CSC 1012

Introduction to Computer Programming Assignment (Individual)

Logistics Management System

Project Report

Prepared by : AS20240596

Github repository link :

https://github.com/Kalutara2004/1012-LogisticsManagementSystem1.git

Content

Topic Page No

Introduction

The Logistics Management System addresses all core requirements specified in the assignment brief, including city management, distance tracking, vehicle selection, delivery cost calculations, and performance reporting. By utilizing parallel arrays for data storage and maintaining a modular function-based architecture, the program achieves robust functionality while adhering to the constraints of basic C programming constructs. The implementation showcases the practical application of algorithmic thinking in solving logistics problems such as route optimization, cost estimation, and resource management.

Functions

- Functions and their uses.

1) Main Function

Purpose : Entry point of the program that initializes the system and runs the main menu loop.

Parameters : None

Return Type : int

Functionality :

- Calls initializeSystem() to set up initial data.
- Displays main menu continuously using a do-while loop.
- Handles user input validation using clearInputBuffer().
- Routes user to appropriate modules based on menu selection.
- Provides clean exit when user chooses option 7.

2) System Initialization

- initializeSystem

Purpose : Sets up initial system data including default cities and distance

Parameters :

- int* cityCount - pointer to store number of cities
- char cities[][MAX_NAME_LENGTH] - 2D array for city names
- int distance[][MAX_CITIES] - 2D array for distance matrix
- int* deliveryCount - pointer to store number of deliveries

Return Type : void

Functionality :

- Initializes distance matrix with 0 for same city, -1 for no connection
- Adds 4 default cities: Colombo, Kandy, Galle, Jaffna

- Sets up predefined distances between default cities
- Sets delivery count to 0
- Displays initialization message

3) Menu Display Functions

➢ displayMainMenu

Purpose : Shows the main navigation menu
Parameters : None
Return Type : void
Functionality :

- Displays formatted main menu with 7 options
- Uses visual separators for better readability

➢ displayCities

Purpose : Shows list of all available cities
Parameters :

- int cityCount - number of cities
- char cities[][MAX_NAME_LENGTH] - array of city names

Return Type : void
Functionality :

- Lists all cities with their indices
- Handles case when no cities are available

4) City Management Module

➢ cityManagement

Purpose : Main controller for city-related operations
Parameters :

- int* cityCount - pointer to city count
- char cities[][MAX_NAME_LENGTH] - city names array
- int distance[][MAX_CITIES] - distance matrix
- int deliveryCount - current delivery count
- char deliverySource/Destination[][] - delivery records

Return Type : void
Functionality :

- Displays city management submenu
- Routes to add, rename, remove, or display cities
- Returns to main menu when done

➢ addCity

Purpose : Adds a new city to the system
Parameters :

- int* cityCount - pointer to city count
- char cities[][MAX_NAME_LENGTH] - city names array
- int distance[][MAX_CITIES] - distance matrix

Return Type : void
Functionality :

- Checks if maximum city limit reached
- Validates city name doesn't already exist
- Updates distance matrix with new city
- Increments city count

➢ renameCity

Purpose : Changes the name of an existing city
Parameters :

- int cityCount - current city count
- char cities[][MAX_NAME_LENGTH] - city names array
- int deliveryCount - current delivery count
- char deliverySource/Destination[][] - delivery records

Return Type : void
Functionality :

- Validates city index exists
- Ensures new name doesn't conflict with existing cities
- Updates all delivery records that reference the renamed city

➢ removeCity

Purpose : Removes a city from the system
Parameters :

- int* cityCount - pointer to city count
- char cities[][MAX_NAME_LENGTH] - city names array
- int distance[][MAX_CITIES] - distance matrix
- int deliveryCount - current delivery count
- char deliverySource/Destination[][] - delivery records

Return Type : void
Functionality :

- Checks if city has existing deliveries
- Requests confirmation if deliveries exist
- Removes city from cities array and distance matrix
- Updates delivery records
- Decrements city count

5) Distance Management Module

➢ distanceManagement

Purpose : Main controller for distance-related operations
Parameters :

- int cityCount - number of cities
- char cities[][MAX_NAME_LENGTH] - city names array
- int distance[][MAX_CITIES] - distance matrix

Return Type : void
Functionality :

- Provides submenu for distance operations
- Routes to input distance or display distance table

➢ inputDistance

Purpose : Sets or updates distance between two cities
Parameters :

- int cityCount - number of cities
- char cities[][MAX_NAME_LENGTH] - city names array
- int distance[][MAX_CITIES] - distance matrix

Return Type : int (0 for success, -1 for failure)
Functionality :

- Validates city indices
- Prevents setting distance from city to itself
- Ensures distance is non-negative
- Sets symmetrical distances (A→B = B→A)

➢ displayDistanceTable

Purpose : Shows formatted distance matrix
Parameters :

- int cityCount - number of cities
- char cities[][MAX_NAME_LENGTH] - city names array
- int distance[][MAX_CITIES] - distance matrix

Return Type : void
Functionality :

- Displays matrix with city headers
- Shows "N/A" for no connection
- Formats output in a readable table

6) Vehicle Management Module

➢ vehicleManagement

Purpose : Displays available vehicle types and specifications
Parameters : None
Return Type : void
Functionality :

- Shows formatted table of 3 vehicle types
- Displays capacity, rate, speed, and efficiency
- Uses fixed global arrays for vehicle data

7) Delivery Management Module

➢ deliveryRequest

Purpose : Handles new delivery order processing
Parameters :

- Multiple arrays for delivery data storage

- City and distance data
- Delivery count pointer

Return Type : void
Functionality :

- Validates sufficient cities exist
- Checks delivery storage capacity
- Gets source, destination, vehicle type, and weight
- Validates all inputs
- Finds optimal route and calculates costs
- Saves delivery record

➢ calculateDeliveryCost

Purpose : Calculates and displays detailed delivery cost breakdown
Parameters :

- Source/destination cities, vehicle type, weight, distance
- Vehicle specifications arrays

Return Type : float (customer charge amount)
Functionality :

- Calculates base cost, fuel cost, operational cost
- Applies 25% profit margin
- Displays formatted cost estimation report
- Returns final customer charge

➢ saveDeliveryRecord

Purpose : Stores delivery information in system arrays
Parameters :

- Multiple arrays for delivery data storage
- City and vehicle data

- Delivery count pointer

Return Type : void
Functionality :

