**Logistics and Management System**

**Name: Dasin Madanayake**

**Student ID: AS20240566**

**Course: University of Sri Jayawardanapura BSc Physical Science**

**Submission Date: 10/21/2025**

**1. Introduction**

The Logistics Management System is designed to automate and simplify the process of managing deliveries, cities, and vehicles. It allows managing city data, calculating distances, handling delivery requests, calculating delivery costs and fuel usage, and generating performance reports for efficient management. The system supports up to 30 cities and 50 delivery records and it is coded using the C programming language using arrays, pointers, functions and file handling.

**2. Objectives**

The main objectives of the Logistics Management System are:

1. Storing and managing city names.
2. Storing and managing distances between cities efficiently in a software system
3. Storing and managing data of vehicles and their properties like capacity and fuel consumption
4. Handle delivery requests based on starting and destination cities, cargo weight, and vehicle selection.
5. Calculating of delivery costs, operational costs, fuel consumption, profit, and final customer charges.
6. Storing and maintaining delivery records for future reference.
7. Generate performance reports.

**3. System Design**

**3.1 Data Structures**

* **Cities:**  
  char cities[int MAX\_CITIES][50];
* **Distance Matrix:**  
  float distance[MAX\_CITIES][MAX\_CITIES]
* **Vehicles:**  
  Vehicles are represented as arrays:
  + int capacity[3] ={1000,5000,10000};
  + float ratePerKm[3] = {30, 40, 80} ;
  + float avgSpeed[3] = {60, 50, 45} ;
  + float fuelEff[3] = {12, 6, 4} ;
  + char types[3][20]= {"Van", "Truck", "Lorry"};
* **Deliveries:**  
  Multiple arrays store delivery details:
  + Int deliverySrc[MAX\_DEL]
  + Int deliveryDes[MAX\_DEL]
  + Float deliveryWei[MAX\_DEL]
  + int deliveryVehicle[MAX\_DEL]
  + float deliveryCost, float deliveryTime, float fuelUsed, float fuelCost, float profit, float customerCharge

**3.2 Functional Modules**

1. **City Management**
   * Add, rename, remove, and display cities.
2. **Distance Management**
   * Enter, update, and display distances between city pairs.
3. **Vehicle Management**
   * Displays all vehicles with their capacity, rate per km, average speed, and fuel efficiency.
4. **Delivery Requests**
   * Users select source and destination cities, vehicle type, and cargo weight.
   * Validates weight and vehicle data.
5. **Cost, Time, and Fuel Calculation**  
   For each delivery, these are calculated:
   * **Delivery Cost:** distance × rate × (1 + weight / 10000)
   * **Delivery Time:** distance / average speed
   * **Fuel Used:** distance/fuel efficiency
   * **Fuel Cost:** fuel used × fuel price
   * **Operational Cost:** delivery cost + fuel cost
   * **Profit:** 25% of delivery cost
   * **Customer Charge:** operational cost + profit
6. **Delivery Records**
   * Display detailed information about each delivery
7. **Performance Report**
   * Display a summary of total deliveries, total distance, average delivery time, total revenue, and total profit.
8. **File Handling functions**
   * saveRoutes and loadRoutes store and retrieve city and distance data.
   * saveDeliveries and loadDeliveries store and retrieve delivery records.

**4. Algorithm Design**

**4.1 City Management**

* Can add, rename and remove cities.

**4.2 Distance Management**

* Can enter and update distances between cities.
* Display formatted distance table.

**4.3 Delivery Request Handling**

1. User enters source and destination cities.
2. Then, enter the cargo weight and vehicle type.
3. Check weight limitations.
4. Validate city indices.
5. Record valid delivery into arrays.

**4.4 Cost, Time, and Fuel Calculations**

* Loop all deliveries and compute cost, time, fuel, profit, and final charge.

**4.5 Minimum Distance Algorithm**

* Calculates all possible routes and selects the one with the minimum distance (supports only for 2 to 4 cities).

