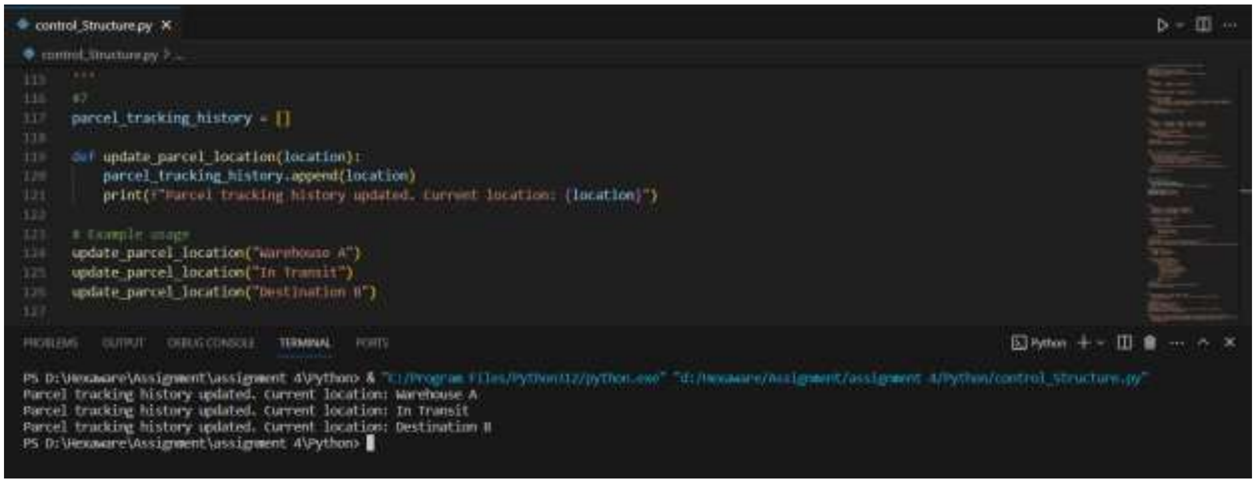
6.



```python
'''
#6
courier = {"CourierID": 1, "CurrentLocation": "A", "Destination": "B"}

def track_courier_location(courier):
    print(f"Courier {courier['CourierID']} is currently at {courier['CurrentLocation']}.")
    while courier['CurrentLocation'] != courier['Destination']:
        new_location = input("Enter the new location of the courier: ")
        courier['CurrentLocation'] = new_location
        print(f"Courier {courier['CourierID']} is now at {courier['CurrentLocation']}.")

    print(f"Courier {courier['CourierID']} has reached its destination {courier['Destination']}.")

# Example usage
track_courier_location(courier)


'''
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Courier 1 is currently at A.
Enter the new location of the courier: B
Courier 1 is now at B.
Courier 1 has reached its destination B.
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

7.

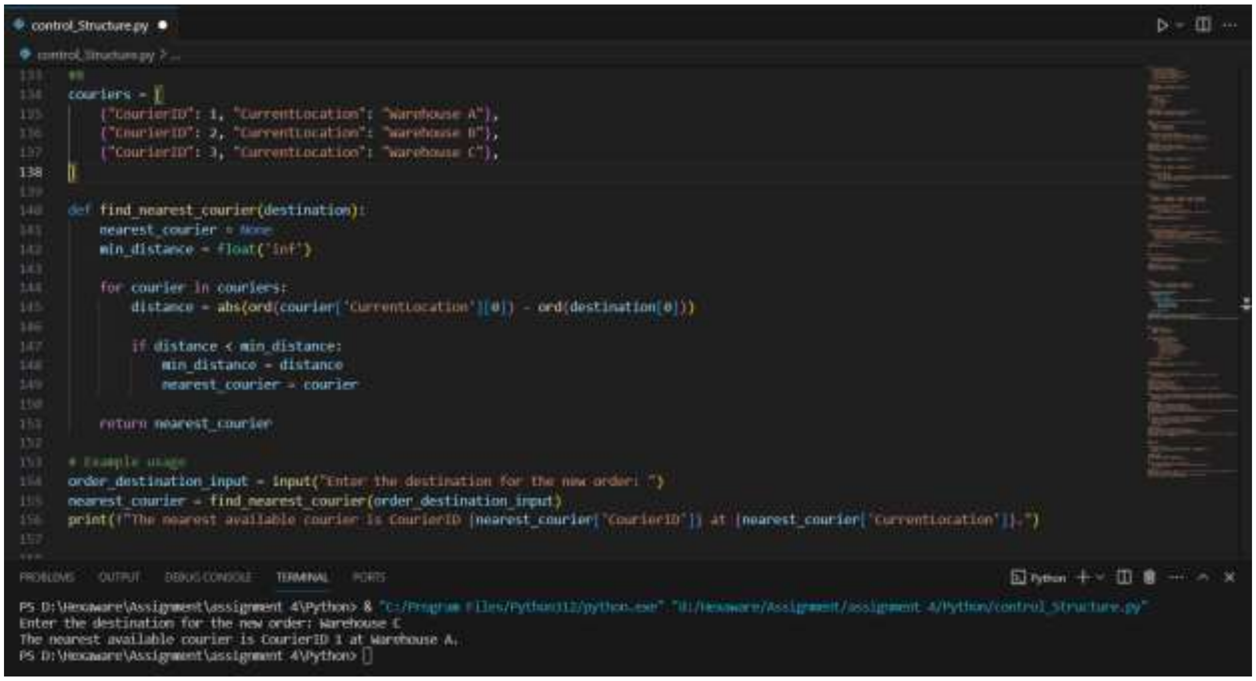


control_Structure.py

```python
'''
#7
parcel_tracking_history = []

def update_parcel_location(location):
    parcel_tracking_history.append(location)
    print(f"Parcel tracking history updated. Current location: {location}")

# Example usage
update_parcel_location("Warehouse A")
update_parcel_location("In Transit")
update_parcel_location("Destination B")
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Parcel tracking history updated. Current location: Warehouse A
Parcel tracking history updated. Current location: In Transit
Parcel tracking history updated. Current location: Destination B
PS D:\Hexaware\Assignment\assignment 4\Python>
```

8.

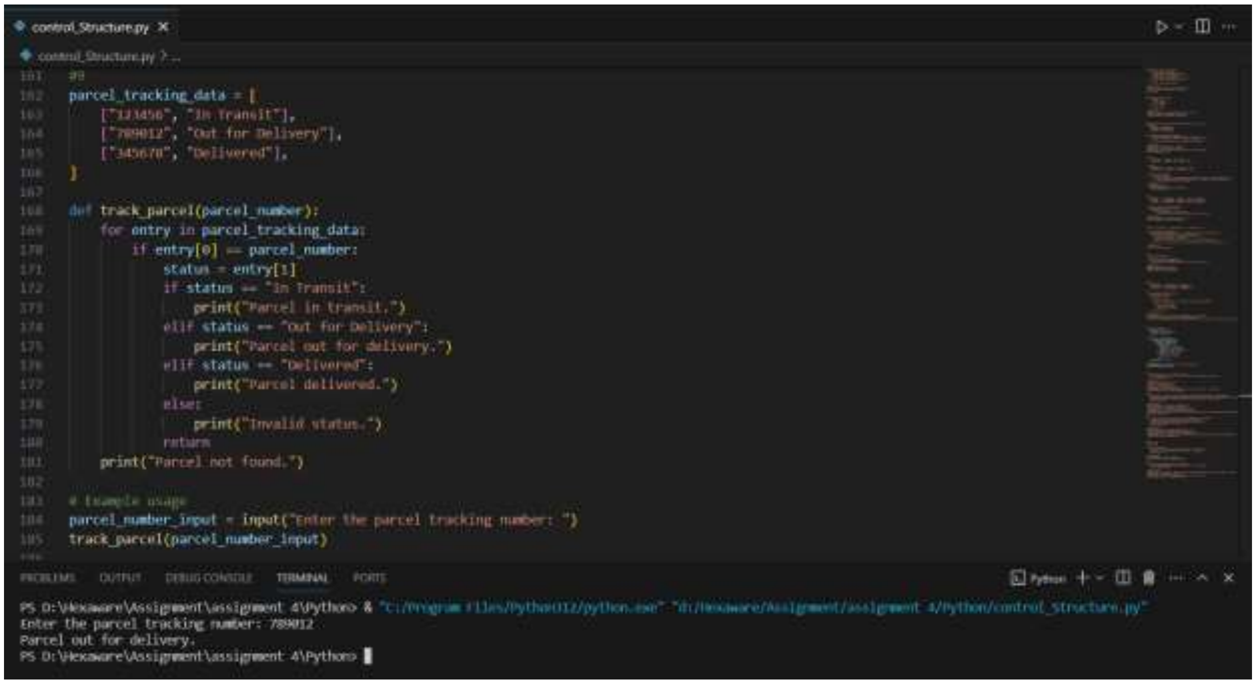


control_Structure.py

```python
#8
couriers = [
    {"CourierID": 1, "CurrentLocation": "Warehouse A"},
    {"CourierID": 2, "CurrentLocation": "Warehouse B"},
    {"CourierID": 3, "CurrentLocation": "Warehouse C"},
]

def find_nearest_courier(destination):
    nearest_courier = None
    min_distance = float('inf')

    for courier in couriers:
        distance = abs(ord(courier['CurrentLocation'][0]) - ord(destination[0]))

        if distance < min_distance:
            min_distance = distance
            nearest_courier = courier

    return nearest_courier

# Example usage
order_destination_input = input("Enter the destination for the new order: ")
nearest_courier = find_nearest_courier(order_destination_input)
print(f"The nearest available courier is CourierID {nearest_courier['CourierID']} at {nearest_courier['CurrentLocation']}.")
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Enter the destination for the new order: Warehouse C
The nearest available courier is CourierID 1 at Warehouse A.
PS D:\Hexaware\Assignment\assignment 4\Python>
```

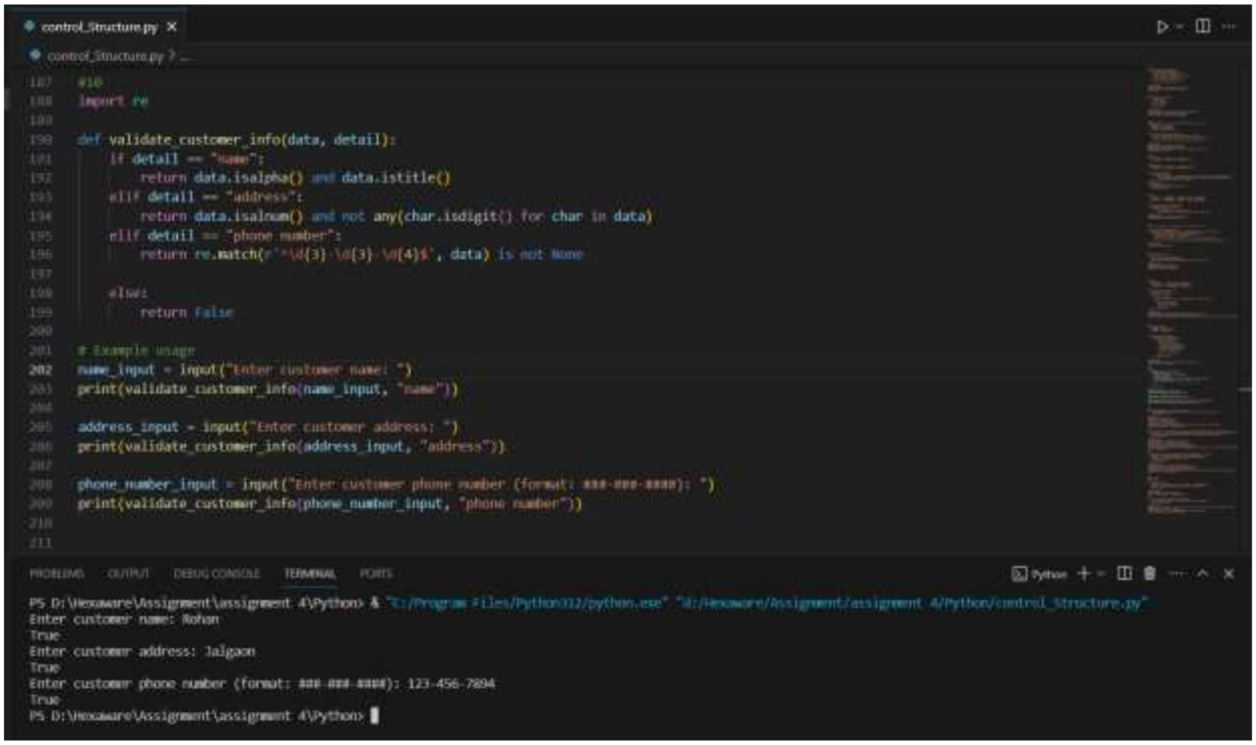
9.