- Assigns unique delivery ID
- Stores all delivery parameters
- Calculates and stores financial data
- Increments delivery count

➢ findMinimumDistance

Purpose : Finds shortest path between two cities (currently basic implementation)
Parameters :

- Source/destination indices
- Path array and length pointer
- City count and distance matrix

Return Type : float (minimum distance or -1 if no route)
Functionality :

- Currently returns direct distance if available
- Placeholder for future route optimization algorithm

7) Reporting Module

➢ reports

Purpose : Generates performance and statistical reports
Parameters :

- Delivery count and various delivery data arrays

Return Type : void
Functionality :

- Calculates totals: distance, time, revenue, profit

- Finds longest and shortest routes
- Computes averages
- Displays formatted performance metrics

8)File Handling Module

➢ fileHandling

Purpose : Manages data persistence through file operations
Parameters :

- City count, cities array, distance matrix

Return Type : void
Functionality :

- Saves city list and distance matrix to "routes.txt"
- Uses formatted text file storage
- Handles file operation errors

9) Utility Functions

➢ clearInputBuffer

Purpose : Clears standard input buffer to prevent input issues
Parameters : None
Return Type : void
Functionality :

- Removes leftover characters from input stream
- Prevents scanf issues in subsequent inputs

➢ updateDeliveryRecordsAfterCityRemoval

Purpose : Handles delivery record updates when city is removed

Parameters :

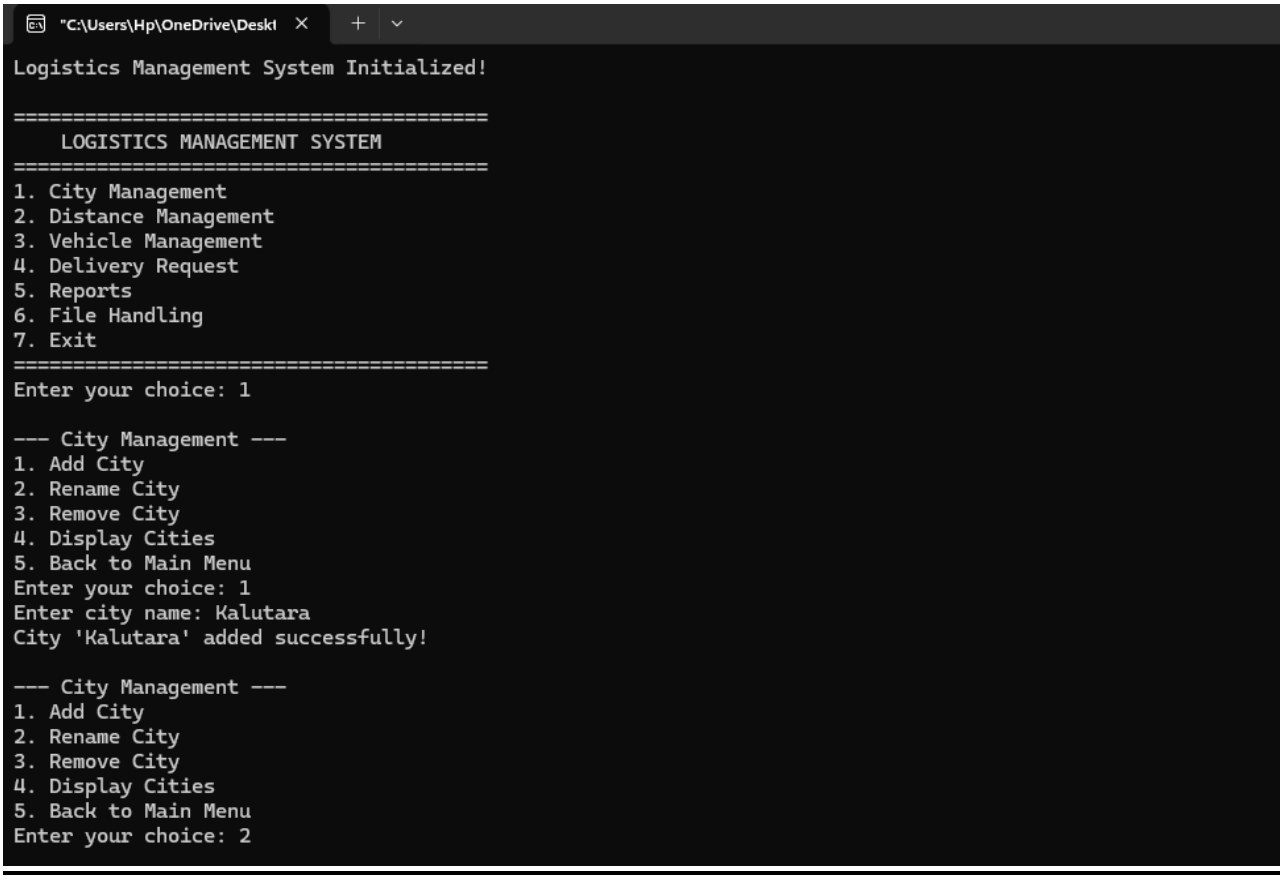
- Removed city index and name
- Delivery count and record arrays