**5. File Handling**

**5.1 Routes File** (routes.txt)

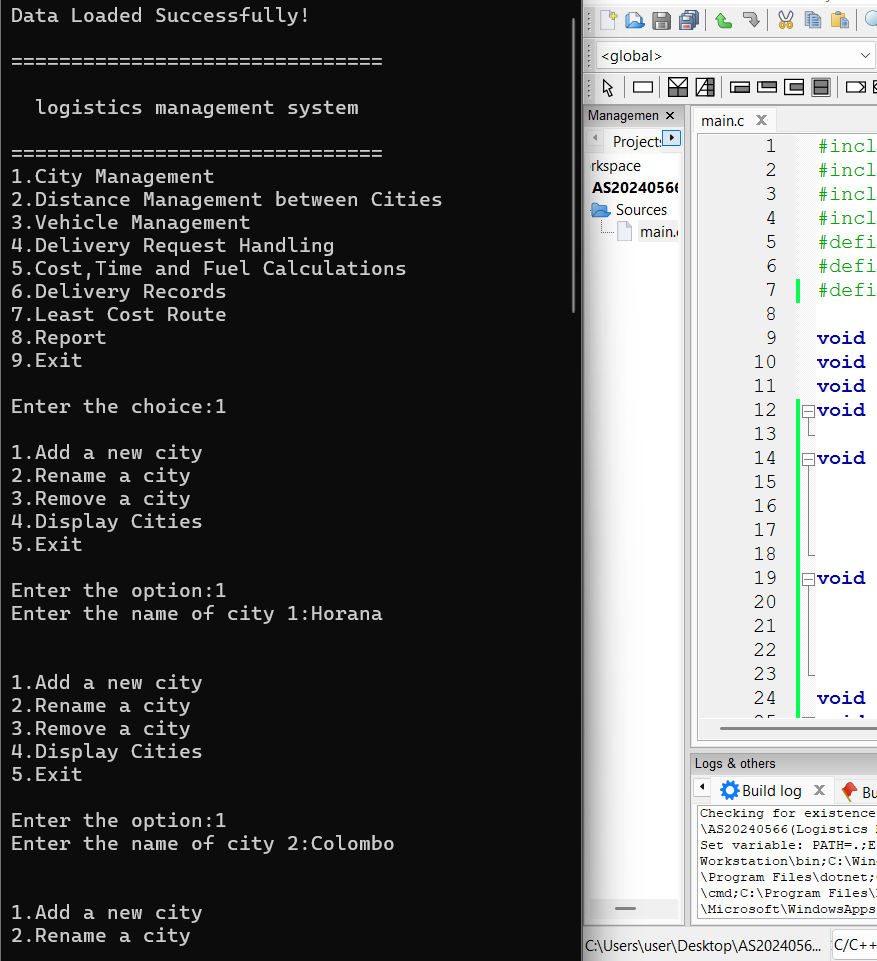
* Stores city names and the distance between cities.

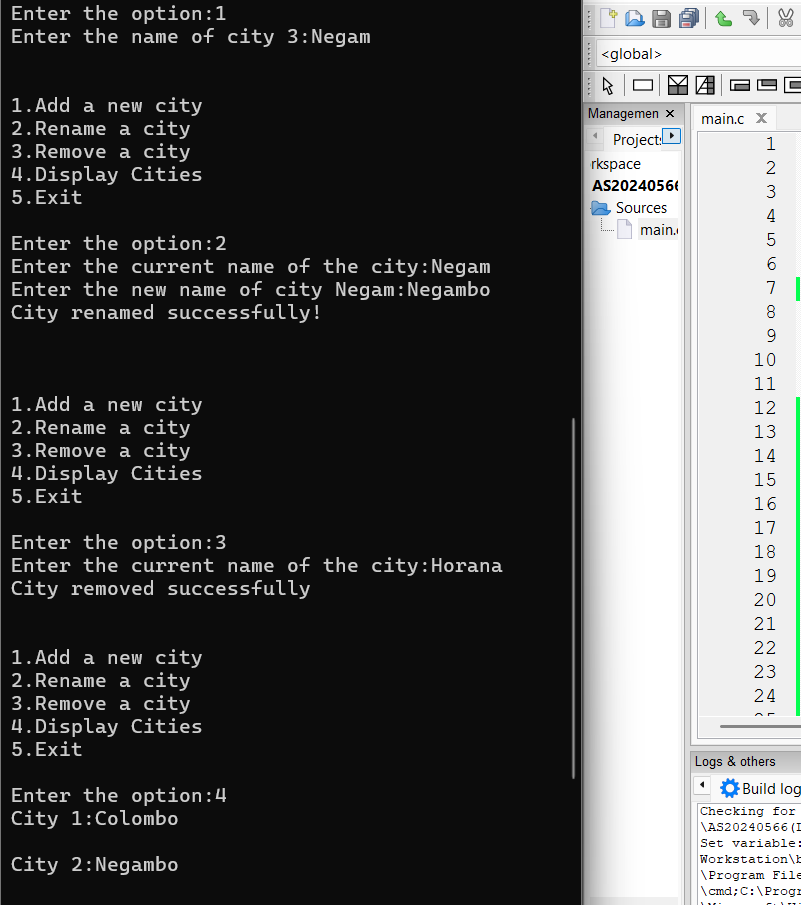
**5.2 Deliveries File** (deliveries.txt)

* Stores all delivery records.