```python
#9
parcel_tracking_data = [
    ["123456", "In Transit"],
    ["789012", "Out for Delivery"],
    ["345678", "Delivered"],
]

def track_parcel(parcel_number):
    for entry in parcel_tracking_data:
        if entry[0] == parcel_number:
            status = entry[1]
            if status == "In Transit":
                print("Parcel in transit.")
            elif status == "Out for Delivery":
                print("Parcel out for delivery.")
            elif status == "Delivered":
                print("Parcel delivered.")
            else:
                print("Invalid status.")
            return
    print("Parcel not found.")

# Example usage
parcel_number_input = input("Enter the parcel tracking number: ")
track_parcel(parcel_number_input)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Enter the parcel tracking number: 789012
Parcel out for delivery.
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

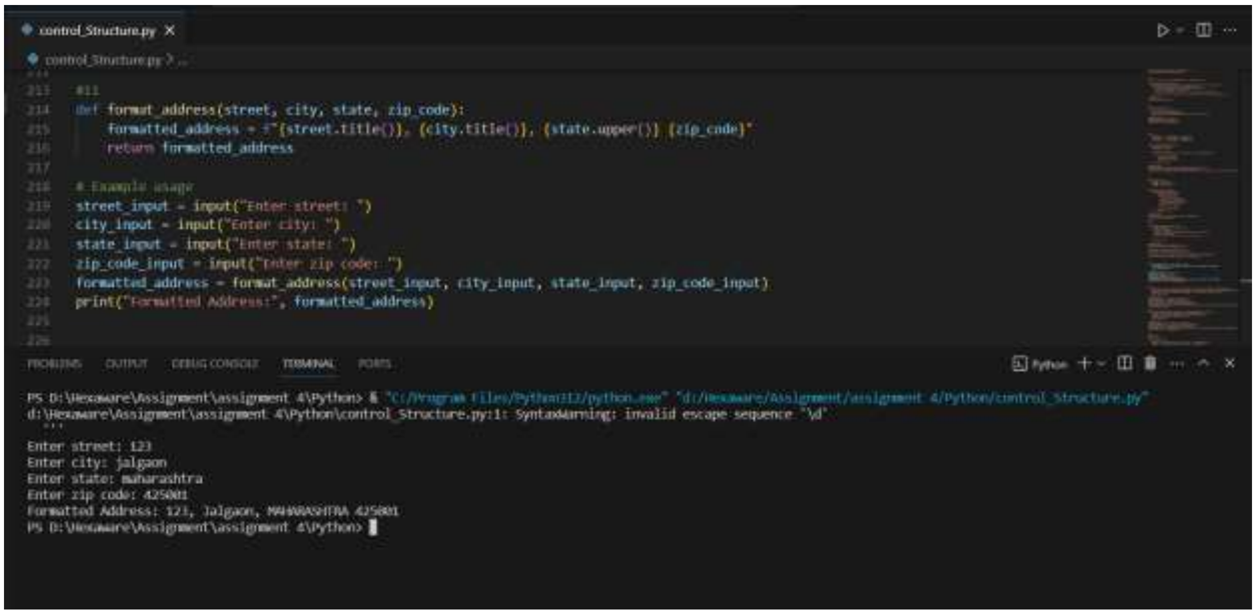
10.



```python
#10
import re

def validate_customer_info(data, detail):
    if detail == "name":
        return data.isalpha() and data.istitle()
    elif detail == "address":
        return data.isalnum() and not any(char.isdigit() for char in data)
    elif detail == "phone number":
        return re.match(r'^\d{3}-\d{3}-\d{4}$', data) is not None

    else:
        return False

# Example usage
name_input = input("Enter customer name: ")
print(validate_customer_info(name_input, "name"))

address_input = input("Enter customer address: ")
print(validate_customer_info(address_input, "address"))

phone_number_input = input("Enter customer phone number (format: ###-###-####): ")
print(validate_customer_info(phone_number_input, "phone number"))
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Enter customer name: Rohan
True
Enter customer address: Jalgaon
True
Enter customer phone number (format: ###-###-####): 123-456-7894
True
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

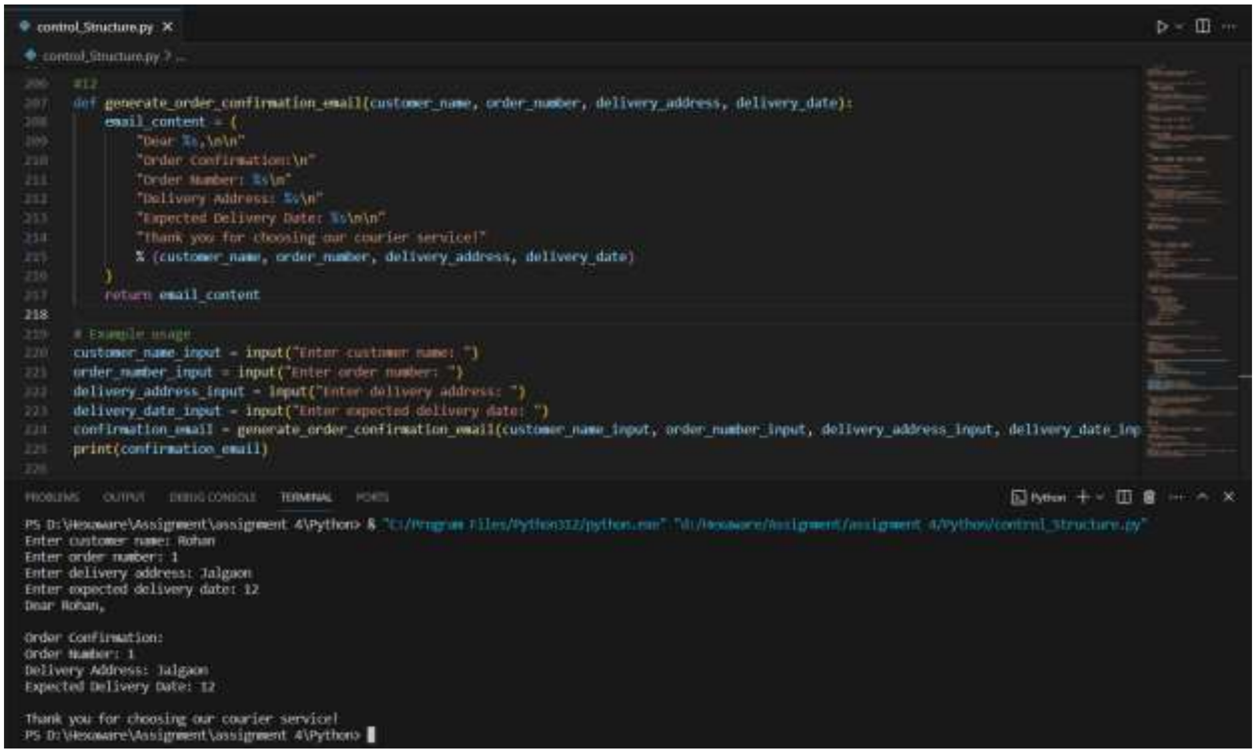
11.



```python
#11
def format_address(street, city, state, zip_code):
    formatted_address = f"{street.title()}, {city.title()}, {state.upper()} {zip_code}"
    return formatted_address

# Example usage
street_input = input("Enter street: ")
city_input = input("Enter city: ")
state_input = input("Enter state: ")
zip_code_input = input("Enter zip code: ")
formatted_address = format_address(street_input, city_input, state_input, zip_code_input)
print("Formatted Address:", formatted_address)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
d:\Hexaware\Assignment\assignment 4\Python\control_Structure.py:1: SyntaxWarning: invalid escape sequence '\d'
  '''
Enter street: 123
Enter city: jalgaon
Enter state: maharashtra
Enter zip code: 425001
Formatted Address: 123, Jalgaon, MAHARASHTRA 425001
PS D:\Hexaware\Assignment\assignment 4\Python>
```

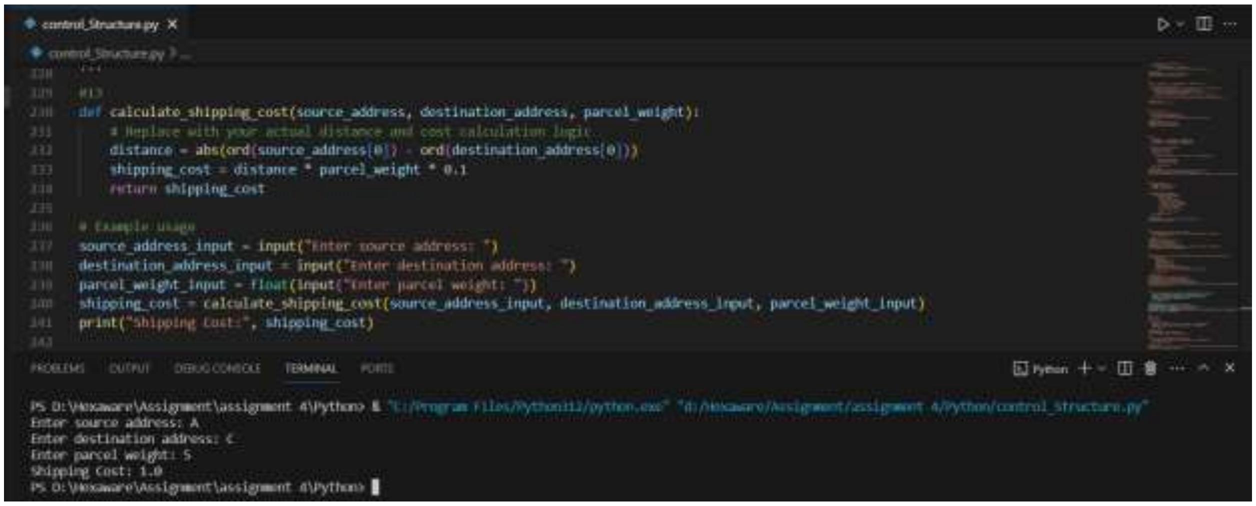
12.



```python
#12
def generate_order_confirmation_email(customer_name, order_number, delivery_address, delivery_date):
    email_content = (
        "Dear %s,\n\n"
        "Order Confirmation:\n"
        "Order Number: %s\n"
        "Delivery Address: %s\n"
        "Expected Delivery Date: %s\n\n"
        "Thank you for choosing our courier service!"
        % (customer_name, order_number, delivery_address, delivery_date)
    )
    return email_content

# Example usage
customer_name_input = input("Enter customer name: ")
order_number_input = input("Enter order number: ")
delivery_address_input = input("Enter delivery address: ")
delivery_date_input = input("Enter expected delivery date: ")
confirmation_email = generate_order_confirmation_email(customer_name_input, order_number_input, delivery_address_input, delivery_date_inp
print(confirmation_email)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Enter customer name: Rohan
Enter order number: 1
Enter delivery address: Jalgaon
Enter expected delivery date: 12
Dear Rohan,

Order Confirmation:
Order Number: 1
Delivery Address: Jalgaon
Expected Delivery Date: 12

Thank you for choosing our courier service!
PS D:\Hexaware\Assignment\assignment 4\Python>
```

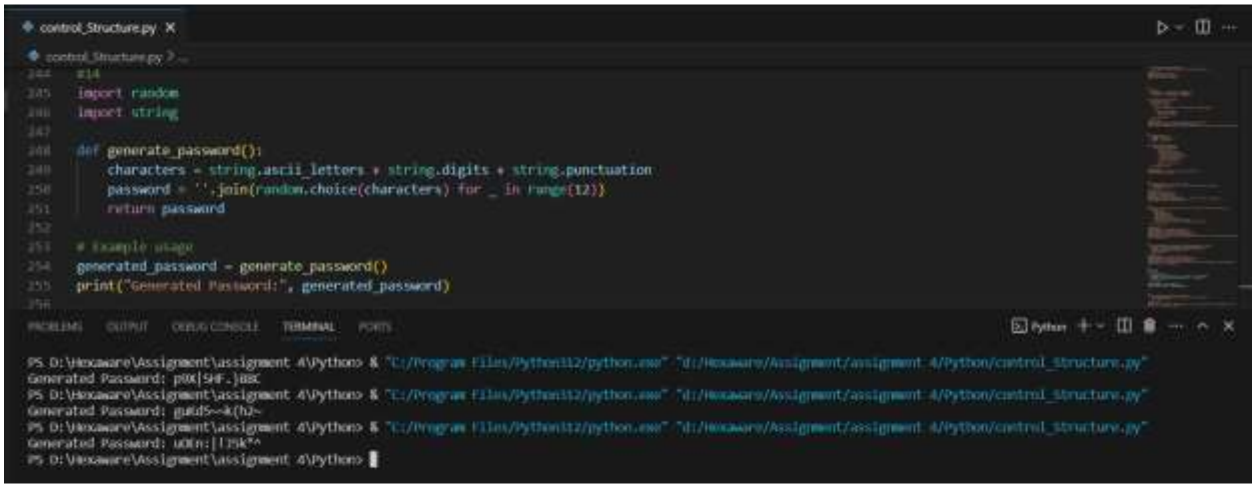
13.