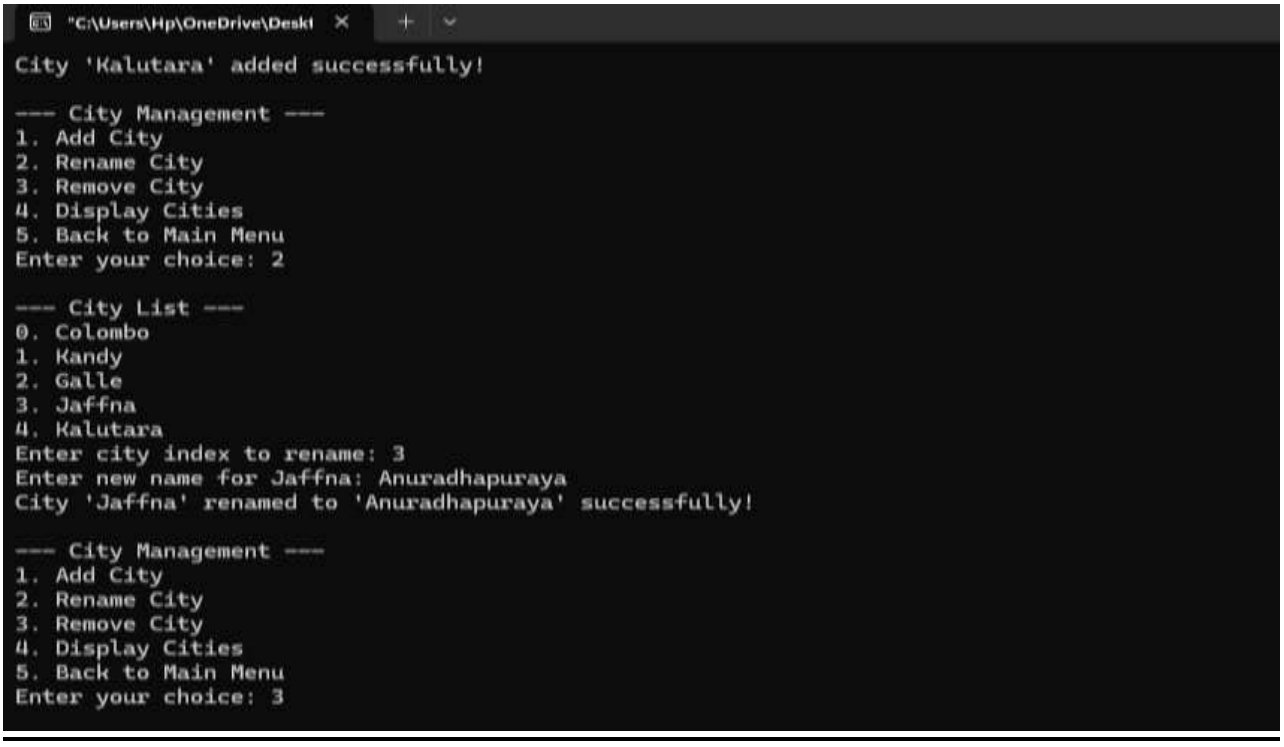
Return Type : void

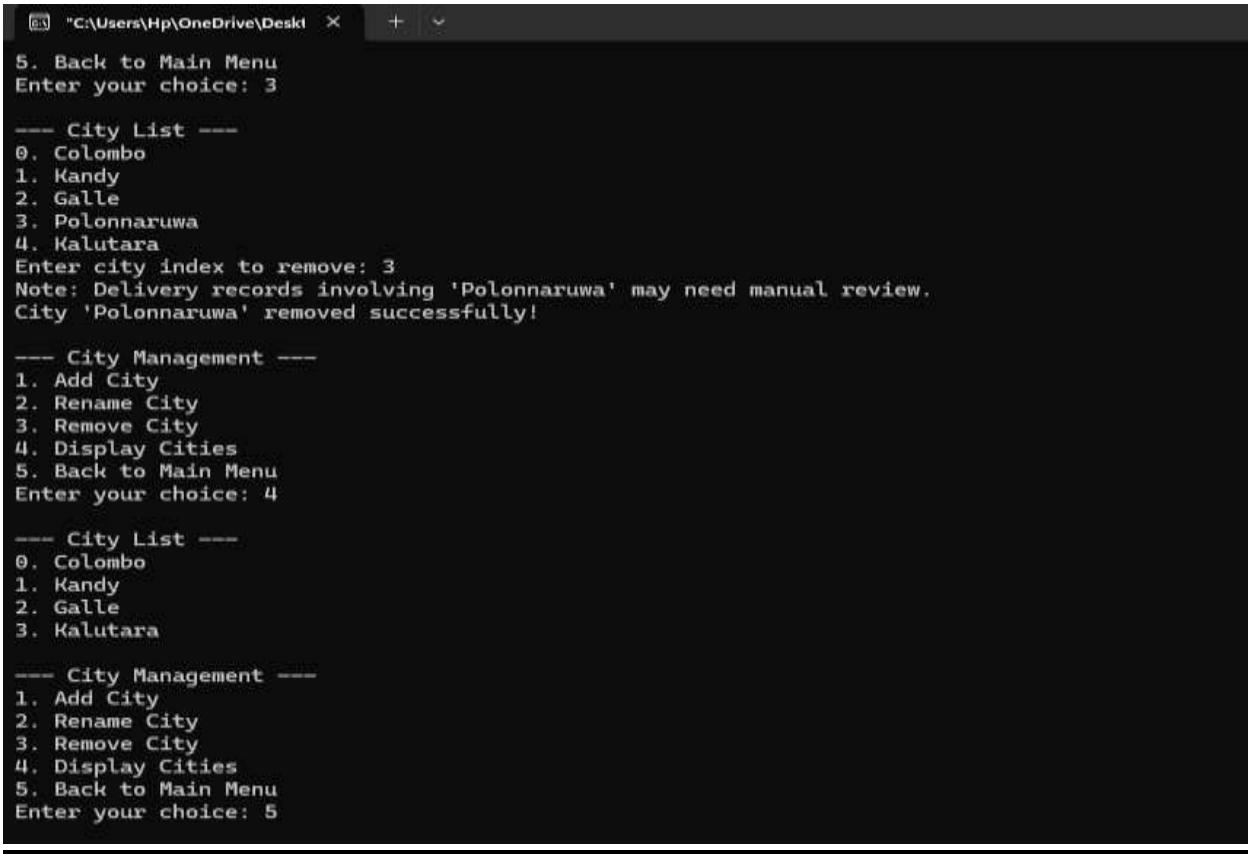
Functionality :

