Procurement and Logistics Management System

Keerthi K	PES1UG22CS284	5E
Kota Shriya	PES1UG22CS290	5E

Implementation Document

1. Introduction

This document outlines the implementation details for the Logistics and Procurement Management System. It includes the technical choices made, coding standards, design patterns used, database setup, and a brief description of each functional module in the system.

2. Technology Stack

Backend:

- **Programming Language:** Python (Django/Flask)
- **Framework:**Python (Django/Flask for RESTful API design)
- **Database:** MySQL (Structured storage for Supplier, Inventory, Orders, and Shipment data)
- **Version Control:** GitHub (Code repository and version control)

Frontend:

- **HTML5/CSS3/JavaScript**

API:

- RESTful APIs will be used for communication between the frontend and backend. These will handle CRUD operations for suppliers, inventory, orders, and shipments.

3. Coding Details and Structure

Backend Code Structure

Main Modules:

- **Supplier Management Module**
  - CRUD operations for suppliers
  - Fetching Supplier Data
  - APIs: `createSupplier()`, `updateSupplier()`, `deleteSupplier()`, `getSupplier()`
- **Inventory Management Module**
  - Inventory tracking based on shipments and orders
  - APIs: `addInventory()`, `updateInventory()`, `removeInventory()`
- **Order Management Module**
  - Order creation and procurement tracking
  - APIs: `createOrder()`, `updateOrder()`, `deleteOrder()`, `getOrderDetails()`
- **Shipment Management Module**
  - Track shipment details, status, and logistics
  - APIs: `createShipment()`, `updateShipment()`, `getShipmentStatus()`

Database Design

Tables:

- **Supplier Table:**
  - `supplierID`, `supplierName`, `contactInfo`
- **Inventory Table:**
  - `inventoryID`, `itemName`, `stockLevel`, `locationID`
- **CustomerOrder Table:**
  - `orderID`, `customerID`, `orderDate`, `status`
- **Shipment Table:**
  - `shipmentID`, `deliveryRoute`, `logisticsID`, `deliveryDate`

---

4. API Implementation

1. Supplier Management API:

- **Endpoint:** `/supplier`
- **Methods:**
  - `POST /supplier`: Create a new supplier
  - `GET /supplier/{id}`: Get supplier details by ID
  - `PUT /supplier/{id}`: Update supplier details
  - `DELETE /supplier/{id}`: Delete a supplier

2. Inventory Management API:

- **Endpoint:** `/inventory`
- **Methods:**
  - `POST /inventory`: Add inventory
  - `GET /inventory/{id}`: Get inventory details
  - `PUT /inventory/{id}`: Update inventory stock level
  - `DELETE /inventory/{id}`: Remove inventory

3. Order Management API:

- **Endpoint:** `/order`
- **Methods:**
  - `POST /order`: Create a new customer order
  - `GET /order/{id}`: Fetch order details
  - `PUT /order/{id}`: Update order status
  - `DELETE /order/{id}`: Cancel an order

---

5. Testing and Debugging

- **Unit Tests:** Each module will have its own set of unit tests.
- **Integration Tests:** Ensure that all modules (supplier, inventory, shipment, etc.) work together seamlessly.
- **Automated Testing:** Using PyTest for backend unit tests.
- **Debugging Tools:** IDE-based debugging tools like VSCode for Python.