```
'''
#13
def calculate_shipping_cost(source_address, destination_address, parcel_weight):
    # Replace with your actual distance and cost calculation logic
    distance = abs(ord(source_address[0]) - ord(destination_address[0]))
    shipping_cost = distance * parcel_weight * 0.1
    return shipping_cost

# Example usage
source_address_input = input("Enter source address: ")
destination_address_input = input("Enter destination address: ")
parcel_weight_input = float(input("Enter parcel weight: "))
shipping_cost = calculate_shipping_cost(source_address_input, destination_address_input, parcel_weight_input)
print("Shipping Cost:", shipping_cost)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Enter source address: A
Enter destination address: C
Enter parcel weight: 5
Shipping Cost: 1.0
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

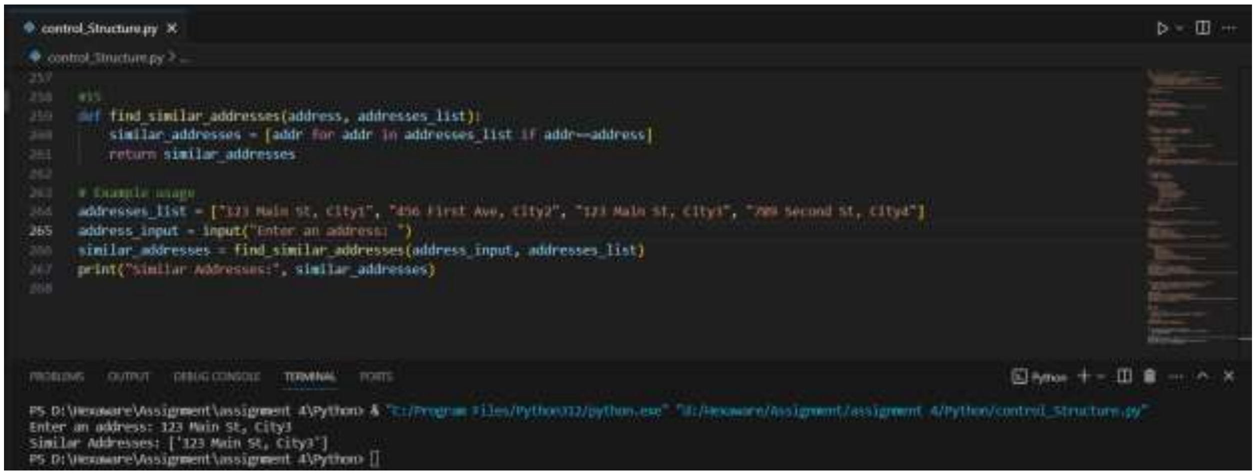
14.



```
#14
import random
import string

def generate_password():
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for _ in range(12))
    return password

# Example usage
generated_password = generate_password()
print("Generated Password:", generated_password)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Generated Password: p9X|5HF.}8BC
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Generated Password: gukd5~-k(h2~
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Generated Password: uOEn:|!75k*^
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

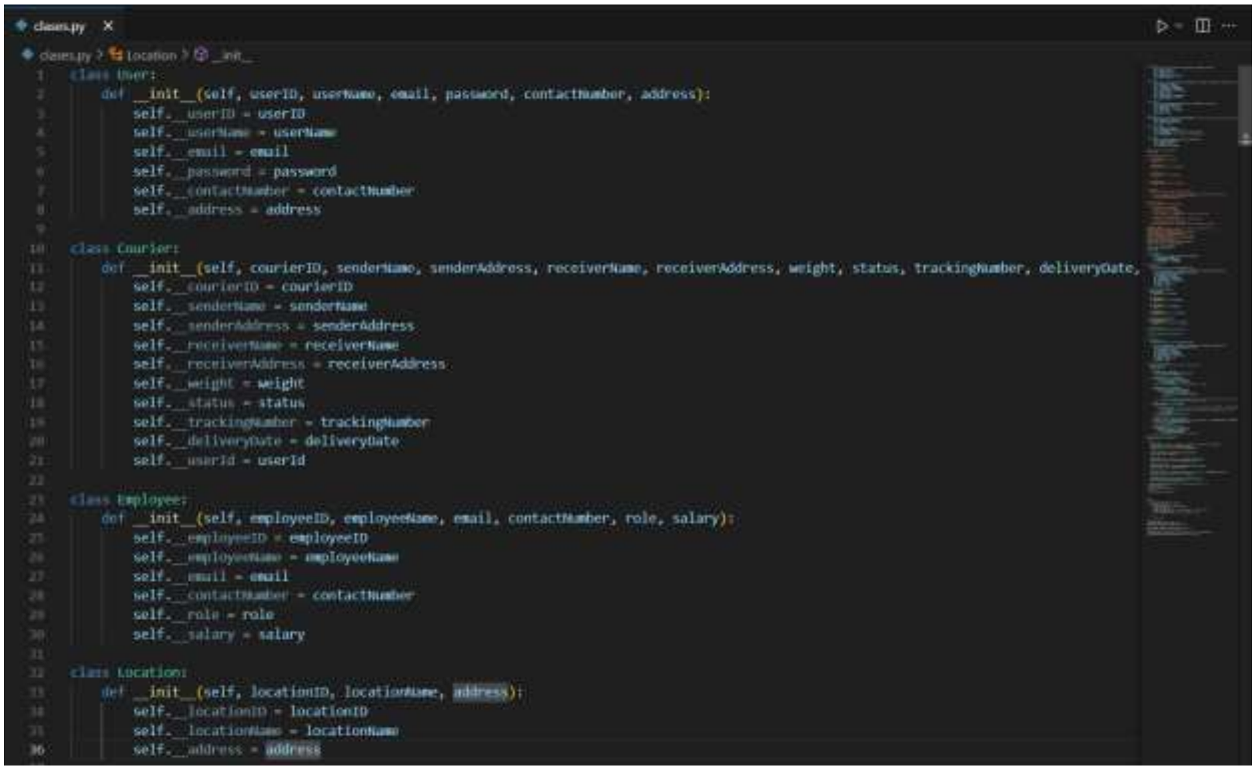
15.



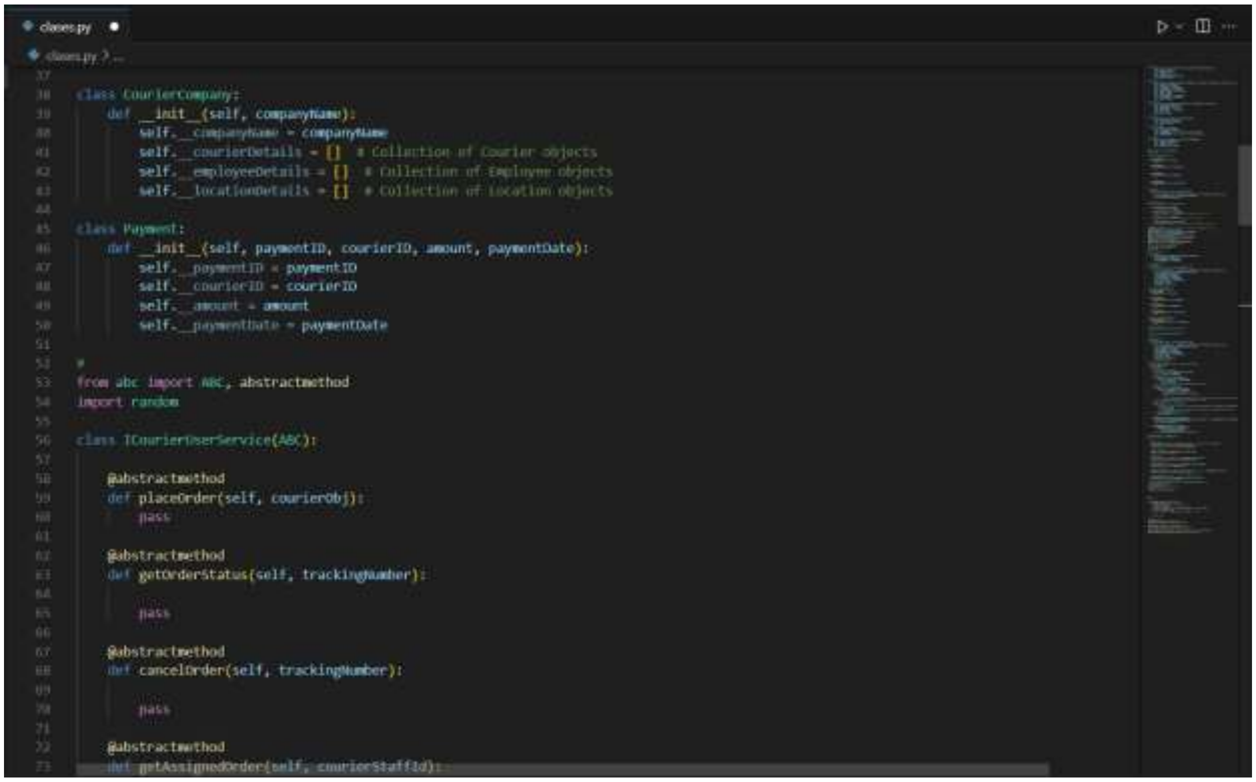
```

#15
def find_similar_addresses(address, addresses_list):
    similar_addresses = [addr for addr in addresses_list if addr==address]
    return similar_addresses

# Example usage
addresses_list = ["123 Main St, City1", "456 First Ave, City2", "123 Main St, City3", "789 Second St, City4"]
address_input = input("Enter an address: ")
similar_addresses = find_similar_addresses(address_input, addresses_list)
print("Similar Addresses:", similar_addresses)
```

```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/control_Structure.py"
Enter an address: 123 Main St, City3
Similar Addresses: ['123 Main St, City3']
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