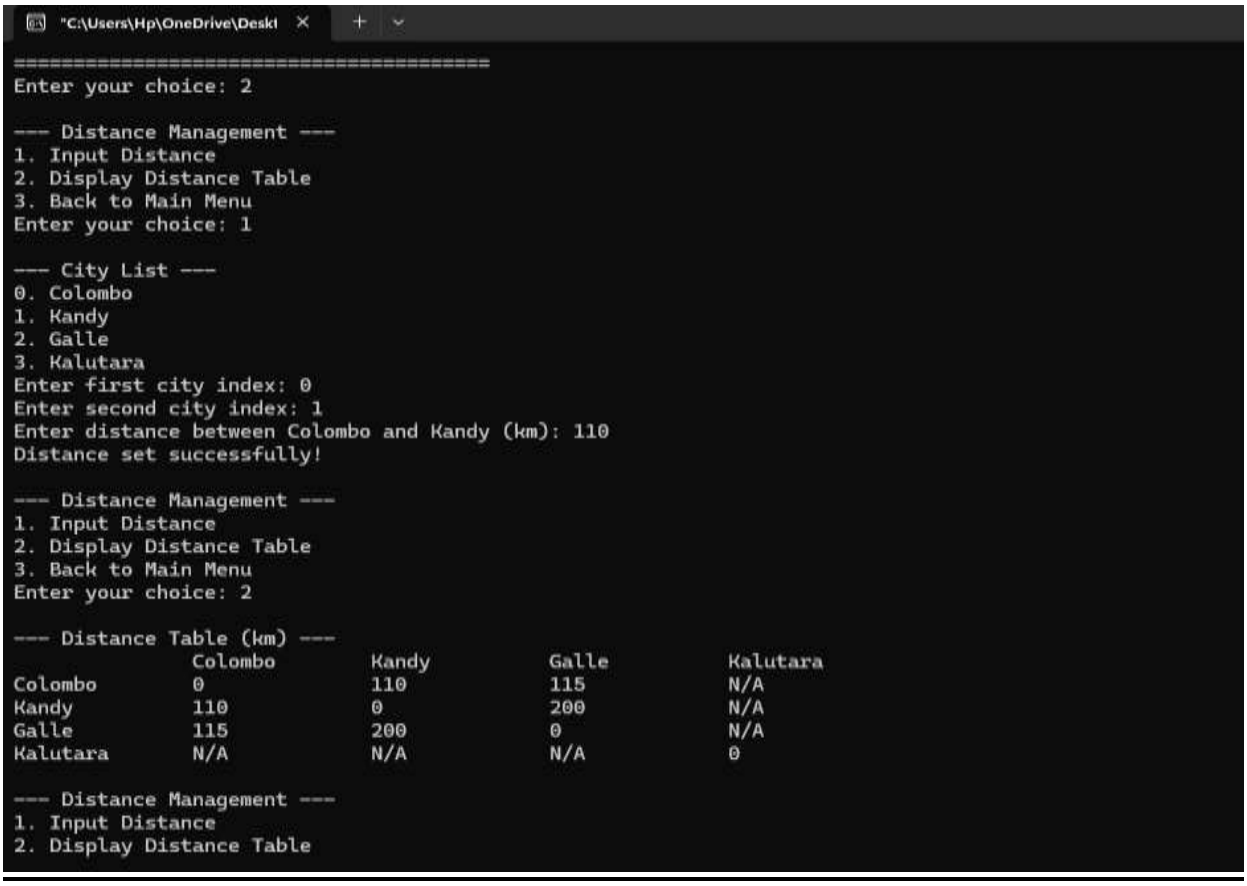
- Currently provides warning message
- Placeholder for more sophisticated record updating

