SAVEETHA
INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
(Declared as Deemed to be University under Section 3 of UGC Act 1956)

ENGINEER TO EXCEL

TITLE OF THE PROJECT

Inventory Management Systems (IMS)

NAME : K. Arun kumar reddy , 192211542

N. Harsha Vardhan gowd, 192211571,

N. Santhosh kumar reddy, 192211691

COURSE NAME: Database Management System (DBMS)

COURSE CODE: CSA0537

Guided by Dr. Carmel Mary Belinda

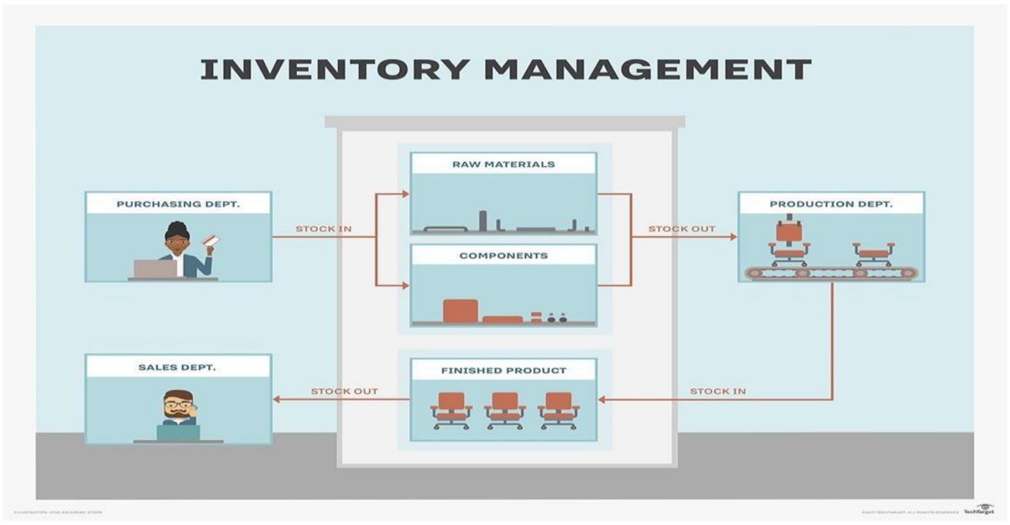
TITLE OF THE PROJECT

Inventory Management Systems (IMS)

TABLE OF CONTENTS:

Abstract:

This study aims to develop an Inventory Management Systems (IMS) that can provide better control and handling product stock, customer order, customer service and order delivery that relates to company inventory information. The target user is the owner and employee in Small and Medium Enterprise (SME) retail store that stills manages inventory manually in Malaysia. IMS helps retail store to track down the next arrival of product stocks and record customer order for reservation for the product in the store inventory. In this study, the developer used PHP for backend system development and HTML, CSS, JavaScript for frontend system development. This study also applies Rapid Application Development (RAD) software methodology that emphasize on iterative development process. Even though the inventory management system has been fully developed by the developer, there are still limitations found and future enhancement that can be made towards the system.



An Inventory Management System for a company in Indonesia has been developed to manage warehouse inventory that is produced with the goals of reducing error on recording product stock into the warehouse inventory and make process of product in and out of the warehouse inventory more effective. To achieve that, simple functionality has included in the system by the researcher which the overall functionality is view of inventory information data including product detailed report within time range and form template for managing the inventory product information. As from the research conclusion, the author work of web-based inventory management application is efficient and effective which will lessen some job on inventory management (S Pasaribu, 2021).

Keywords: CSS, HTML, Inventory Management Systems (IMS), JavaScript, PHP, Rapid Application Development (RAD),
Small and Medium Enterprise (SME)

INTRODUCTION:

Inventory management is one of the crucial supply chain components in retail store. Every day or by weekly, the store needs to update the stock that is coming in or out. Most of the retail store must at east have a warehouse to store their products. To remain customer satisfaction whenever the shop does not have the product retail company provide delivery service through customer order for product that comes from warehouse inventory. Having inventory management system within an organization is important because the business can monitor and control their product stock and business revenue that is going on within the organization. At the same time, it is also to determine the suitable product quantity to restock according to customer and market demand which will reduce business loss of overstocking (Plimer & Borisov, 2016). Without having good inventory management system within organization, it can cause many businesses risk especially for retail store such as out of stock and product that is not sold due to market demand which will bring a dissatisfaction to customer and business lost (Patil & Divekar, 2014). Moreover, inventory management in an organization are done manually such as updating and checking inventory stocks in an excel and or logbook in some company nowadays and in the old days.