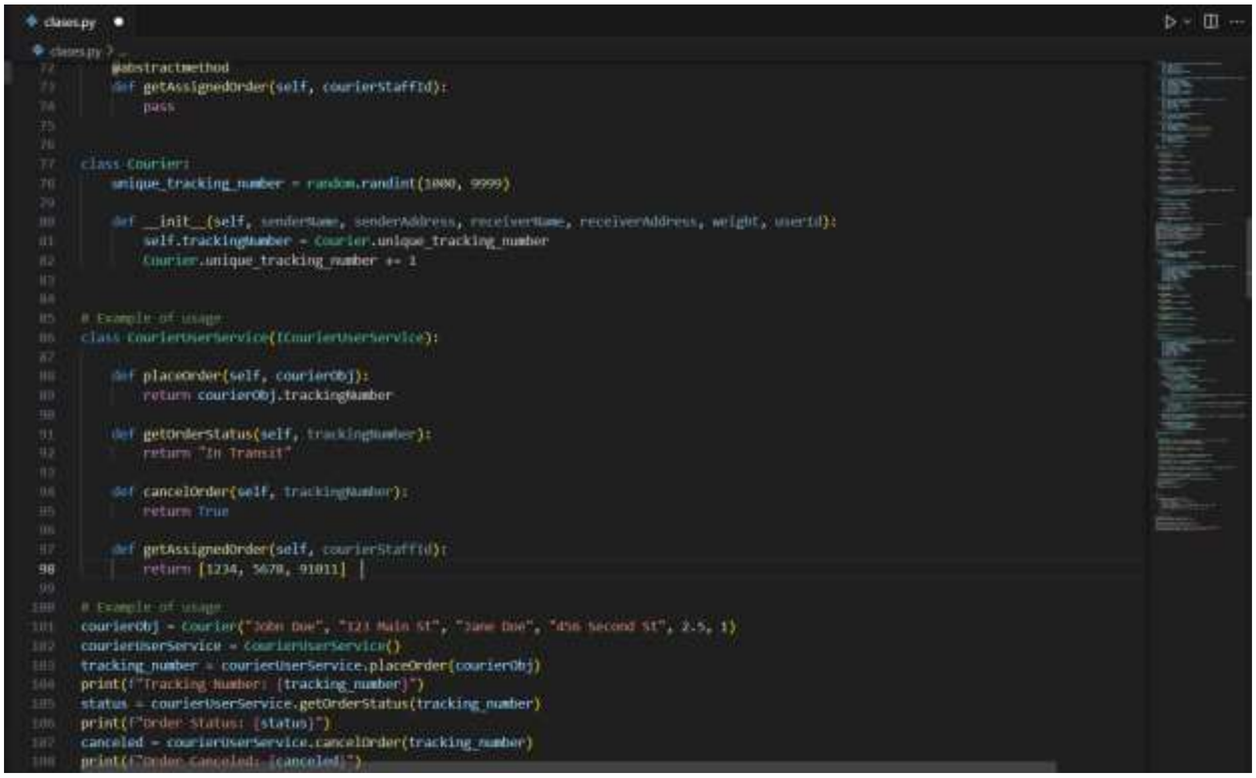
Task(5,6,7,8):



```python
class User:
    def __init__(self, userID, userName, email, password, contactNumber, address):
        self.__userID = userID
        self.__userName = userName
        self.__email = email
        self.__password = password
        self.__contactNumber = contactNumber
        self.__address = address

class Courier:
    def __init__(self, courierID, senderName, senderAddress, receiverName, receiverAddress, weight, status, trackingNumber, deliveryDate,
        self.__courierID = courierID
        self.__senderName = senderName
        self.__senderAddress = senderAddress
        self.__receiverName = receiverName
        self.__receiverAddress = receiverAddress
        self.__weight = weight
        self.__status = status
        self.__trackingNumber = trackingNumber
        self.__deliveryDate = deliveryDate
        self.__userId = userId

class Employee:
    def __init__(self, employeeID, employeeName, email, contactNumber, role, salary):
        self.__employeeID = employeeID
        self.__employeeName = employeeName
        self.__email = email
        self.__contactNumber = contactNumber
        self.__role = role
        self.__salary = salary

class Location:
    def __init__(self, locationID, locationName, address):
        self.__locationID = locationID
        self.__locationName = locationName
        self.__address = address
```

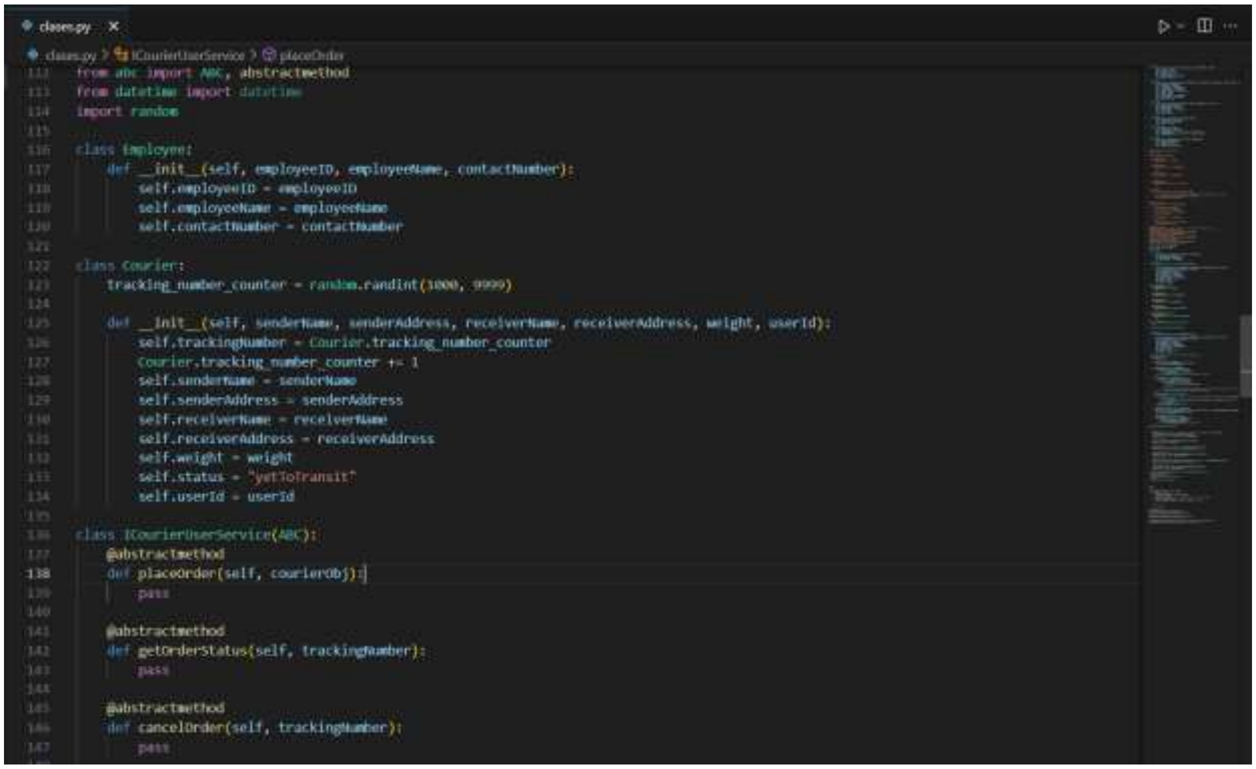


```python

class CourierCompany:
    def __init__(self, companyName):
        self.__companyName = companyName
        self.__courierDetails = []  # Collection of Courier objects
        self.__employeeDetails = []  # Collection of Employee objects
        self.__locationDetails = []  # Collection of Location objects

class Payment:
    def __init__(self, paymentID, courierID, amount, paymentDate):
        self.__paymentID = paymentID
        self.__courierID = courierID
        self.__amount = amount
        self.__paymentDate = paymentDate

#
from abc import ABC, abstractmethod
import random

class ICourierUserService(ABC):

    @abstractmethod
    def placeOrder(self, courierObj):
        pass

    @abstractmethod
    def getOrderStatus(self, trackingNumber):

        pass

    @abstractmethod
    def cancelOrder(self, trackingNumber):

        pass

    @abstractmethod
    def getAssignedOrder(self, courierStaffId):
```

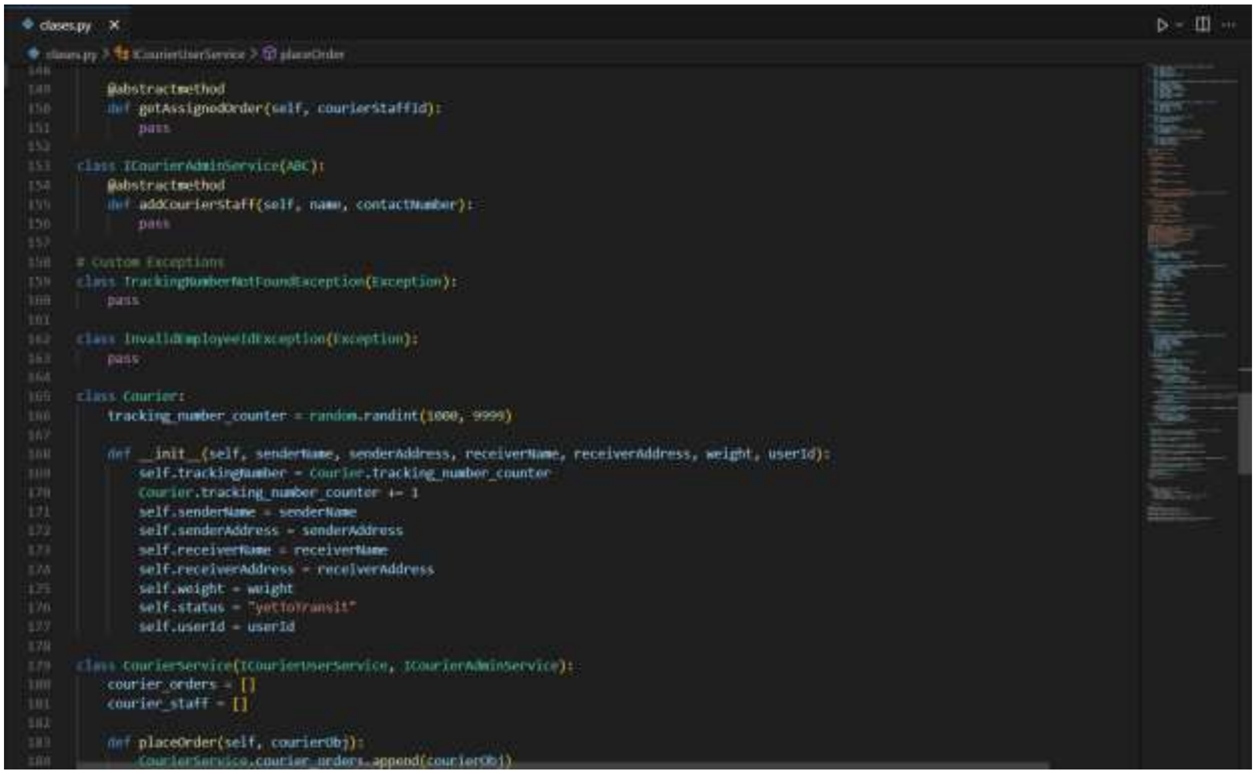


```python
    @abstractmethod
    def getAssignedOrder(self, courierStaffId):
        pass


class Courier:
    unique_tracking_number = random.randint(1000, 9999)

    def __init__(self, senderName, senderAddress, receiverName, receiverAddress, weight, userId):
        self.trackingNumber = Courier.unique_tracking_number
        Courier.unique_tracking_number += 1


# Example of usage
class CourierUserService(ICourierUserService):

    def placeOrder(self, courierObj):
        return courierObj.trackingNumber

    def getOrderStatus(self, trackingNumber):
        return "In Transit"

    def cancelOrder(self, trackingNumber):
        return True

    def getAssignedOrder(self, courierStaffId):
        return [1234, 5678, 91011]

# Example of usage
courierObj = Courier("John Doe", "123 Main St", "Jane Doe", "456 Second St", 2.5, 1)
courierUserService = CourierUserService()
tracking_number = courierUserService.placeOrder(courierObj)
print(f"Tracking Number: {tracking_number}")
status = courierUserService.getOrderStatus(tracking_number)
print(f"Order Status: {status}")
canceled = courierUserService.cancelOrder(tracking_number)
print(f"Order Canceled: {canceled}")
```