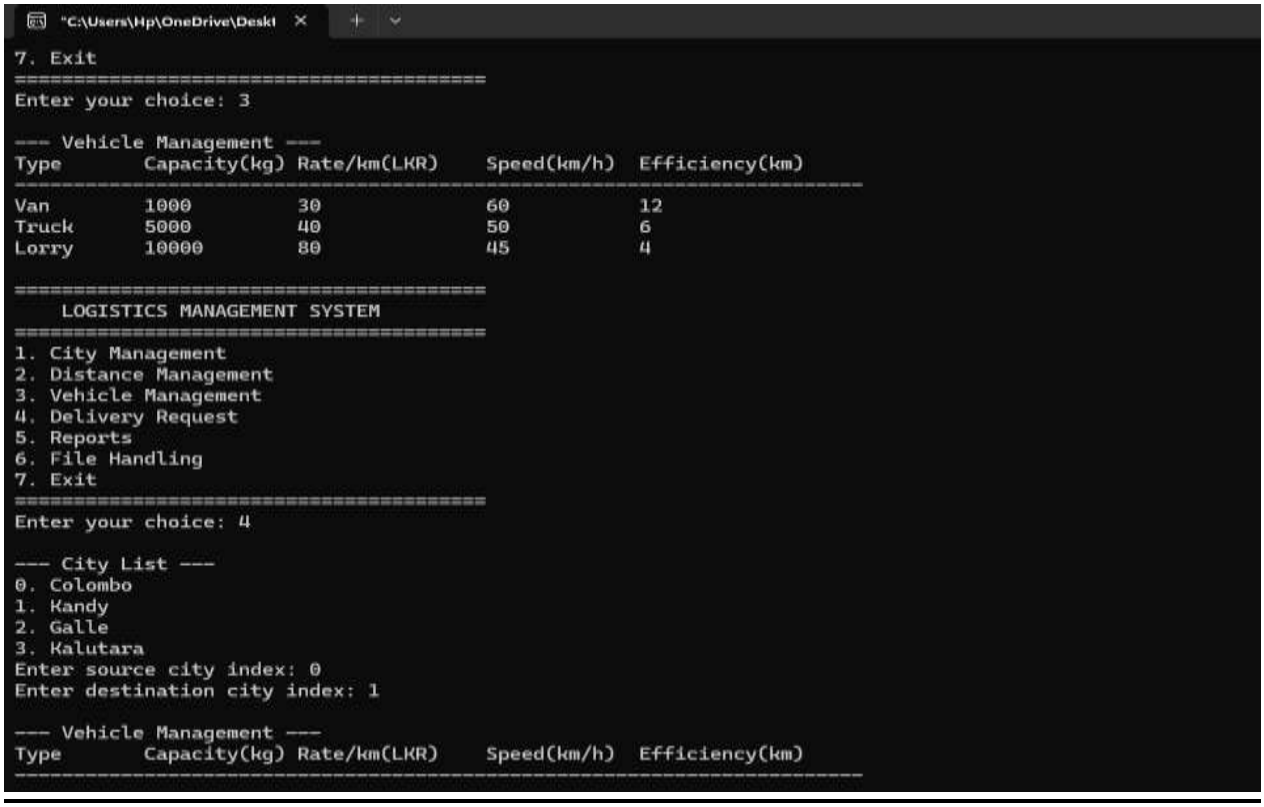
Features

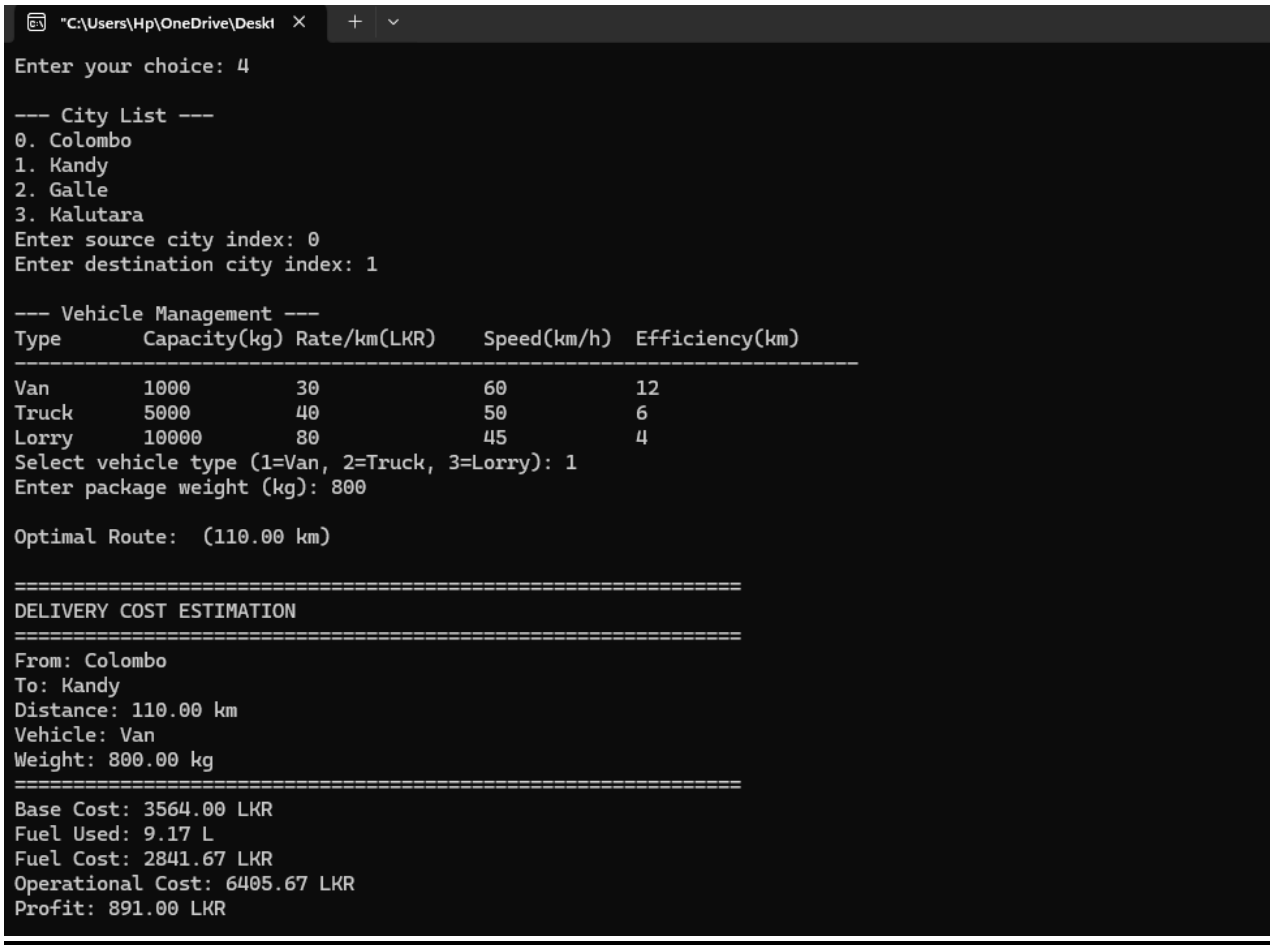

"C:\Users\Hp\OneDrive\Deskt
Logistics Management System Initialized!
==
LOGISTICS MANAGEMENT SYSTEM
==
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
==
Enter your choice: 1
--- City Management ---
1. Add City
2. Rename City
3. Remove City
4. Display Cities
5. Back to Main Menu
Enter your choice: 1
Enter city name: Kalutara
City 'Kalutara' added successfully!
--- City Management ---
1. Add City
2. Rename City
3. Remove City
4. Display Cities
5. Back to Main Menu
Enter your choice: 2