An Inventory Management System (IMS) is a crucial tool for businesses to efficiently monitor, control, and optimize their inventory levels and operations. It involves the use of software and processes to track the flow of goods from suppliers to customers, ensuring that the right products are available in the right quantities at the right time. The primary goal of an IMS is to strike a balance between having enough inventory to meet customer demand without excessive overstocking that can lead to increased storage costs, obsolescence, and tied-up capital. Effective inventory management encompasses various activities such as inventory tracking, forecasting demand, setting reorder points, managing suppliers, and analyzing inventory turnover rates.

Key features of an Inventory Management System typically include:

- . Inventory Tracking: Keeping real-time records of inventory levels,

- 2. locations, and movements within the supply chain. Demand Forecasting: 3. Using historical data and statistical methods to predict future demand for products 1
- Improved inventory accuracy and visibility
- Reduced stockouts and overstock situations
- Enhanced order fulfilment and customer satisfaction

Methodology:

Developing an effective Inventory Management System (IMS) requires a structured methodology to ensure that the system meets the business requirements, is scalable, and can adapt to changing needs. Here's a step-by-step methodology that can be followed:

- Define Requirements: Start by clearly defining the requirements of the IMS. This includes understanding the types of products to be managed, inventory levels, order processing workflows, reporting needs, integration with other systems (e.g., accounting, sales), user roles and permissions, and any regulatory or compliance requirements.

- Gather Data: Collect and analyze historical inventory data, sales data, supplier information, and any other relevant data sources. This data will be used for demand forecasting, setting reorder points, and optimizing inventory levels.

- Select Technology Stack: Choose the appropriate technology stack for developing the IMS, considering factors such as scalability, security, integration capabilities, and ease of maintenance. Common technologies used for IMS development include databases (e.g., SQL, NoSQL), programming languages (e.g., Python, Java), and frameworks/libraries (e.g., Django, Flask for web development).

- Design Database Schema: Design the database schema for storing inventory data, including tables for items, inventory levels, transactions, suppliers, purchase orders, sales orders, etc. Ensure normalization and data integrity principles are followed to avoid redundancy and inconsistencies.

- Develop Core Functionality: Implement the core functionality of the IMS, such as inventory tracking, demand forecasting, reorder point calculation, supplier management, order processing, reporting, and user management. Use modular and reusable code practices to facilitate future enhancements and maintenance.

- Integrate External Systems: If required, integrate the IMS with external systems such as ERP systems, e-commerce platforms, accounting software, and logistics systems. Ensure seamless data flow and synchronization between systems to avoid data silos and manual data entry errors.

- Implement Security Measures: Implement robust security measures to protect sensitive inventory data, user credentials, and system functionalities. This includes authentication, authorization, data encryption, audit trails, and regular security audits and updates.

- Test and QA: Conduct thorough testing and quality assurance (QA) of the IMS to identify and resolve any bugs, performance issues, or usability issues. Perform unit testing, integration testing, regression testing, and user acceptance testing (UAT) with real-world scenarios.

- Training and Deployment: Provide training to users and stakeholders on how to use the IMS effectively. Prepare documentation, user manuals, and training materials. Deploy the IMS in a staged manner, starting with a pilot phase before full-scale deployment to ensure smooth transition and minimize disruptions.

- Monitor and Optimize: Continuously monitor the performance of the IMS, gather feedback from users, and make iterative improvements based on insights and feedback. Use analytics and reporting tools to track key performance indicators (KPIs) such as inventory turnover, stockout rates, order fulfillment metrics, and cost savings.

Implementation:

- Implementing an Inventory Management System (IMS) involves translating the design and functionality into a working system that can be used by the business. Here's a step-by-step guide to implementing an IMS:

- Set Up Development Environment:

- Install the necessary software and tools for development, such as a database management system (e.g., MySQL, PostgreSQL), a programming environment (e.g., Python, Java), and any frameworks or libraries required for the IMS development.
- Database Creation:

- Create the database schema based on the design specifications. Define tables for items, inventory levels, transactions, suppliers, purchase orders, sales orders, users, roles, etc. Set up relationships and constraints to ensure data integrity.
- Backend Development:

- Develop the backend logic of the IMS, including functions for inventory tracking, demand forecasting, reorder point calculation, supplier management, order processing, reporting, and user management. Use appropriate programming languages and frameworks to implement these functionalities.
- Frontend Development:

- Design and develop the user interface (UI) for the IMS. Create screens for viewing and managing inventory, placing orders, generating reports, and administering user roles and permissions. Ensure the UI is intuitive, user-friendly, and responsive across different devices.
- Integration with External Systems:

- If the IMS needs to integrate with external systems such as ERP software, e-commerce platforms, or accounting systems, implement the necessary APIs or connectors to enable seamless data exchange and synchronization.
- Security Implementation:

- Implement security measures to protect the IMS and its data. This includes authentication mechanisms (e.g., username/password, multi-factor authentication), authorization rules (based on user roles and permissions), data encryption for sensitive information, and audit logging for tracking system activities.

Code:

Database schema Creation:

```
-- Create Tables--
CREATE TABLE Items (
   ItemID INT PRIMARY KEY,
   ItemName VARCHAR(255) NOT NULL,
   Category VARCHAR(50),
   UnitPrice DECIMAL(10, 2) NOT NULL
);

CREATE TABLE Inventory (
   InventoryID INT PRIMARY KEY,
   ItemID INT,
   Quantity INT,
   FOREIGN KEY (ItemID) REFERENCES Items(ItemID)
);

CREATE TABLE Suppliers (
   SupplierID INT PRIMARY KEY,
   SupplierName VARCHAR(255) NOT NULL,
   ContactInfo VARCHAR(100)
);

CREATE TABLE PurchaseOrders (
   OrderID INT PRIMARY KEY,
   SupplierID INT,
   OrderDate DATE,
   Status VARCHAR(20),
   FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)
);

CREATE TABLE OrderDetails (
   OrderDetailID INT PRIMARY KEY,
   OrderID INT,
   ItemID INT,
   Quantity INT,
   FOREIGN KEY (OrderID) REFERENCES PurchaseOrders(OrderID),
   FOREIGN KEY (ItemID) REFERENCES Items(ItemID)
);
```

Insearting data:

```
INSERT INTO Items (ItemID, ItemName, Category, UnitPrice)
VALUES (1, 'Product A', 'Electronics', 500.00);

INSERT INTO Inventory (InventoryID, ItemID, Quantity)
VALUES (1, 1, 100);
```

Managing Suppliers:

```
INSERT INTO Suppliers (SupplierID, SupplierName, ContactInfo)
VALUES (1, 'Supplier X', 'contact@supplierx.com');

-- Update supplier contact info
UPDATE Suppliers
SET ContactInfo = 'newemail@supplierx.com'
WHERE SupplierID = 1;
```

Inventory Management:

```
-- Update inventory after receiving items from a purchase order
UPDATE Inventory
SET Quantity = Quantity + 50
WHERE ItemID = 1;

-- Calculate total inventory value
```

```
SELECT SUM(Quantity * UnitPrice) AS TotalInventoryValue
FROM Inventory
JOIN Items ON Inventory.ItemID = Items.ItemID;
```

Conclusion:

In conclusion, developing and implementing an Inventory Management System (IMS) is crucial for businesses to effectively manage their inventory, streamline operations, and optimize resources. A well-designed IMS can improve inventory accuracy, reduce stockouts and overstock situations, enhance order fulfillment, and ultimately contribute to cost savings and increased customer satisfaction.

Database Design: Creating a structured database schema to store inventory data, supplier information, purchase orders, and related details.

- Functionality Development: Implementing core functionalities such as inventory tracking, demand forecasting, reorder point calculation, supplier management, order processing, reporting, and user management.

- Integration: Integrating the IMS with external systems such as ERP software, e-commerce platforms, and accounting systems to enable seamless data exchange and synchronization.

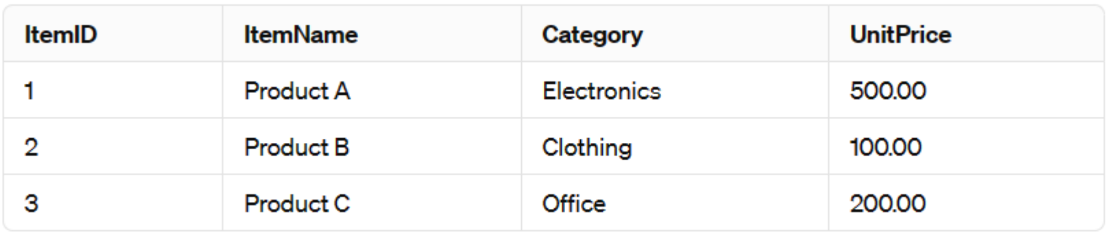
- Security Measures: Implementing robust security measures to protect sensitive inventory data, user credentials, and system functionalities.