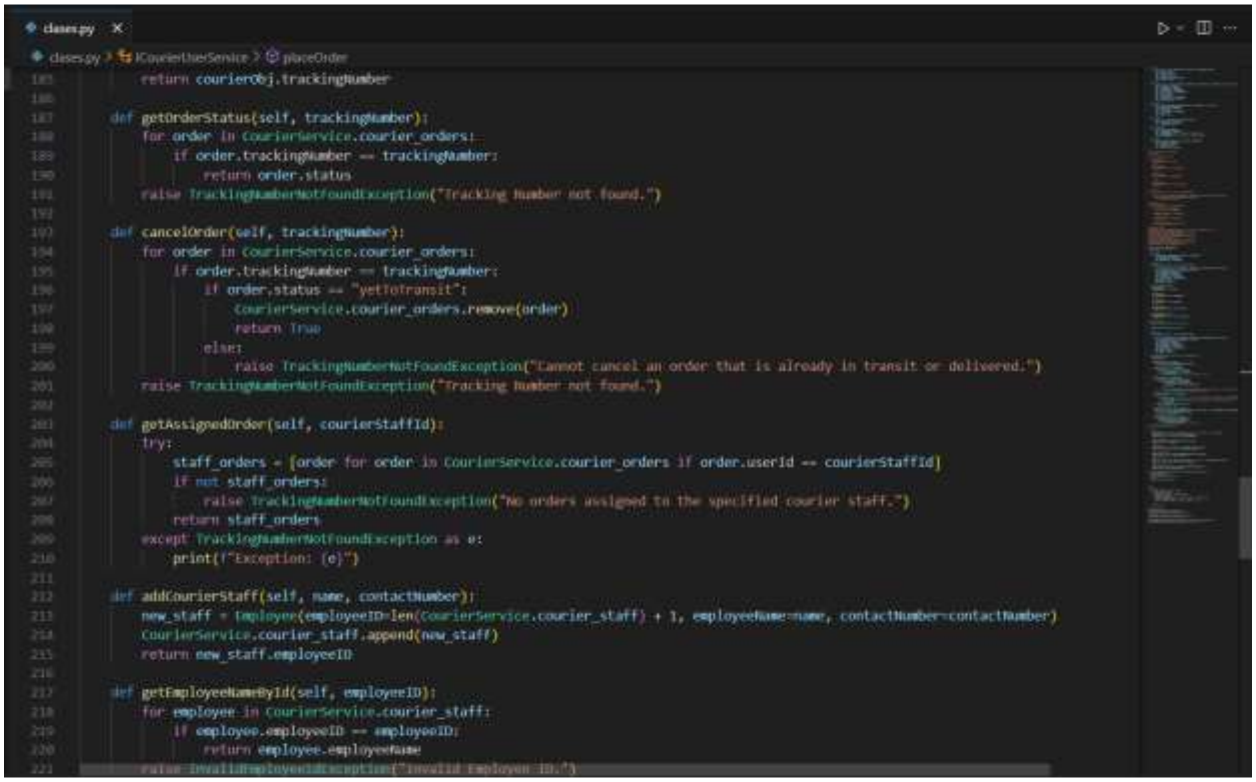
Implementation of methods in classes:



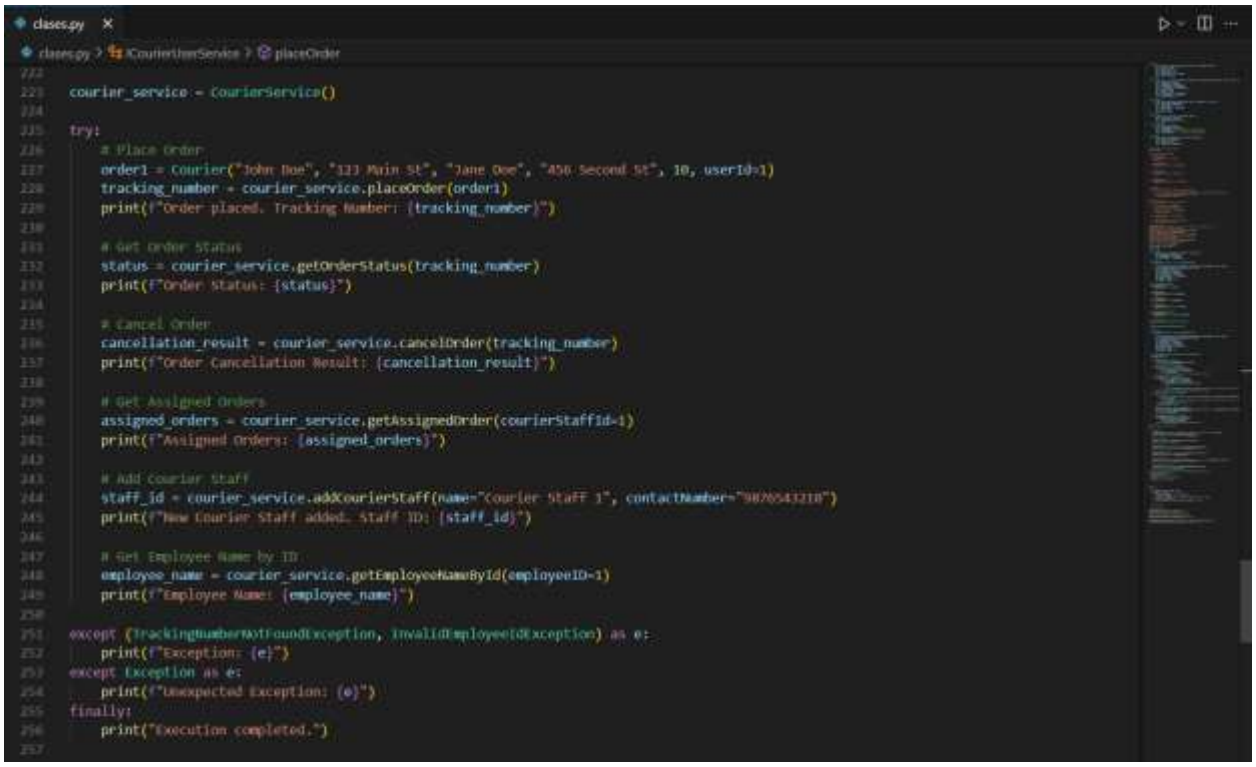
```python
from abc import ABC, abstractmethod
from datetime import datetime
import random

class Employee:
    def __init__(self, employeeID, employeeName, contactNumber):
        self.employeeID = employeeID
        self.employeeName = employeeName
        self.contactNumber = contactNumber

class Courier:
    tracking_number_counter = random.randint(1000, 9999)

    def __init__(self, senderName, senderAddress, receiverName, receiverAddress, weight, userId):
        self.trackingNumber = Courier.tracking_number_counter
        Courier.tracking_number_counter += 1
        self.senderName = senderName
        self.senderAddress = senderAddress
        self.receiverName = receiverName
        self.receiverAddress = receiverAddress
        self.weight = weight
        self.status = "yetToTransit"
        self.userId = userId

class ICourierUserService(ABC):
    @abstractmethod
    def placeOrder(self, courierObj):
        pass

    @abstractmethod
    def getOrderStatus(self, trackingNumber):
        pass

    @abstractmethod
    def cancelOrder(self, trackingNumber):
        pass
```



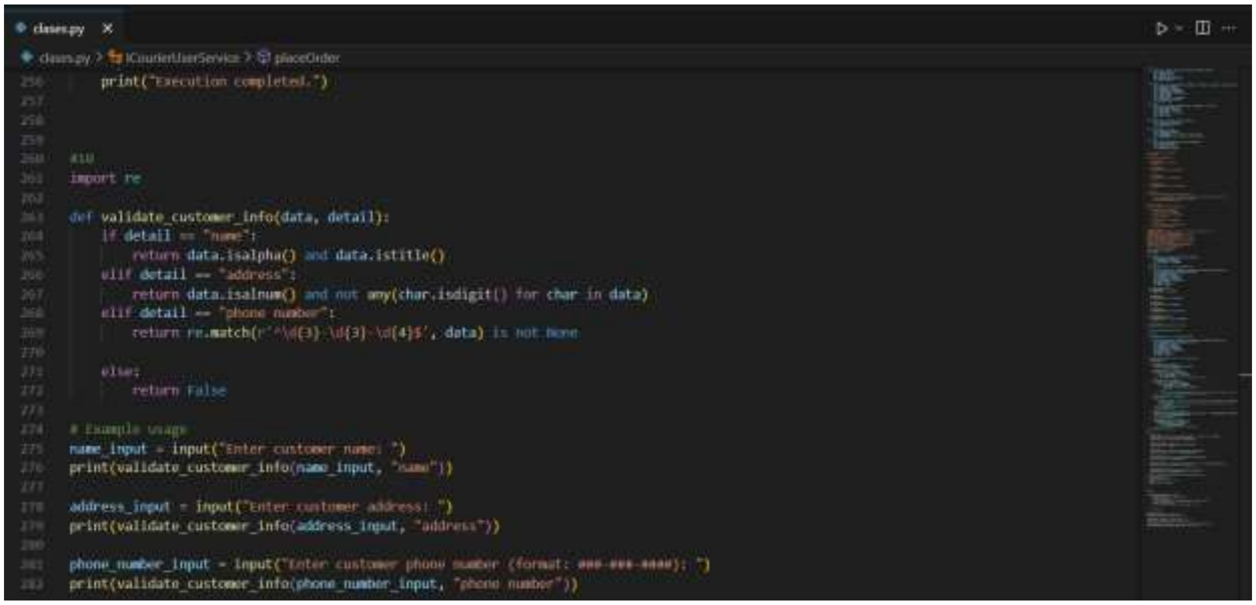
```python
    @abstractmethod
    def getAssignedOrder(self, courierStaffId):
        pass

class ICourierAdminService(ABC):
    @abstractmethod
    def addCourierStaff(self, name, contactNumber):
        pass

# Custom Exceptions
class TrackingNumberNotFoundException(Exception):
    pass

class InvalidEmployeeIdException(Exception):
    pass

class Courier:
    tracking_number_counter = random.randint(1000, 9999)

    def __init__(self, senderName, senderAddress, receiverName, receiverAddress, weight, userId):
        self.trackingNumber = Courier.tracking_number_counter
        Courier.tracking_number_counter += 1
        self.senderName = senderName
        self.senderAddress = senderAddress
        self.receiverName = receiverName
        self.receiverAddress = receiverAddress
        self.weight = weight
        self.status = "yetToTransit"
        self.userId = userId

class CourierService(ICourierUserService, ICourierAdminService):
    courier_orders = []
    courier_staff = []

    def placeOrder(self, courierObj):
        CourierService.courier_orders.append(courierObj)
```



```python
        return courierObj.trackingNumber

    def getOrderStatus(self, trackingNumber):
        for order in CourierService.courier_orders:
            if order.trackingNumber == trackingNumber:
                return order.status
        raise TrackingNumberNotFoundException("Tracking Number not found.")

    def cancelOrder(self, trackingNumber):
        for order in CourierService.courier_orders:
            if order.trackingNumber == trackingNumber:
                if order.status == "yetToTransit":
                    CourierService.courier_orders.remove(order)
                    return True
                else:
                    raise TrackingNumberNotFoundException("Cannot cancel an order that is already in transit or delivered.")
        raise TrackingNumberNotFoundException("Tracking Number not found.")

    def getAssignedOrder(self, courierStaffId):
        try:
            staff_orders = [order for order in CourierService.courier_orders if order.userId == courierStaffId]
            if not staff_orders:
                raise TrackingNumberNotFoundException("No orders assigned to the specified courier staff.")
            return staff_orders
        except TrackingNumberNotFoundException as e:
            print(f"Exception: {e}")

    def addCourierStaff(self, name, contactNumber):
        new_staff = Employee(employeeID=len(CourierService.courier_staff) + 1, employeeName=name, contactNumber=contactNumber)
        CourierService.courier_staff.append(new_staff)
        return new_staff.employeeID

    def getEmployeeNameById(self, employeeID):
        for employee in CourierService.courier_staff:
            if employee.employeeID == employeeID:
                return employee.employeeName
        raise InvalidEmployeeIdException("Invalid Employee ID.")
```

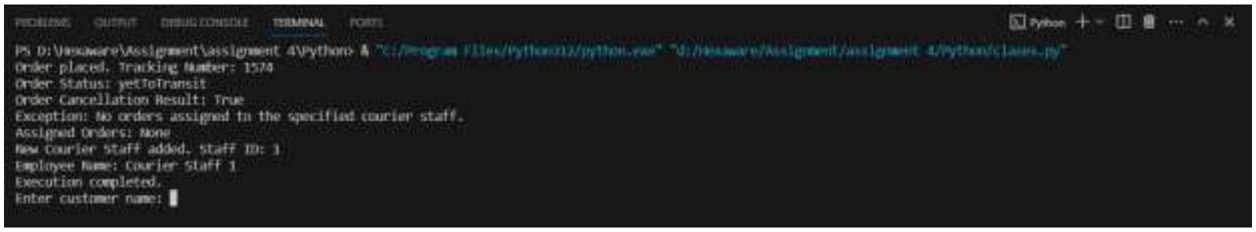


```python

courier_service = CourierService()

try:
    # Place Order
    order1 = Courier("John Doe", "123 Main St", "Jane Doe", "456 Second St", 10, userId=1)
    tracking_number = courier_service.placeOrder(order1)
    print(f"Order placed. Tracking Number: {tracking_number}")

    # Get Order Status
    status = courier_service.getOrderStatus(tracking_number)
    print(f"Order Status: {status}")

    # Cancel Order
    cancellation_result = courier_service.cancelOrder(tracking_number)
    print(f"Order Cancellation Result: {cancellation_result}")

    # Get Assigned Orders
    assigned_orders = courier_service.getAssignedOrder(courierStaffId=1)
    print(f"Assigned Orders: {assigned_orders}")

    # Add Courier Staff
    staff_id = courier_service.addCourierStaff(name="Courier Staff 1", contactNumber="9876543210")
    print(f"New Courier Staff added. Staff ID: {staff_id}")

    # Get Employee Name by ID
    employee_name = courier_service.getEmployeeNameById(employeeID=1)
    print(f"Employee Name: {employee_name}")

except (TrackingNumberNotFoundException, InvalidEmployeeIdException) as e:
    print(f"Exception: {e}")
except Exception as e:
    print(f"Unexpected Exception: {e}")
finally:
    print("Execution completed.")