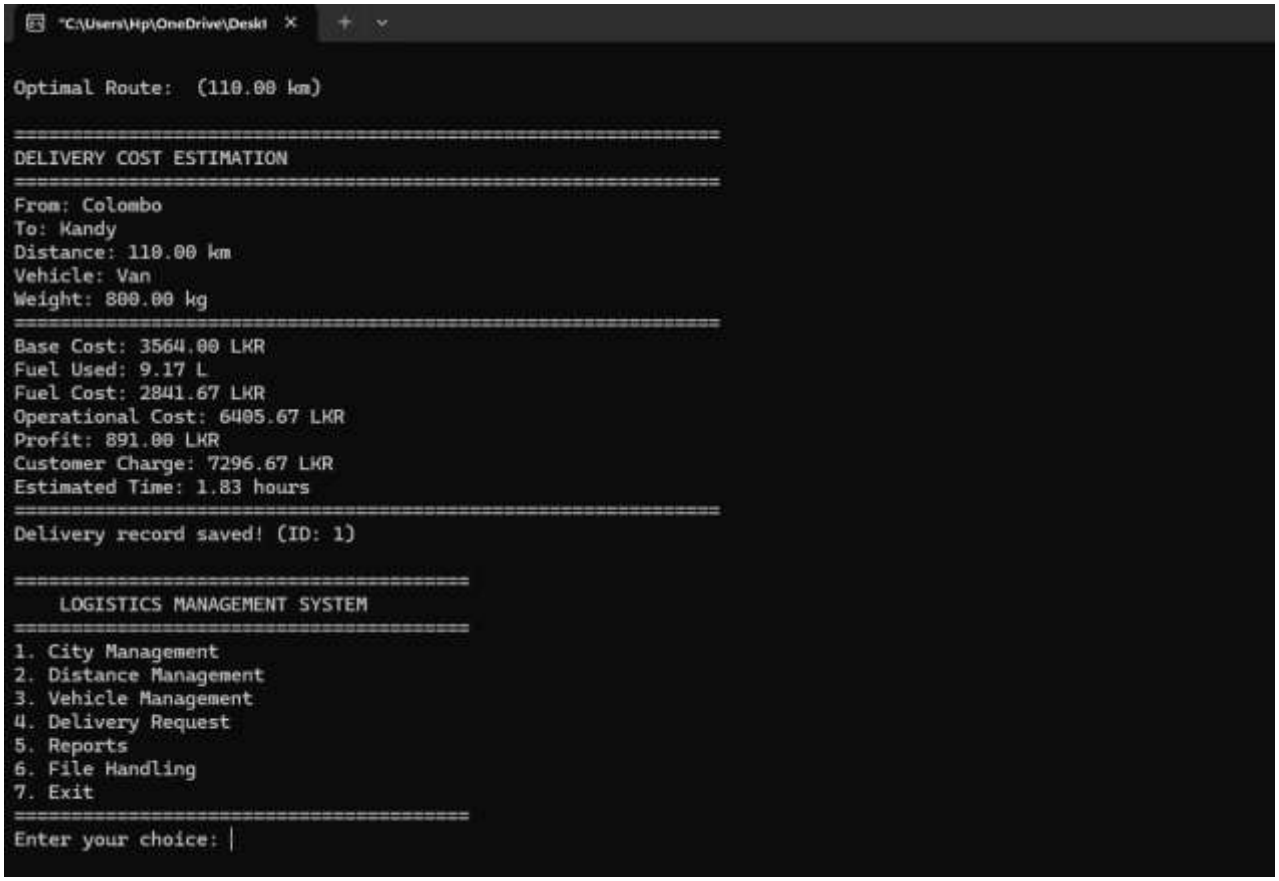

"C:\Users\Hp\OneDrive\Deskt
City 'Kalutara' added successfully!
--- City Management ---
1. Add City
2. Rename City
3. Remove City
4. Display Cities
5. Back to Main Menu
Enter your choice: 2
--- City List ---
0. Colombo
1. Kandy
2. Galle
3. Jaffna
4. Kalutara
Enter city index to rename: 3
Enter new name for Jaffna: Anuradhapuraya
City 'Jaffna' renamed to 'Anuradhapuraya' successfully!
--- City Management ---
1. Add City
2. Rename City
3. Remove City
4. Display Cities
5. Back to Main Menu
Enter your choice: 3

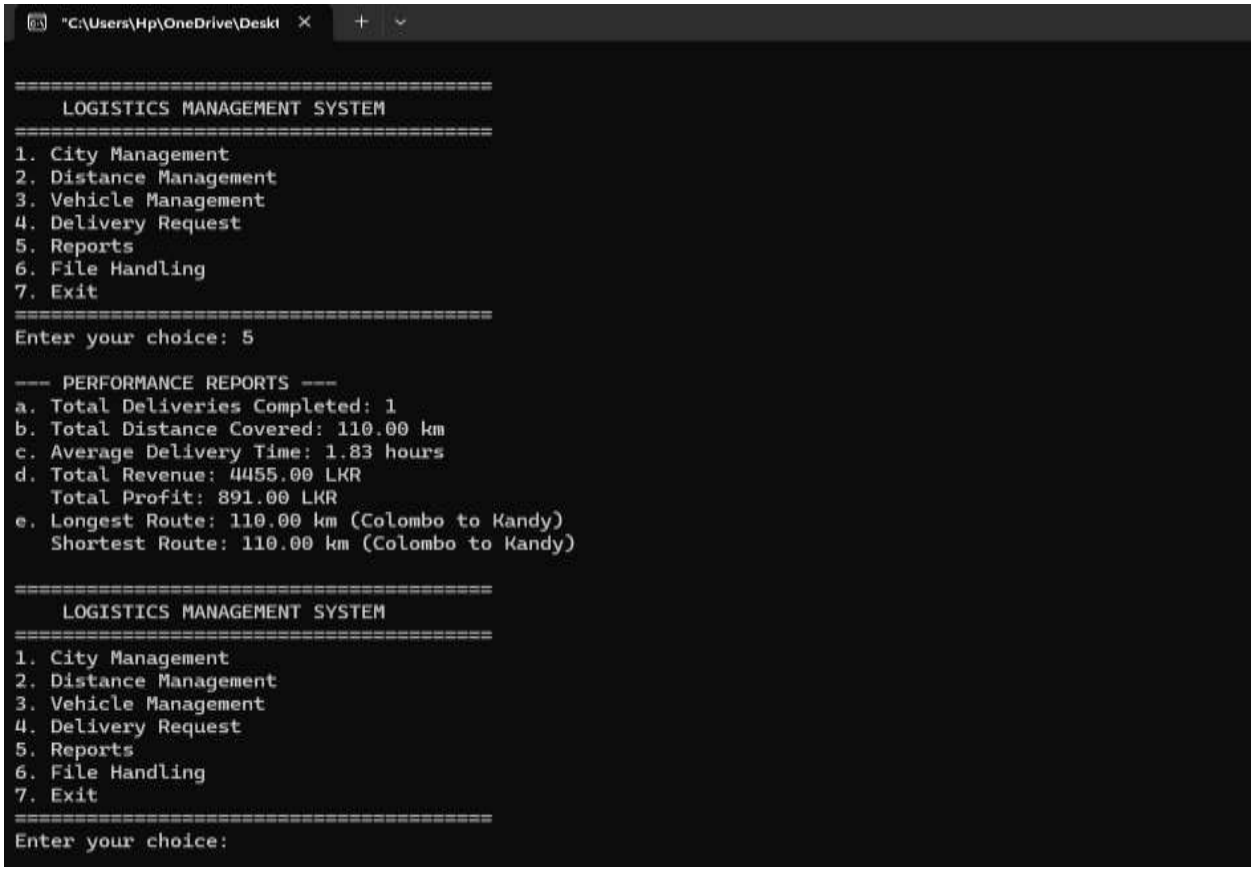

"C:\Users\Hp\OneDrive\Deskt
5. Back to Main Menu
Enter your choice: 3
--- City List ---
0. Colombo
1. Kandy
2. Galle
3. Polonnaruwa
4. Kalutara
Enter city index to remove: 3
Note: Delivery records involving 'Polonnaruwa' may need manual review.
City 'Polonnaruwa' removed successfully!
--- City Management ---
1. Add City
2. Rename City
3. Remove City
4. Display Cities
5. Back to Main Menu
Enter your choice: 4
--- City List ---
0. Colombo
1. Kandy
2. Galle
3. Kalutara
--- City Management ---
1. Add City
2. Rename City
3. Remove City
4. Display Cities
5. Back to Main Menu
Enter your choice: 5