- Testing and Training: Conducting thorough testing and quality assurance (QA) to ensure the IMS functions correctly and providing training to users for effective system usage.

Result: The result of developing an Inventory Management System (IMS) project is a functional system that enables businesses to efficiently manage their inventory, suppliers, purchase orders, and related processes.

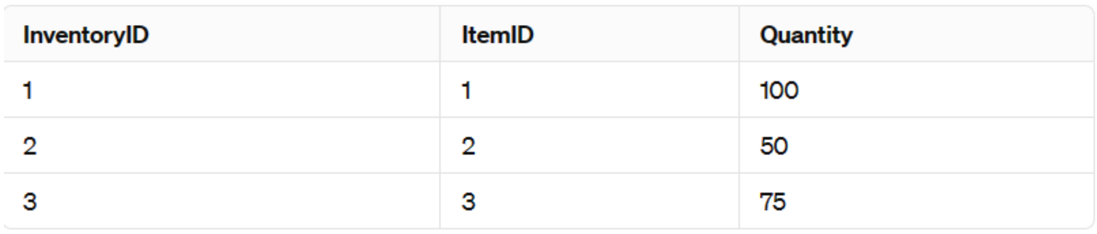
Outputs:

Items table



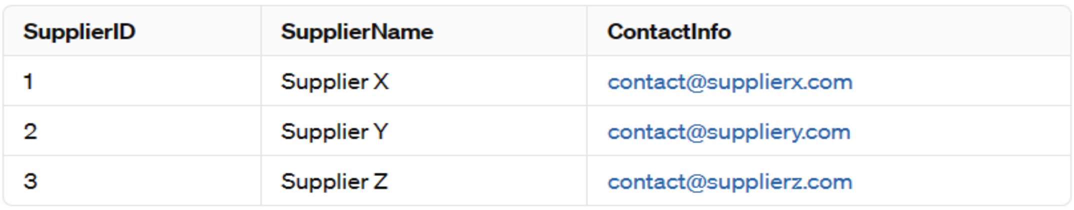
ItemID	ItemName	Category	UnitPrice
1	Product A	Electronics	500.00
2	Product B	Clothing	100.00
3	Product C	Office	200.00

Inventory table



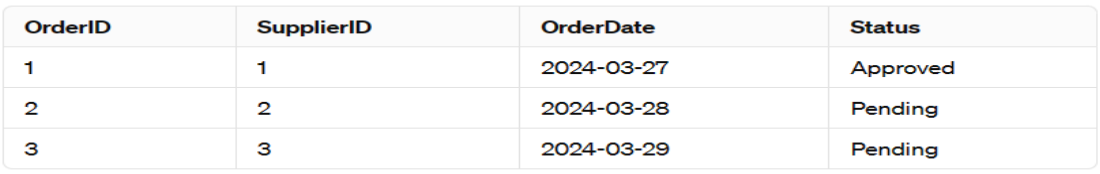
InventoryID	ItemID	Quantity
1	1	100
2	2	50
3	3	75

Supplier table



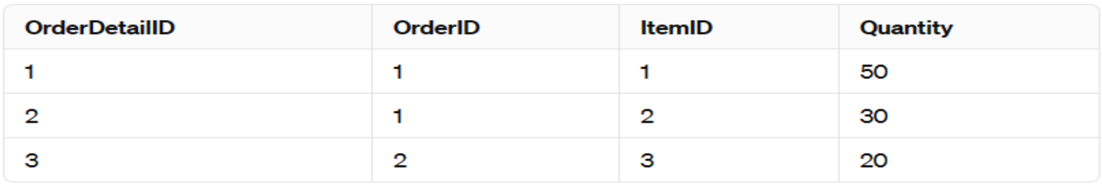
SupplierID	SupplierName	ContactInfo
1	Supplier X	contact@supplierx.com
2	Supplier Y	contact@suppliery.com
3	Supplier Z	contact@supplierz.com

Purchase order table



OrderID	SupplierID	OrderDate	Status
1	1	2024-03-27	Approved
2	2	2024-03-28	Pending
3	3	2024-03-29	Pending

Order details table



OrderDetailID	OrderID	ItemID	Quantity
1	1	1	50
2	1	2	30
3	2	3	20