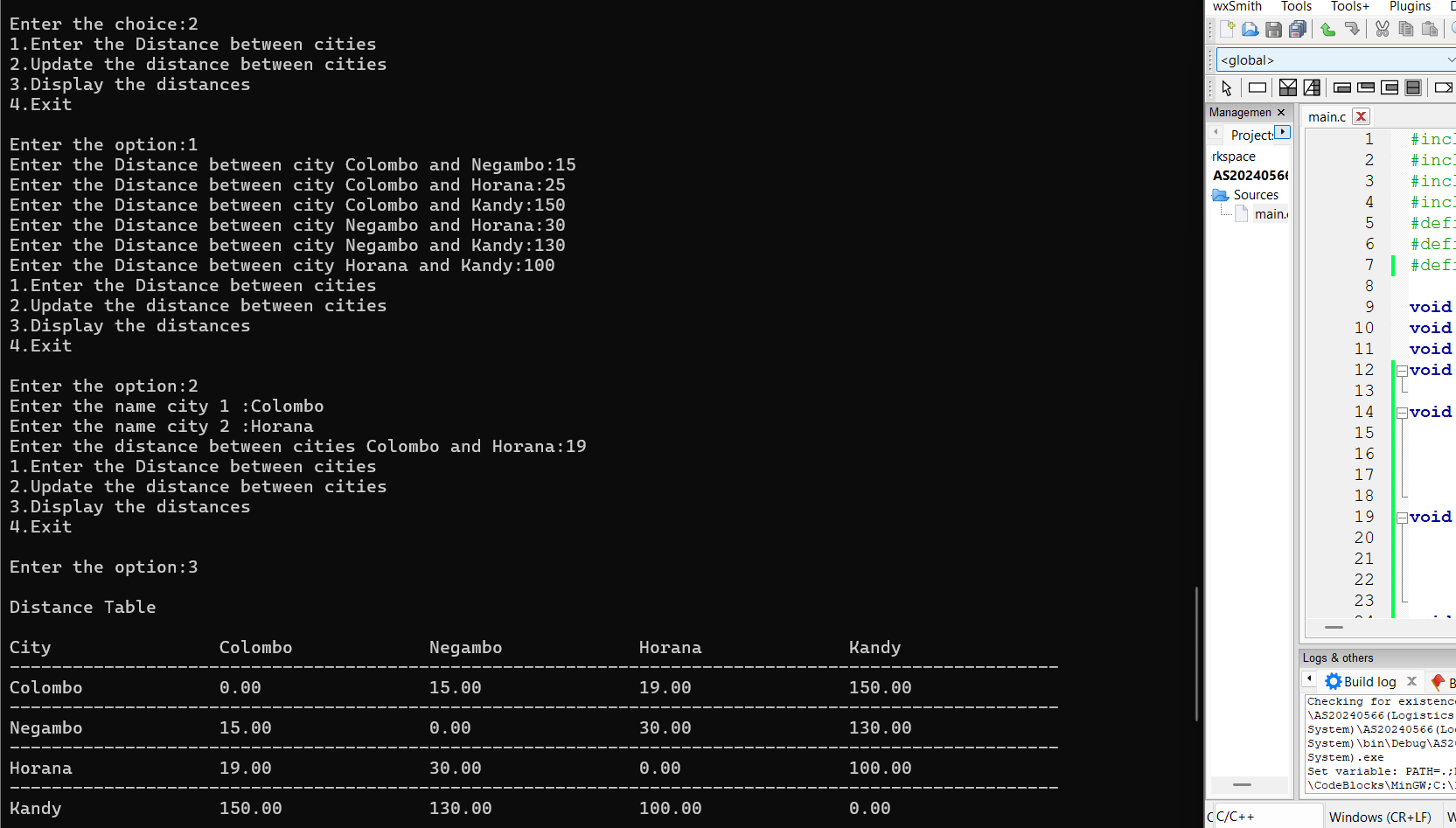
**6. Sample Outputs**

* **cityManage Function**

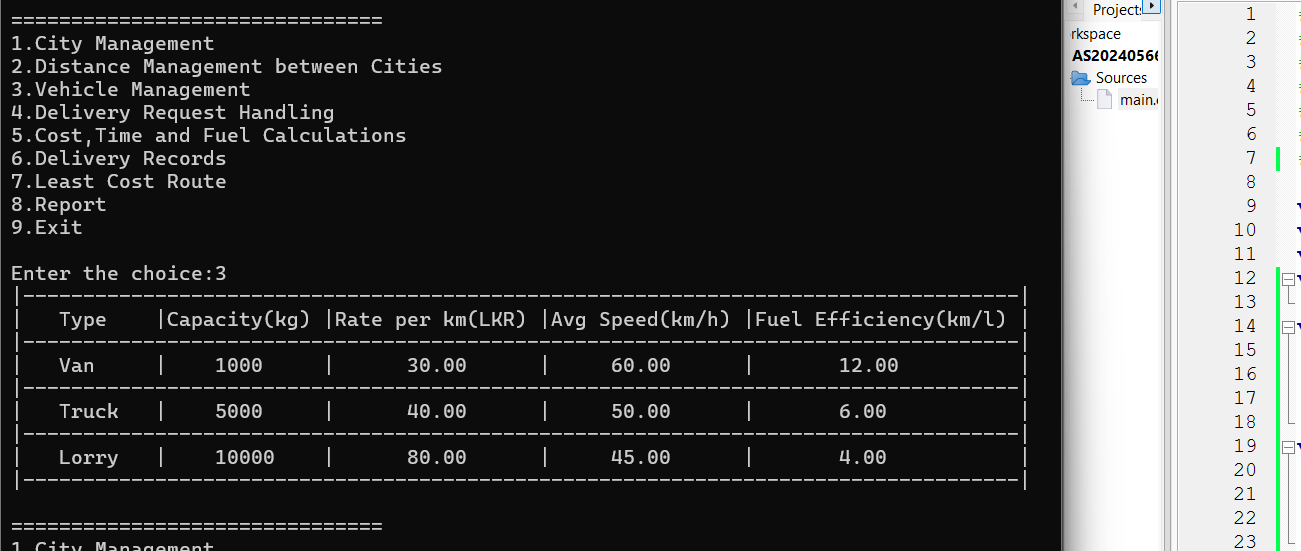
****

****

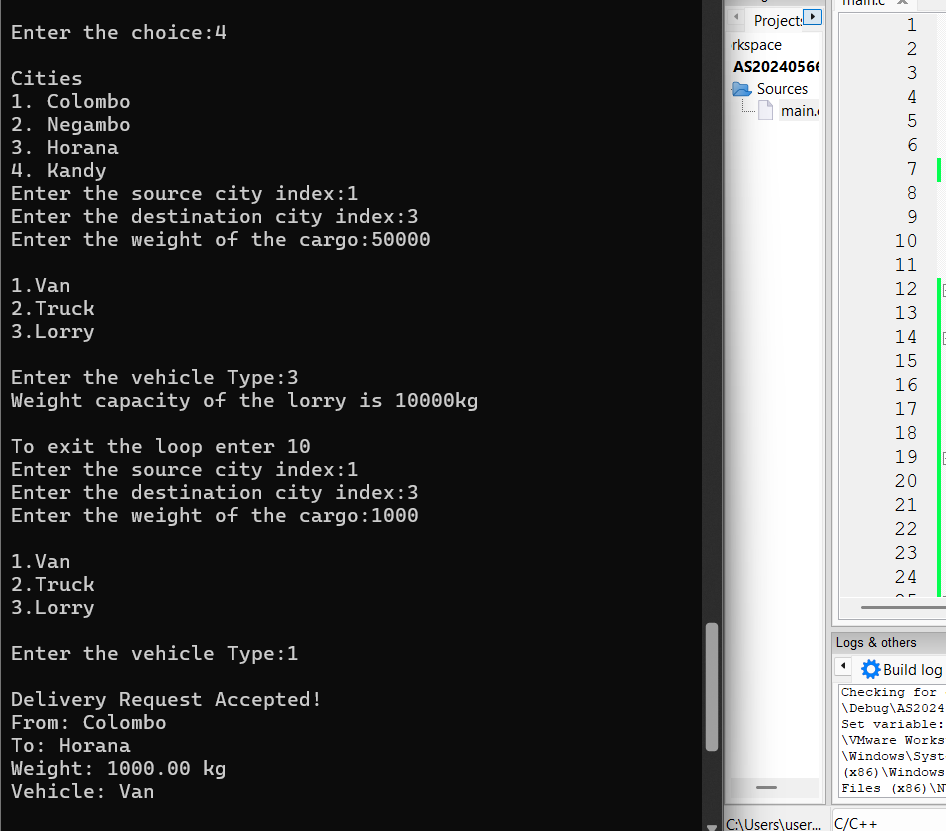