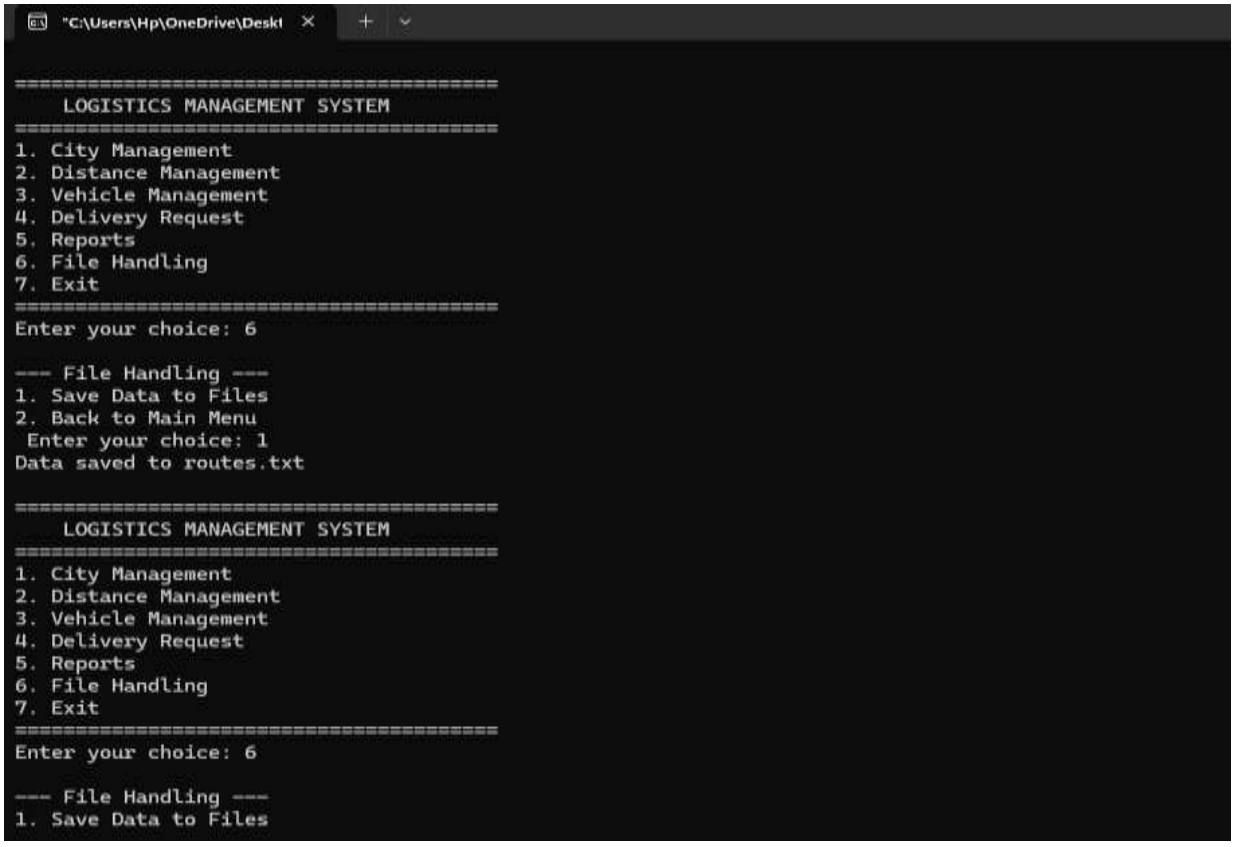

"C:\Users\Hp\OneDrive\Deskt
==========
Enter your choice: 2
--- Distance Management ---
1. Input Distance
2. Display Distance Table
3. Back to Main Menu
Enter your choice: 1
--- City List ---
0. Colombo
1. Kandy
2. Galle
3. Kalutara
Enter first city index: 0
Enter second city index: 1
Enter distance between Colombo and Kandy (km): 110
Distance set successfully!
--- Distance Management ---
1. Input Distance
2. Display Distance Table
3. Back to Main Menu
Enter your choice: 2
--- Distance Table (km) ---
Colombo Kandy Galle Kalutara
Colombo 0 110 115 N/A
Kandy 110 0 200 N/A
Galle 115 200 0 N/A
Kalutara N/A N/A N/A 0
--- Distance Management ---
1. Input Distance
2. Display Distance Table