```



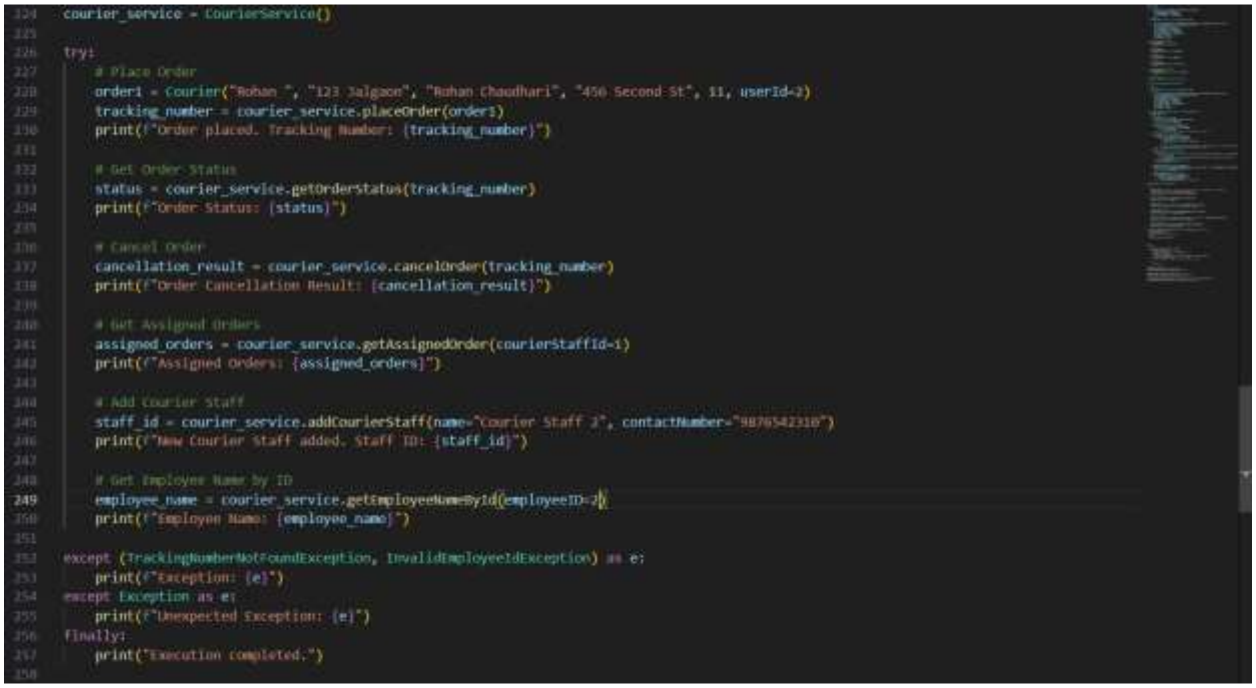
```python
    print("Execution completed.")



#10
import re

def validate_customer_info(data, detail):
    if detail == "name":
        return data.isalpha() and data.istitle()
    elif detail == "address":
        return data.isalnum() and not any(char.isdigit() for char in data)
    elif detail == "phone number":
        return re.match(r'^\d{3}-\d{3}-\d{4}$', data) is not None

    else:
        return False

# Example usage
name_input = input("Enter customer name: ")
print(validate_customer_info(name_input, "name"))

address_input = input("Enter customer address: ")
print(validate_customer_info(address_input, "address"))

phone_number_input = input("Enter customer phone number (format: ###-###-####): ")
print(validate_customer_info(phone_number_input, "phone number"))
```

Output:



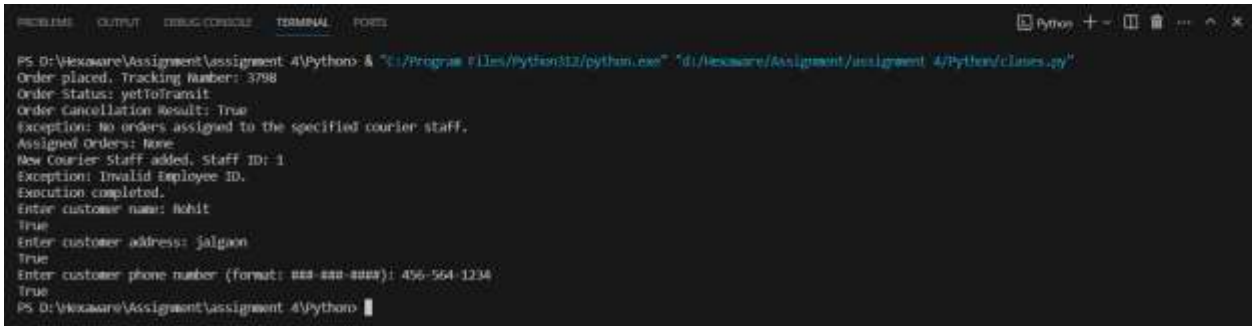
```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/clases.py"
Order placed. Tracking Number: 1574
Order Status: yetToTransit
Order Cancellation Result: True
Exception: No orders assigned to the specified courier staff.
Assigned Orders: None
New Courier Staff added. Staff ID: 1
Employee Name: Courier Staff 1
Execution completed.
Enter customer name: 
```

New inputs:



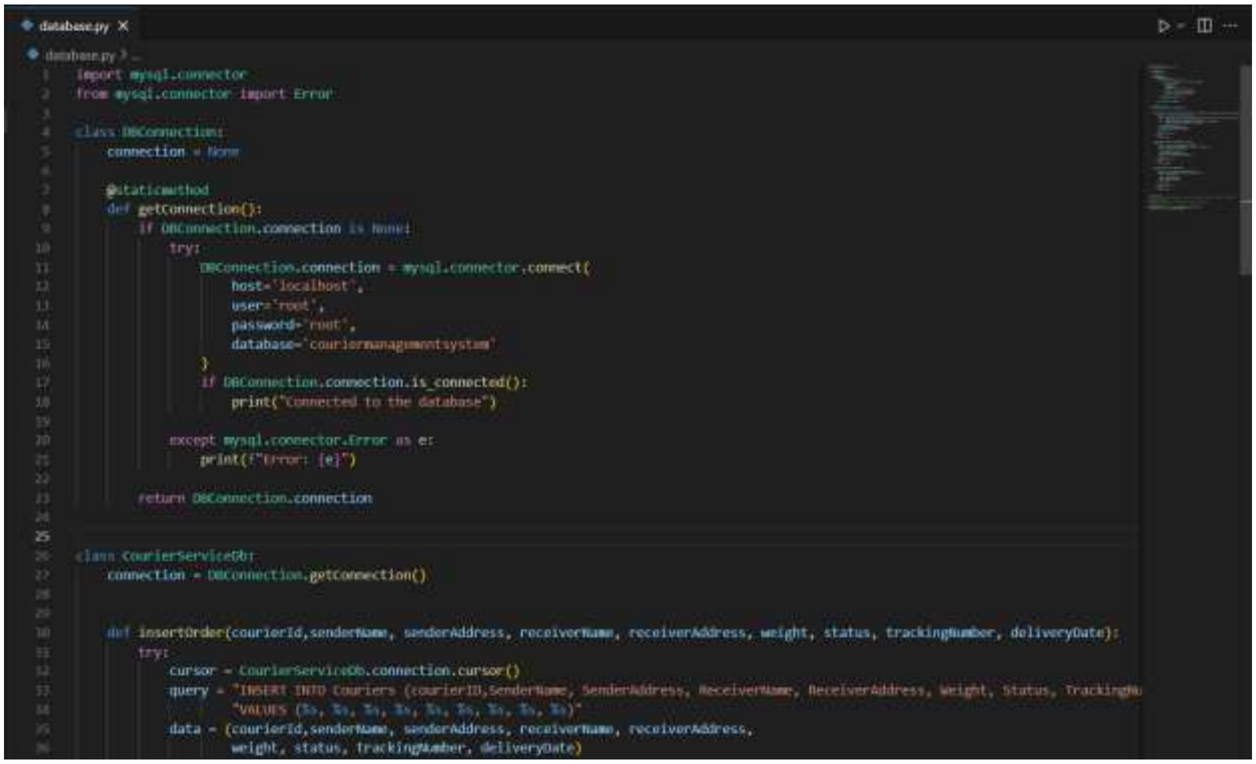
```python
courier_service = CourierService()

try:
    # Place Order
    order1 = Courier("Rohan ", "123 Jalgaon", "Rohan Chaudhari", "456 Second St", 11, userId=2)
    tracking_number = courier_service.placeOrder(order1)
    print(f"Order placed. Tracking Number: {tracking_number}")

    # Get Order Status
    status = courier_service.getOrderStatus(tracking_number)
    print(f"Order Status: {status}")

    # Cancel Order
    cancellation_result = courier_service.cancelOrder(tracking_number)
    print(f"Order Cancellation Result: {cancellation_result}")

    # Get Assigned Orders
    assigned_orders = courier_service.getAssignedOrder(courierStaffId=1)
    print(f"Assigned Orders: {assigned_orders}")

    # Add Courier Staff
    staff_id = courier_service.addCourierStaff(name="Courier Staff 2", contactNumber="9876542310")
    print(f"New Courier Staff added. Staff ID: {staff_id}")

    # Get Employee Name by ID
    employee_name = courier_service.getEmployeeNameById(employeeID=2)
    print(f"Employee Name: {employee_name}")

except (TrackingNumberNotFoundException, InvalidEmployeeIdException) as e:
    print(f"Exception: {e}")
except Exception as e:
    print(f"Unexpected Exception: {e}")
finally:
    print("Execution completed.")
```

Output:



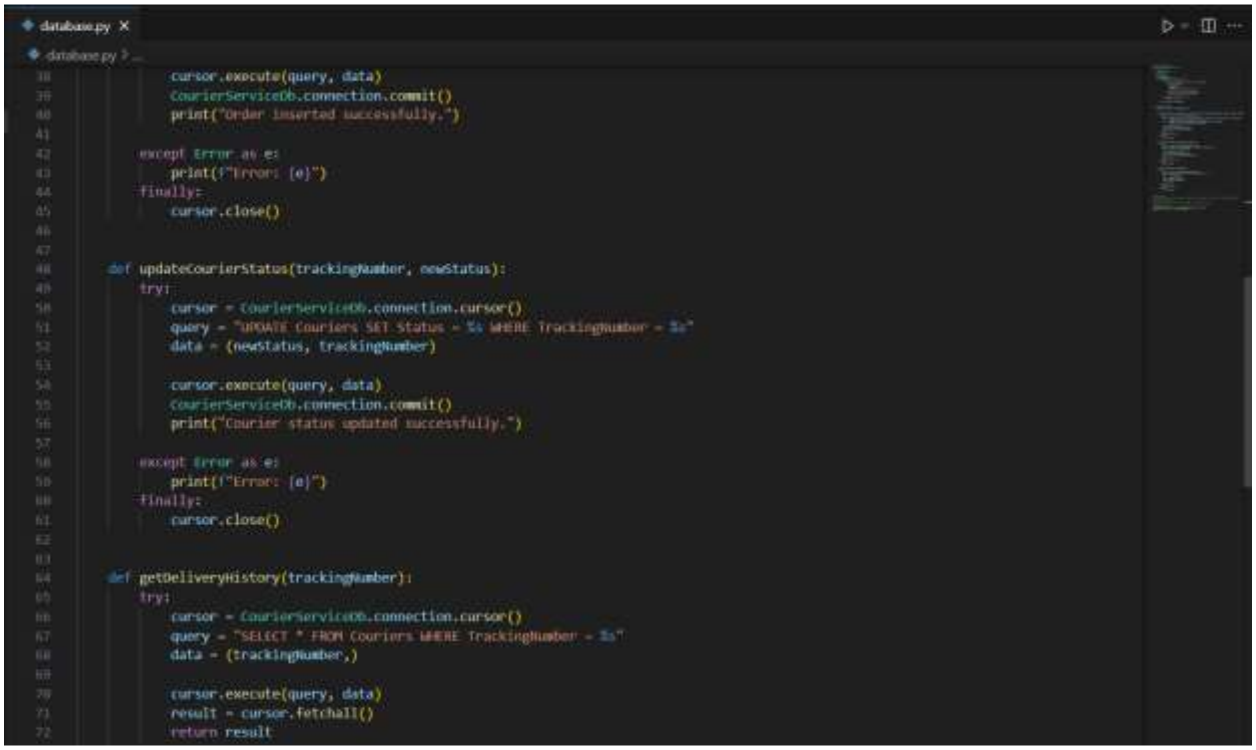
```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/clases.py"
Order placed. Tracking Number: 3798
Order Status: yetToTransit
Order Cancellation Result: True
Exception: No orders assigned to the specified courier staff.
Assigned Orders: None
New Courier Staff added. Staff ID: 1
Exception: Invalid Employee ID.
Execution completed.
Enter customer name: Rohit
True
Enter customer address: jalgaon
True
Enter customer phone number (format: ###-###-####): 456-564-1234
True
PS D:\Hexaware\Assignment\assignment 4\Python> 
```