* **distanceManage Function**

****

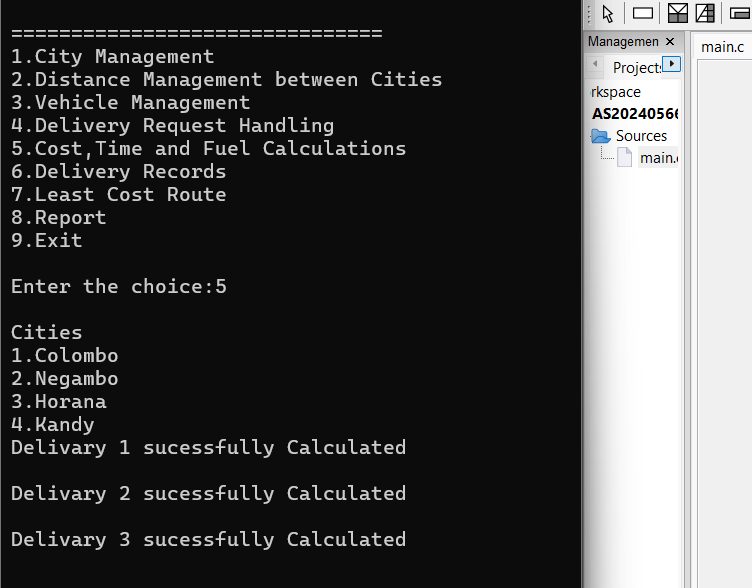
* **vehicleManage Function**

****

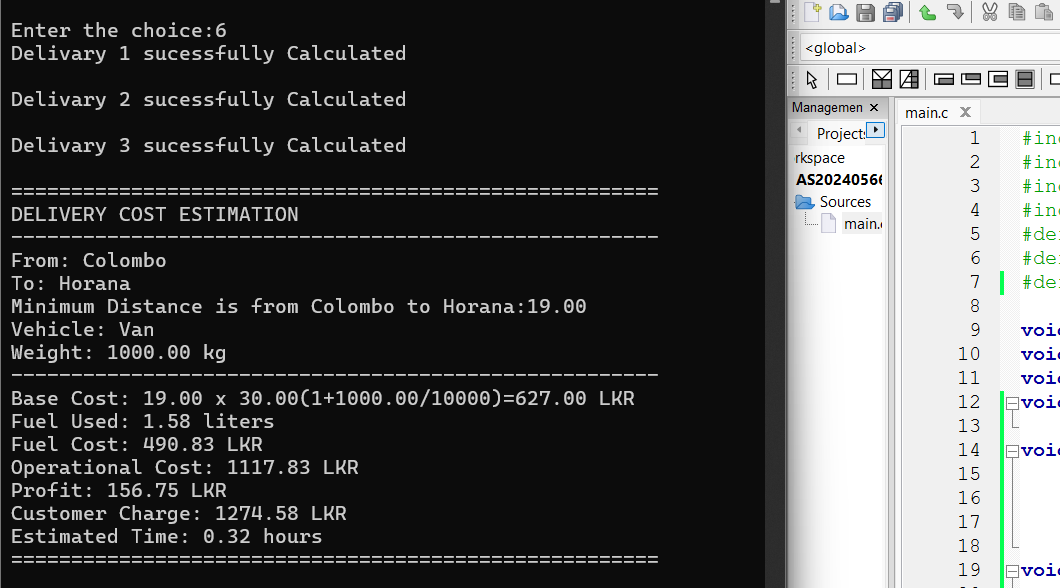
