**Logistics Management System – Project Report**

**Student Name:**

**M.M.S.K.Dhananjana**

**Student ID:**

**AS20240448**

**GitHub Repository:**

https://github.com/Savidya448/Logistics-Management-System.git

**📘 Table of Contents**

1. Introduction
2. System Features
3. Function Descriptions
4. Code Structure
5. GitHub Repository
6. Conclusion

**1. 📖 Introduction**

This project is a menu-driven **Logistics Management System** implemented in the **C programming language**. It simulates basic delivery logistics including city and distance management, vehicle handling, order estimation, and performance reporting. The goal is to apply fundamental programming concepts such as arrays, loops, functions, and conditionals in a real-world application.

**2. ⚙️ System Features Implemented**

* **City Management**: Add and list cities (up to 30).
* **Distance Management**: Store/edit distances in a 2D array; symmetric and no self-distance.
* **Vehicle Management**: Fixed 3 vehicle types with unique speed, efficiency, and cost per km.
* **Delivery Request Handling**:
  + Input: source, destination, weight, and vehicle type.
  + Validates city index, non-zero distance, and vehicle capacity.
* **Cost, Time, and Fuel Calculations**:
  + Follows provided formulas for base cost, fuel usage, profit, and total customer charge.
* **Delivery Records**: Stores up to 50 deliveries.
* **Reports**:
  + Total deliveries, distance covered, average delivery time, revenue, and profit.
* **Menu-Driven Interface**: Easy to navigate options in a loop.

**3. 🔧 Function Descriptions**

| **Function** | **Description** |
| --- | --- |
| addCity() | Adds a new city to the list. |
| listCities() | Displays existing cities with indexes. |
| inputDistance() | Accepts input for distances between any two cities (symmetric). |
| printDistanceTable() | Shows a 2D distance matrix. |
| handleDelivery() | Takes user input, validates, and performs delivery calculations. |
| showReports() | Prints performance stats from all deliveries. |

**4. 🧱 Code Structure**

All code is written in a single file: main.c for simplicity. It includes:

* Definitions of all global constants and structures.
* Global arrays for cities, distances, and deliveries.
* Modular functions for city management, deliveries, and reports.
* A main() function containing the user menu.

*Optional future enhancement:* Split into modules like city.c, delivery.c, report.c.

**5. 🌐 GitHub Repository**

**Link -** https://github.com/Savidya448/Logistics-Management-System.git

**Commit History**

At least 15 commits were made with clear messages, for example:

* Initialized project with city and distance setup
* Added vehicle structure and delivery handling
* Implemented cost and fuel calculation
* Finalized report generation and summary

**6. ✅ Conclusion**

This project helped reinforce concepts of arrays, structures, conditionals, and functions in C. It provided hands-on experience designing a functional application from scratch. With further improvements such as file handling and path optimization, the system can be expanded to a fully usable logistics simulator.