Task9:



database.py

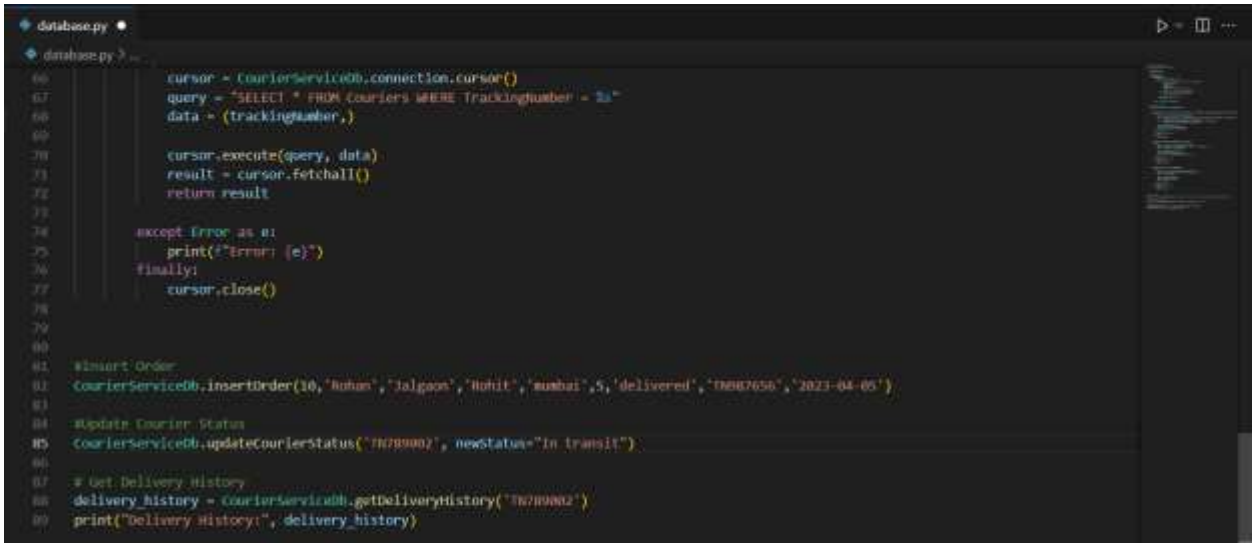
```python
import mysql.connector
from mysql.connector import Error

class DBConnection:
    connection = None

    @staticmethod
    def getConnection():
        if DBConnection.connection is None:
            try:
                DBConnection.connection = mysql.connector.connect(
                    host='localhost',
                    user='root',
                    password='root',
                    database='couriermanagementsystem'
                )
                if DBConnection.connection.is_connected():
                    print("Connected to the database")

            except mysql.connector.Error as e:
                print(f"Error: {e}")

        return DBConnection.connection


class CourierServiceDb:
    connection = DBConnection.getConnection()


    def insertOrder(courierId,senderName, senderAddress, receiverName, receiverAddress, weight, status, trackingNumber, deliveryDate):
        try:
            cursor = CourierServiceDb.connection.cursor()
            query = "INSERT INTO Couriers (courierID,SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNu
                    "VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)"
            data = (courierId,senderName, senderAddress, receiverName, receiverAddress,
                    weight, status, trackingNumber, deliveryDate)
```



database.py

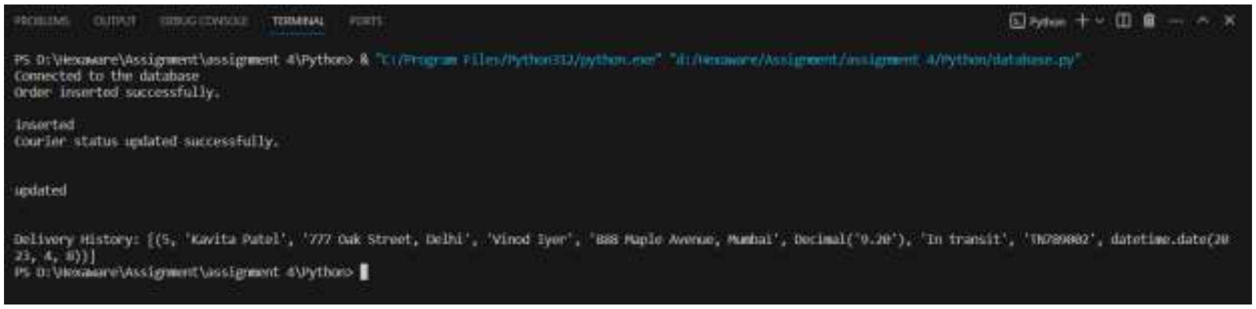
```python
            cursor.execute(query, data)
            CourierServiceDb.connection.commit()
            print("Order inserted successfully.")

        except Error as e:
            print(f"Error: {e}")
        finally:
            cursor.close()


    def updateCourierStatus(trackingNumber, newStatus):
        try:
            cursor = CourierServiceDb.connection.cursor()
            query = "UPDATE Couriers SET Status = %s WHERE TrackingNumber = %s"
            data = (newStatus, trackingNumber)

            cursor.execute(query, data)
            CourierServiceDb.connection.commit()
            print("Courier status updated successfully.")

        except Error as e:
            print(f"Error: {e}")
        finally:
            cursor.close()


    def getDeliveryHistory(trackingNumber):
        try:
            cursor = CourierServiceDb.connection.cursor()
            query = "SELECT * FROM Couriers WHERE TrackingNumber = %s"
            data = (trackingNumber,)

            cursor.execute(query, data)
            result = cursor.fetchall()
            return result
```

Inputs:



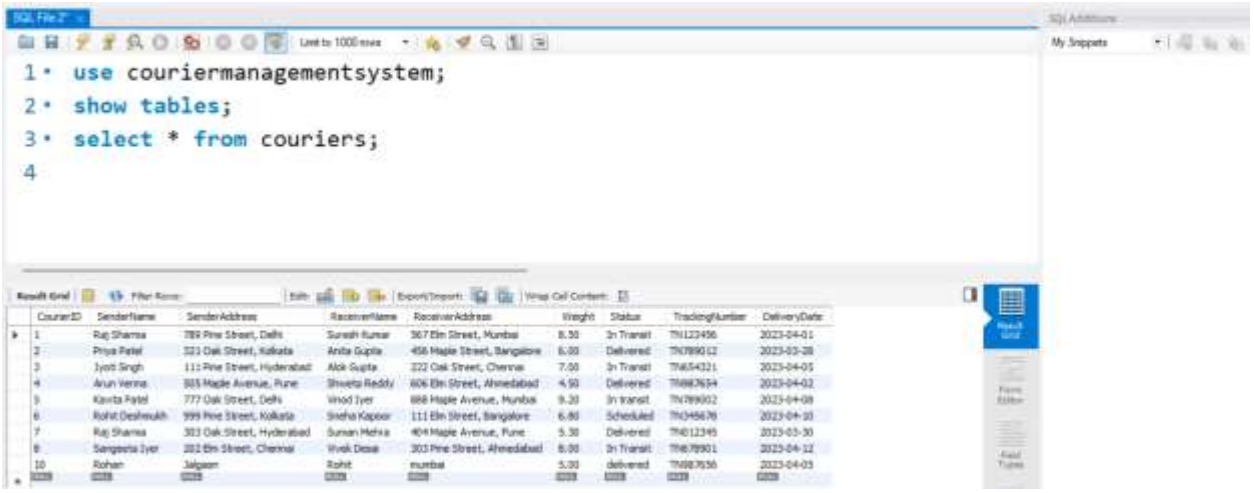
```
            cursor = CourierServiceDb.connection.cursor()
            query = "SELECT * FROM Couriers WHERE TrackingNumber = %s"
            data = (trackingNumber,)

            cursor.execute(query, data)
            result = cursor.fetchall()
            return result

        except Error as e:
            print(f"Error: {e}")
        finally:
            cursor.close()



#Insert Order
CourierServiceDb.insertOrder(10,'Rohan','Jalgaon','Rohit','mumbai',5,'delivered','TN987656','2023-04-05')

#Update Courier Status
CourierServiceDb.updateCourierStatus('TN789002', newStatus="In transit")

# Get Delivery History
delivery_history = CourierServiceDb.getDeliveryHistory('TN789002')
print("Delivery History:", delivery_history)
```

Output:



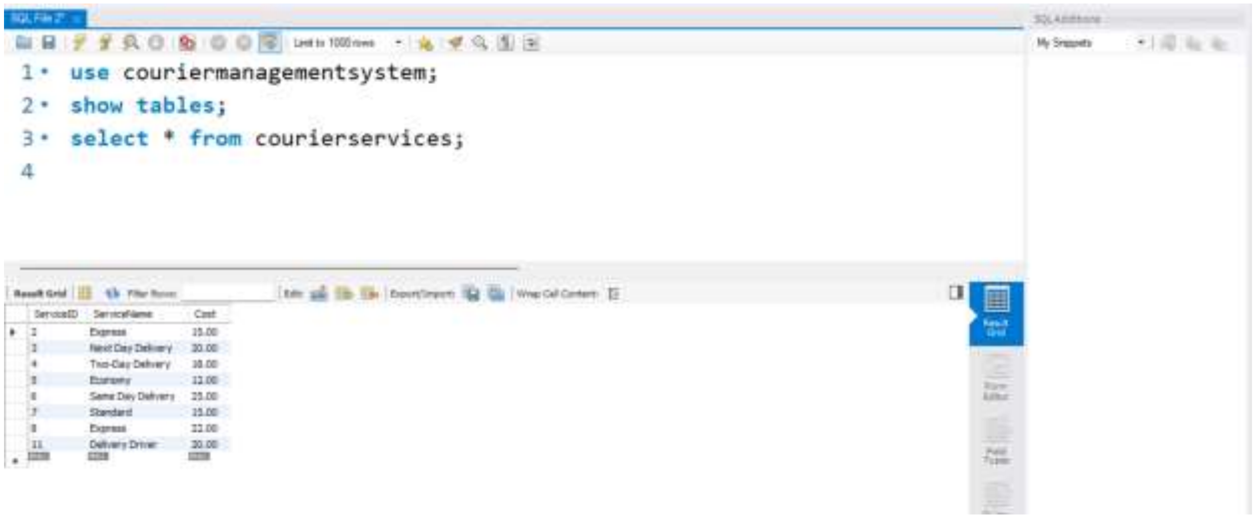
```
PS D:\Hexaware\Assignment\assignment 4\Python> & "C:/Program Files/Python312/python.exe" "d:/Hexaware/Assignment/assignment 4/Python/database.py"
Connected to the database
Order inserted successfully.

inserted
Courier status updated successfully.


updated


Delivery History: [(5, 'Kavita Patel', '777 Oak Street, Delhi', 'Vinod Iyer', '888 Maple Avenue, Mumbai', Decimal('9.20'), 'In transit', 'TN789002', datetime.date(20
23, 4, 8))]
PS D:\Hexaware\Assignment\assignment 4\Python>
```

Database changes:



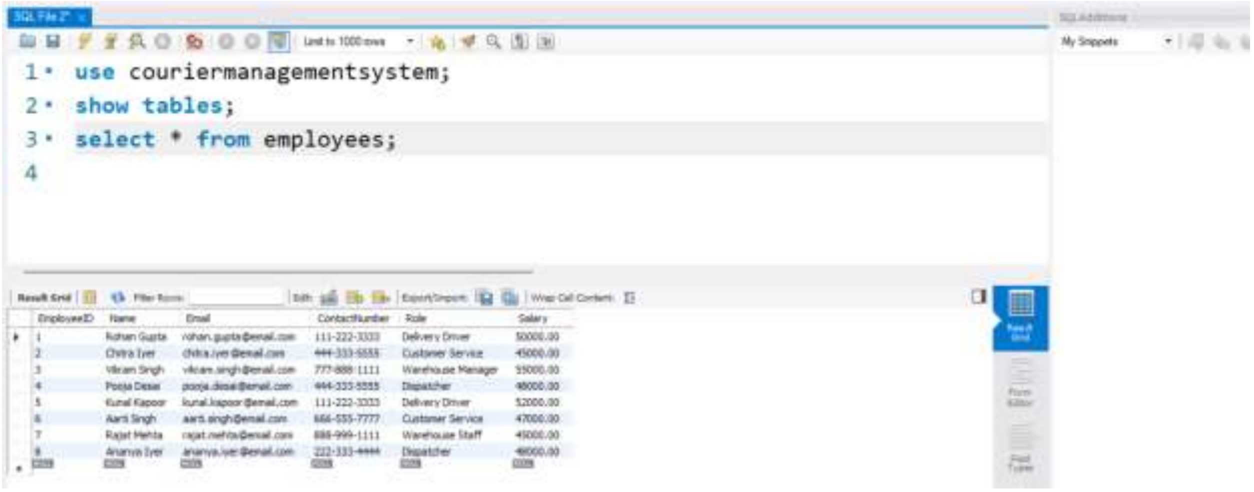
```sql
use couriermanagementsystem;
show tables;
select * from couriers;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	367 Elm Street, Mumbai	8.50	In Transit	TN123456	2023-04-01
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered	TN789012	2023-03-28
3	Jyoti Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TN654321	2023-04-05
4	Arun Verma	505 Maple Avenue, Pune	Shweta Reddy	606 Elm Street, Ahmedabad	4.50	Delivered	TN987654	2023-04-02
5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.20	In transit	TN789002	2023-04-08
6	Rohit Deshmukh	999 Pine Street, Kolkata	Sneha Kapoor	111 Elm Street, Bangalore	6.80	Scheduled	TN345678	2023-04-10
7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehta	404 Maple Avenue, Pune	5.30	Delivered	TN012345	2023-03-30
8	Sangeeta Iyer	202 Elm Street, Chennai	Vivek Desai	303 Pine Street, Ahmedabad	8.00	In Transit	TN678901	2023-04-12
10	Rohan	Jalgaon	Rohit	mumbai	5.00	delivered	TN987656	2023-04-05
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



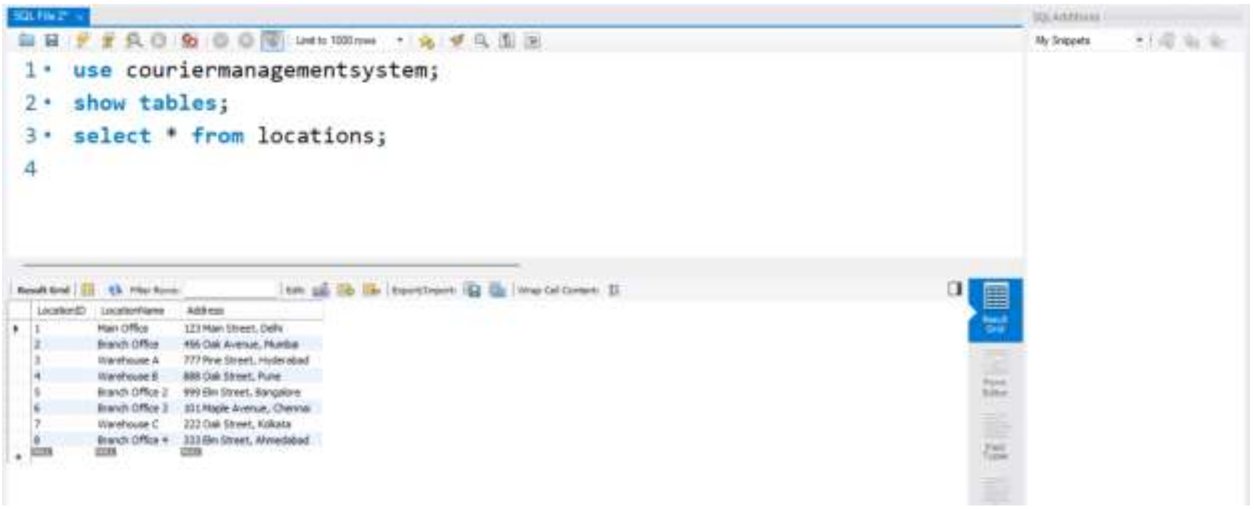
```sql
use couriermanagementsystem;
show tables;
select * from courierservices;
```

ServiceID	ServiceName	Cost
2	Express	15.00
3	Next Day Delivery	20.00
4	Two-Day Delivery	18.00
5	Economy	12.00
6	Same Day Delivery	25.00
7	Standard	15.00
8	Express	22.00
11	Delivery Driver	20.00
NULL	NULL	NULL



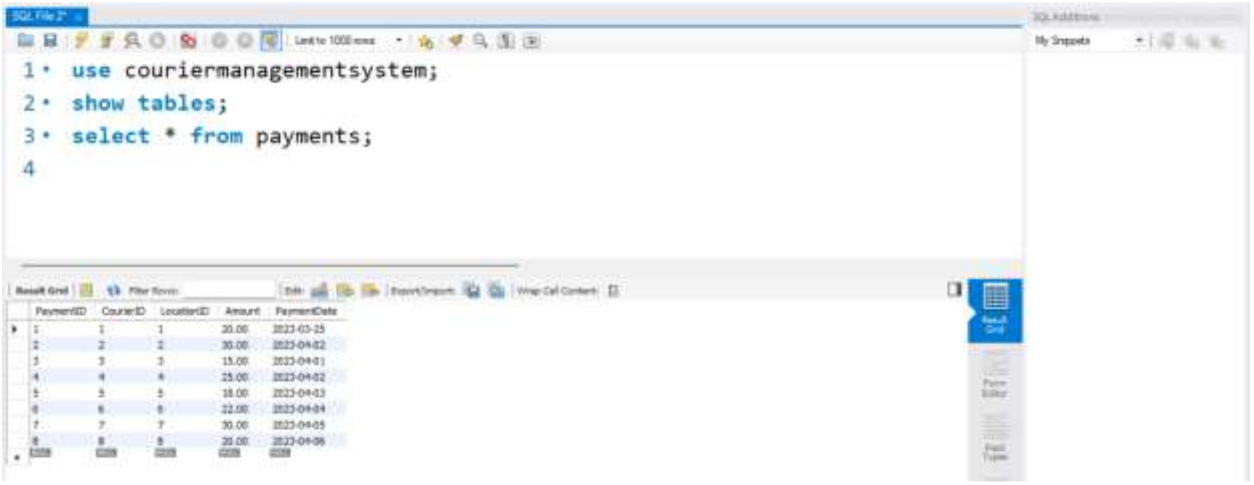
```sql
use couriermanagementsystem;
show tables;
select * from employees;
```

EmployeeID	Name	Email	ContactNumber	Role	Salary
1	Rohan Gupta	rohan.gupta@email.com	111-222-3333	Delivery Driver	50000.00
2	Chitra Iyer	chitra.iyer@email.com	444-333-5555	Customer Service	45000.00
3	Vikram Singh	vikram.singh@email.com	777-888-1111	Warehouse Manager	55000.00
4	Pooja Desai	pooja.desai@email.com	444-333-5555	Dispatcher	48000.00
5	Kunal Kapoor	kunal.kapoor@email.com	111-222-3333	Delivery Driver	52000.00
6	Aarti Singh	aarti.singh@email.com	666-555-7777	Customer Service	47000.00
7	Rajat Mehta	rajat.mehta@email.com	888-999-1111	Warehouse Staff	45000.00
8	Ananya Iyer	ananya.iyer@email.com	222-333-4444	Dispatcher	48000.00
NULL	NULL	NULL	NULL	NULL	NULL



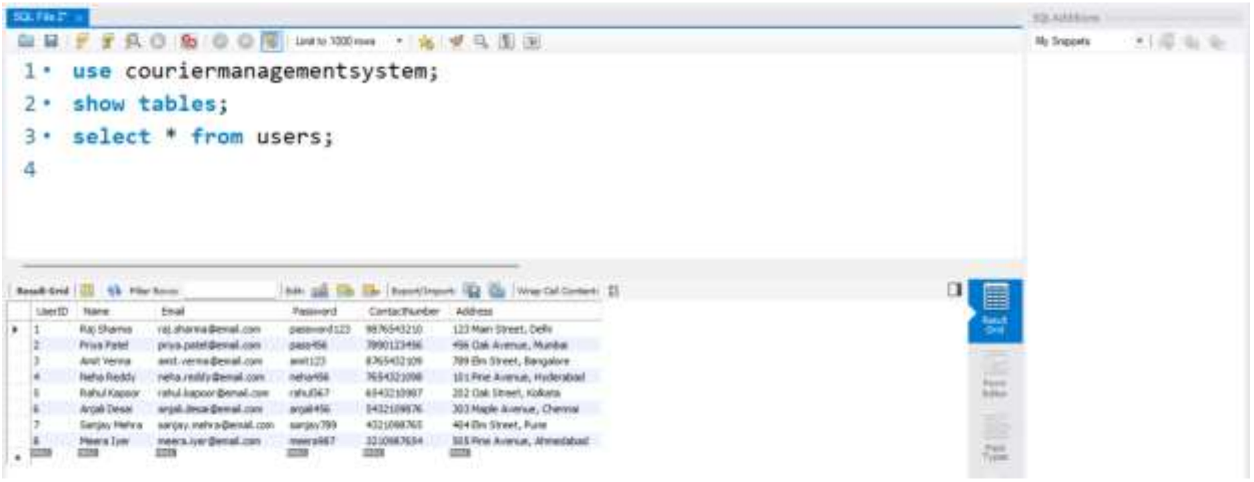
```sql
use couriermanagementsystem;
show tables;
select * from locations;
```

LocationID	LocationName	Address
1	Main Office	123 Main Street, Delhi
2	Branch Office	456 Oak Avenue, Mumbai
3	Warehouse A	777 Pine Street, Hyderabad
4	Warehouse B	888 Oak Street, Pune
5	Branch Office 2	999 Elm Street, Bangalore
6	Branch Office 3	101 Maple Avenue, Chennai
7	Warehouse C	222 Oak Street, Kolkata
8	Branch Office 4	333 Elm Street, Ahmedabad



```sql
use couriermanagementsystem;
show tables;
select * from payments;
```

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	1	20.00	2023-03-25
2	2	2	30.00	2023-04-02
3	3	3	15.00	2023-04-01
4	4	4	25.00	2023-04-02
5	5	5	18.00	2023-04-03
6	6	6	22.00	2023-04-04
7	7	7	30.00	2023-04-05
8	8	8	20.00	2023-04-06



```sql
use couriermanagementsystem;
show tables;
select * from users;
```

UserID	Name	Email	Password	ContactNumber	Address
1	Raj Sharma	raj.sharma@email.com	password123	9876543210	123 Main Street, Delhi
2	Priya Patel	priya.patel@email.com	pass456	7890123456	456 Oak Avenue, Mumbai
3	Amit Verma	amit.verma@email.com	amit123	8765432109	789 Elm Street, Bangalore
4	Neha Reddy	neha.reddy@email.com	neha456	7654321098	101 Pine Avenue, Hyderabad
5	Rahul Kapoor	rahul.kapoor@email.com	rahul567	6543210987	202 Oak Street, Kolkata
6	Anjali Desai	anjali.desai@email.com	anjali456	5432109876	303 Maple Avenue, Chennai
7	Sanjay Mehra	sanjay.mehra@email.com	sanjay789	4321098765	404 Elm Street, Pune
8	Meera Iyer	meera.iyer@email.com	meera987	3210987654	505 Pine Avenue, Ahmedabad