* **deliveryRequest Function**

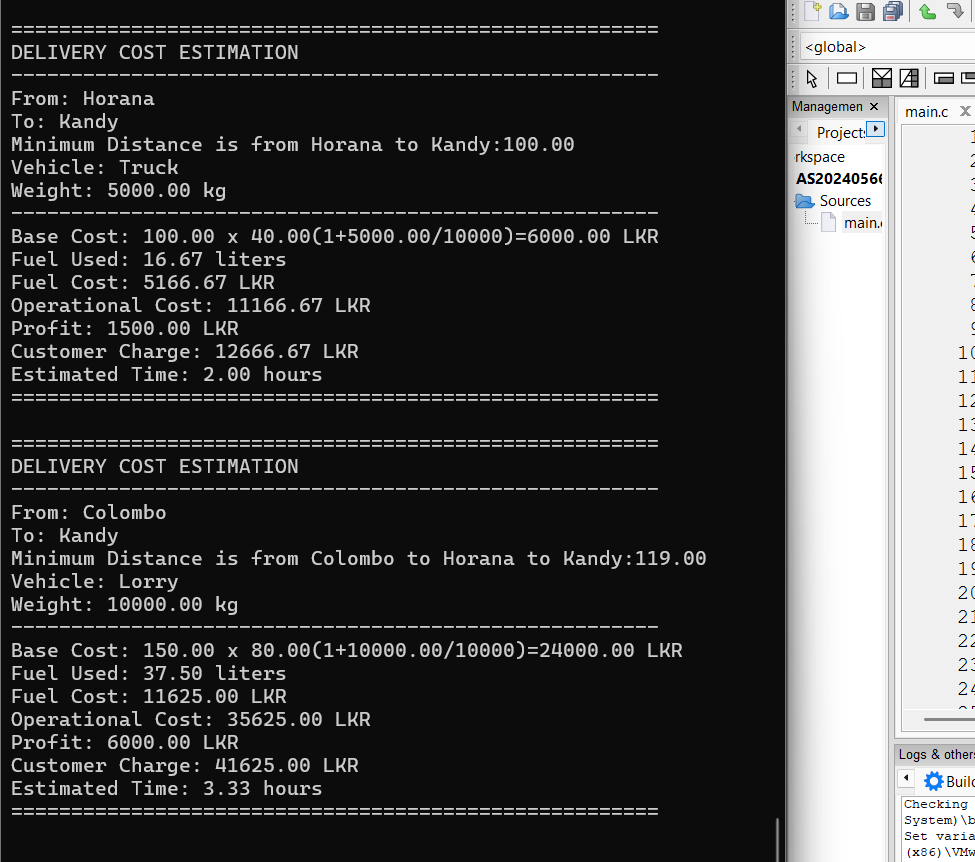
****

* **calcCTF Function**

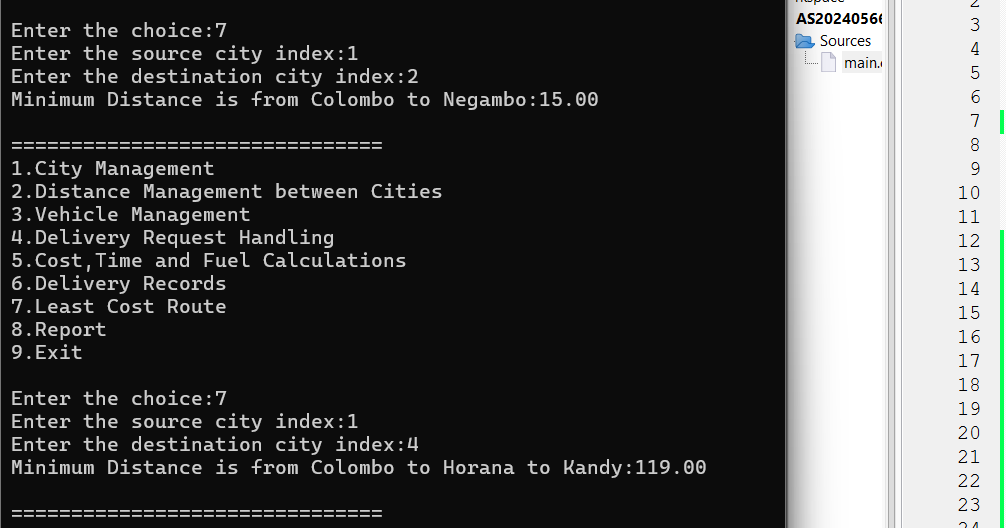