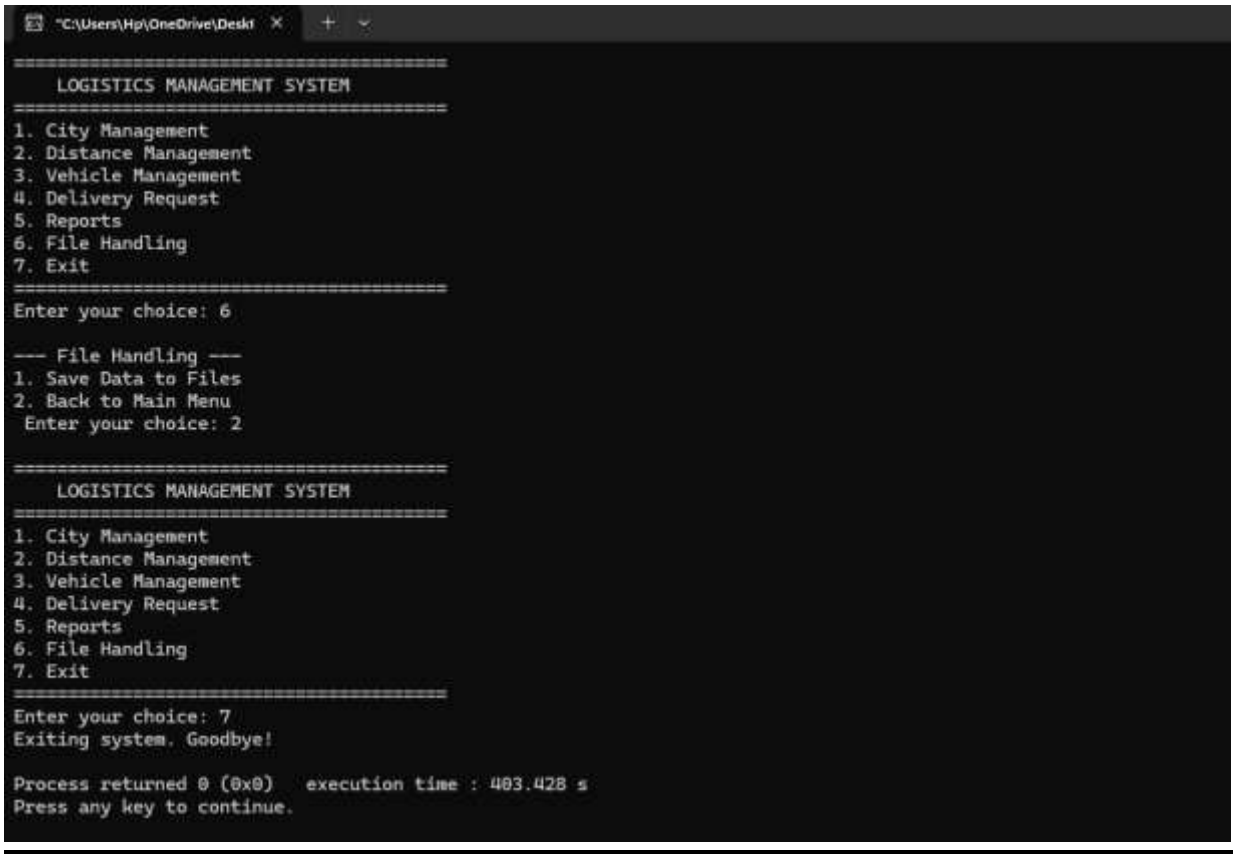

"C:\Users\Hp\OneDrive\Deskt
7. Exit
Enter your choice: 3
--- Vehicle Management ---
Type Capacity(kg) Rate/km(LKR) Speed(km/h) Efficiency(km)
Van 1000 30 60 12
Truck 5000 40 50 6
Lorry 10000 80 45 4
LOGISTICS MANAGEMENT SYSTEM
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
Enter your choice: 4
--- City List ---
0. Colombo
1. Kandy
2. Galle
3. Kalutara
Enter source city index: 0
Enter destination city index: 1
--- Vehicle Management ---
Type Capacity(kg) Rate/km(LKR) Speed(km/h) Efficiency(km)


"C:\Users\Hp\OneDrive\Deskt
Enter your choice: 4
--- City List ---
0. Colombo
1. Kandy
2. Galle
3. Kalutara
Enter source city index: 0
Enter destination city index: 1
--- Vehicle Management ---
Type Capacity(kg) Rate/km(LKR) Speed(km/h) Efficiency(km)
Van 1000 30 60 12
Truck 5000 40 50 6
Lorry 10000 80 45 4
Select vehicle type (1=Van, 2=Truck, 3=Lorry): 1
Enter package weight (kg): 800
Optimal Route: (110.00 km)
DELIVERY COST ESTIMATION
From: Colombo
To: Kandy
Distance: 110.00 km
Vehicle: Van
Weight: 800.00 kg
Base Cost: 3564.00 LKR
Fuel Used: 9.17 L
Fuel Cost: 2841.67 LKR
Operational Cost: 6405.67 LKR
Profit: 891.00 LKR


"C:\Users\Hp\OneDrive\Deskt
Optimal Route: (110.00 km)
DELIVERY COST ESTIMATION
From: Colombo
To: Kandy
Distance: 110.00 km
Vehicle: Van
Weight: 800.00 kg
Base Cost: 3564.00 LKR
Fuel Used: 9.17 L
Fuel Cost: 2841.67 LKR
Operational Cost: 6405.67 LKR
Profit: 891.00 LKR
Customer Charge: 7296.67 LKR
Estimated Time: 1.83 hours
Delivery record saved! (ID: 1)
LOGISTICS MANAGEMENT SYSTEM
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
Enter your choice:


"C:\Users\Hp\OneDrive\Deskt
LOGISTICS MANAGEMENT SYSTEM
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
Enter your choice: 5
--- PERFORMANCE REPORTS ---
a. Total Deliveries Completed: 1
b. Total Distance Covered: 110.00 km
c. Average Delivery Time: 1.83 hours
d. Total Revenue: 4455.00 LKR
Total Profit: 891.00 LKR
e. Longest Route: 110.00 km (Colombo to Kandy)
Shortest Route: 110.00 km (Colombo to Kandy)
LOGISTICS MANAGEMENT SYSTEM
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
Enter your choice:


"C:\Users\Hp\OneDrive\Deskt
LOGISTICS MANAGEMENT SYSTEM
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
Enter your choice: 6
--- File Handling ---
1. Save Data to Files
2. Back to Main Menu
Enter your choice: 1
Data saved to routes.txt
LOGISTICS MANAGEMENT SYSTEM
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
Enter your choice: 6
--- File Handling ---
1. Save Data to Files


"C:\Users\Hp\OneDrive\Deskt
LOGISTICS MANAGEMENT SYSTEM
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
Enter your choice: 6
--- File Handling ---
1. Save Data to Files
2. Back to Main Menu
Enter your choice: 2
LOGISTICS MANAGEMENT SYSTEM
1. City Management
2. Distance Management
3. Vehicle Management
4. Delivery Request
5. Reports
6. File Handling
7. Exit
Enter your choice: 7
Exiting system. Goodbye!
Process returned 0 (0x0) execution time : 403.428 s
Press any key to continue.

Conclusion

The Logistics Management System successfully demonstrates the practical application of core C programming concepts to solve real-world business problems. Through the implementation of this comprehensive system, we have shown how arrays, functions, and modular programming can be effectively combined to create a robust and functional application without relying on advanced data structures or object-oriented principles.

The system meets all specified requirements, providing complete functionality for city management, distance tracking, vehicle selection, delivery processing, cost calculations, and performance reporting. The modular design ensures code maintainability and readability, with each function serving a distinct and well-defined purpose. The use of parallel arrays for data storage, while more cumbersome than structures, effectively demonstrates how complex data relationships can be managed using fundamental programming constructs.

Key achievements of this implementation include:

- Comprehensive input validation and error handling
- Efficient financial calculations with detailed cost breakdowns
- User-friendly menu-driven interface
- Basic file persistence capabilities
- Performance reporting and analytics

While there are opportunities for enhancement, particularly in route optimization algorithms and more sophisticated data persistence, the current implementation provides a solid foundation that fulfills the assignment requirements. This project serves as an excellent example of how systematic problem-solving and structured programming can yield practical, functional software solutions using fundamental computer science principles.