21 CSC205P DATABASE MANAGEMENT SYSTEM

"GENERAL STORE INVENTORY MANGEMENT SYSTEM"

PROJECT REPORT

TEAM MEMBERS:

1) SPANDAN CHOUBEY (RA2311003010392)

2) ABHIK RAJ (RA2311003010406)

3) MUTHUARYAN AASARRI (RA2311003010375)

INTRODUCTION

Problem Understanding

Managing inventory in a general store involves tracking products, orders, customers, suppliers, and payments. Manual processes often lead to stock discrepancies, inefficient order processing, and poor supplier coordination. This project aims to develop an automated **Inventory Management System** to optimize stock control, streamline order fulfillment, and enhance business operations.

Solution

A **MERN stack** (MongoDB, Express.js, React, Node.js) application designed to automate inventory management, supplier coordination, and sales tracking. The system will leverage **MongoDB** for flexible and scalable data handling, **Express.js** and **Node.js** for efficient backend operations, and **React** for a dynamic and user-friendly interface. This full-stack solution will optimize stock control, streamline order fulfillment, and enhance overall business operations.

Features

- **Product Tracking** – Monitor stock levels, prices, and reorder alerts.

- **Order Management** – Process customer orders and track fulfillment.

- **Supplier Coordination** – Manage supplier details and supply chains.

- **Payment Integration** – Record and track customer payments.

- **Employee Management** – Assign roles and track employee details.

- **Category Classification** – Organize products into categories for easy navigation.

- **Role-Based Access Control** – Secure access for admins, employees, and suppliers.

- **Reporting Dashboard** – Generate sales, stock, and financial reports.

DATABASE DESIGN

Entities and Attributes

1. **Store**
   - **Key Attribute:** Store_ID (Primary Key)
   - **Simple Attribute:** Store_Name
   - **Composite Attribute:** Location (Street, City, Pincode)
   - **Multi-valued Attribute:** Contact_Numbers

2. **Employee**
   - **Key Attribute:** Employee_ID (Primary Key)
   - **Composite Attribute:** Name (First_name, Last_Name)
   - **Simple Attributes:** Salary, Position
   - **Composite Attribute:** Address (Street, City, Pincode)

3. **Customer**
   - **Key Attribute:** Customer_ID (Primary Key)
   - **Simple Attributes:** Name, Email
   - **Multi-valued Attribute:** Phone

4. **Supplier**
   - **Key Attribute:** Supplier_ID (Primary Key)
   - **Simple Attributes:** Supplier_Name
   - **Multi-valued Attribute:** Contact_Numbers
   - **Composite Attribute:** Address (Street, City, Pincode)

5. **Product**
   - **Key Attribute:** Product_ID (Primary Key)
   - **Simple Attributes:** Product_Name, Price, Stock_Quantity
   - **Derived Attribute:** Reorder_Level (calculated from Stock Quantity)

6. **Order**

   - o **Key Attribute:** Order_ID (Primary Key)

   - o **Foreign Key:** Customer_ID (references Customer)

   - o **Simple Attributes:** Order_Date, Total_Amount

7. **Payment**

   - o **Key Attribute:** Payment_ID (Primary Key)

   - o **Simple Attributes:** Payment_Method, Amount, Date

8. **Category (For Product Classification)**

   - o **Key Attribute:** Category_ID (Primary Key)

   - o **Simple Attributes:** Category_Name, Description

Mapping Cardinalities

1. **Store employs Employee** - **1:M Relation**

   - o A store employs many employees, but an employee works for only one store.

2. **Customer places Order** -**1:M Relation**

   - o A customer can place many orders, but an order belongs to one customer.

3. **Supplier supplies Product** - **M:M Relation**

   - o A supplier can supply many products, and a product can be supplied by many suppliers.

4. **Order contains Product** - **M:M Relation**

   - o An order can contain multiple products, and a product can be in multiple orders.

5. **Order has Payment** - **1:1 Relation**

   - o An order is associated with exactly one payment.

6. **Product belongs to Category** - **M:1 Relation**

   - o Many products belong to one category.

Extended Features

The ER diagram incorporates aggregation and specialization to enhance data representation.

- **Aggregation:** The "Supplies" relationship between Supplier and Product is aggregated into "Purchases" with Store, effectively modeling the M:N relationship between suppliers, products, and stores.
- **Specialization:** The Product entity is specialized into "Perishable" and "Non-Perishable", allowing better inventory management, such as expiration tracking for perishable goods.

ER DIAGRAM