****

* **deliveryRecords Function**

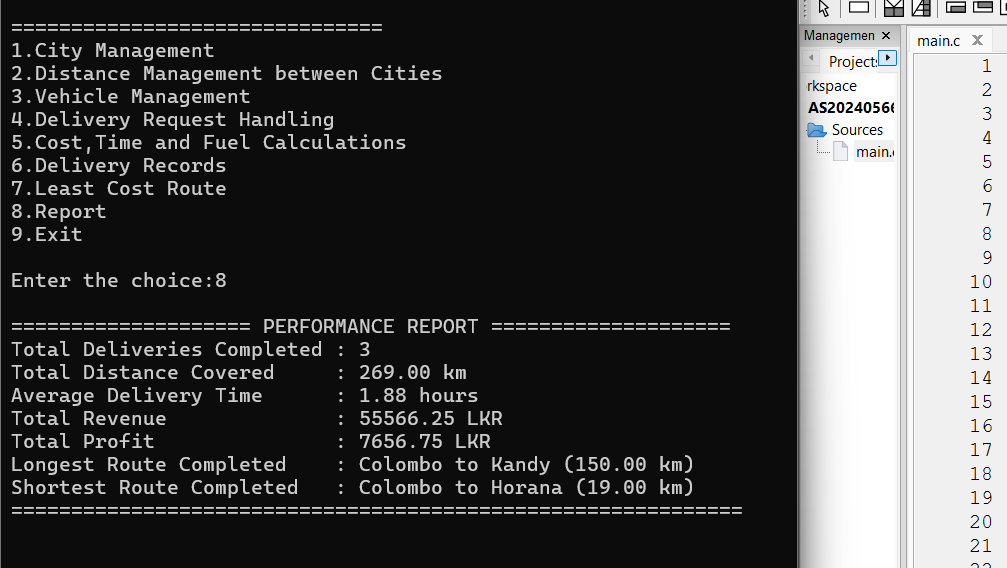
****

****

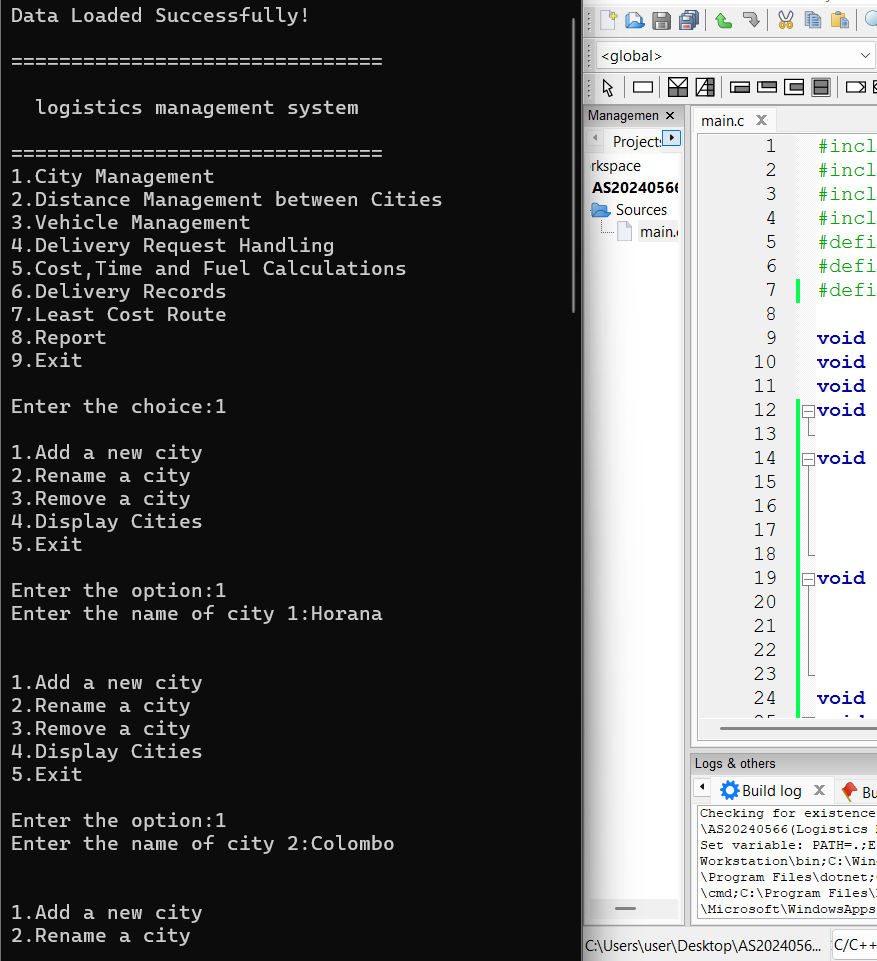
* **leastDistance Function**

****

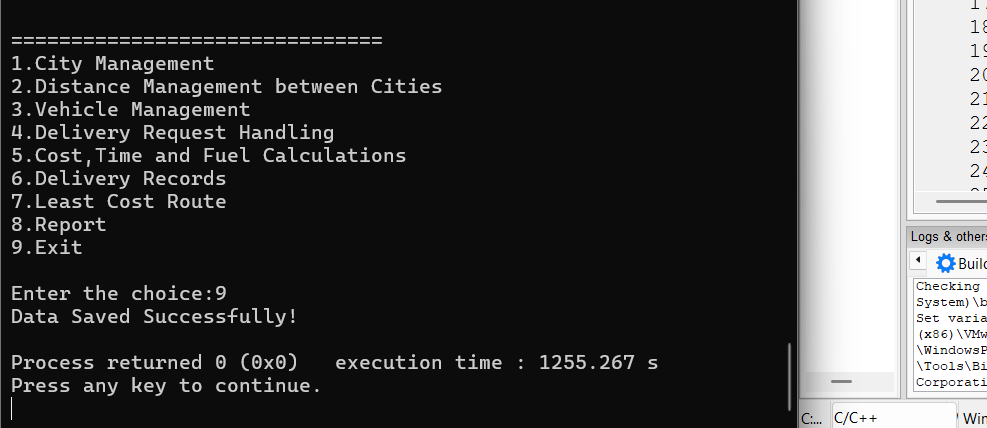
* **performanceReport Function**

****

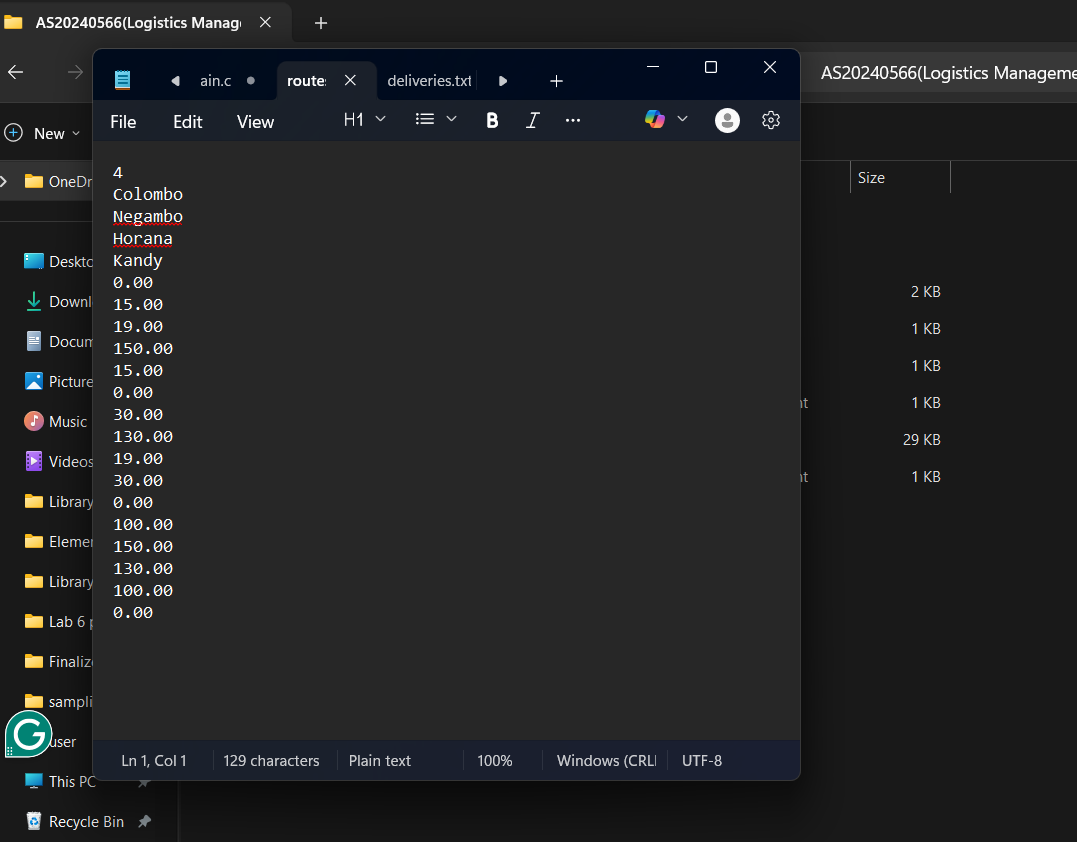
* **File Handling Functions**
* **Data Load**

****

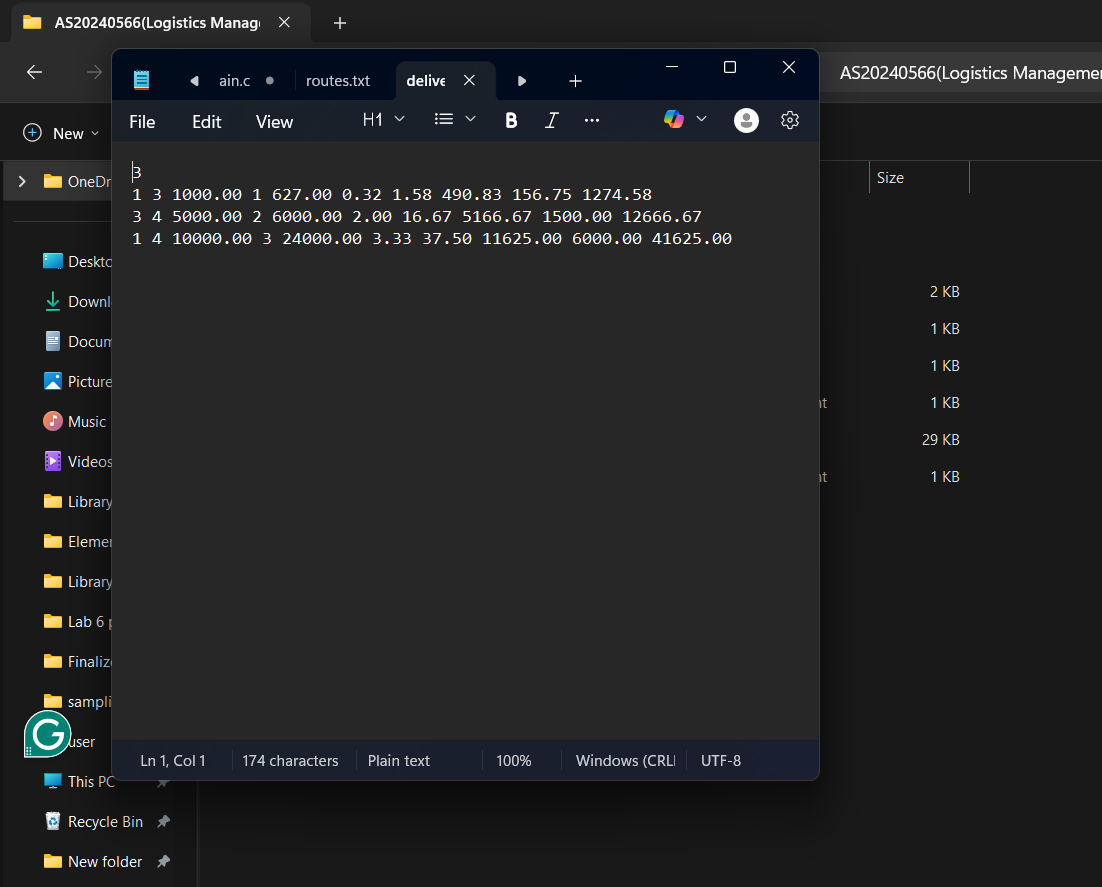
* **Data Saved**

****

* **route.txt**

****

* **deliveries.txt**

****

**7. Limitations**

* Maximum of 30 cities and 50 deliveries.
* Minimum distance calculation supports only on 2 to 4 cities.
* Vehicle types and rates are fixed.

**8. Conclusion**

The Logistics Management System is a program for managing cargo deliveries between different cities. It accurately calculates delivery costs, fuel consumption, and profits while maintaining detailed delivery records. It gives a final performance report of the overall deliveries and helps to save the data in .txt files. The system’s modular design and file persistence make it extendable for future enhancements.