Inventory Management System (IMS) Report

1. Introduction

The Inventory Management System (IMS) is a robust application designed to manage inventory, suppliers, orders, and users efficiently. Built using WPF in Visual Studio and integrated with SQL Server, the system ensures scalability, reliability, and user-friendly operation. It employs a modular architecture, enabling flexibility in feature addition and maintenance.

2. Project Objectives

- **Centralized Inventory Management: Simplify the tracking of stock levels, product details, and categories.**
- **Efficient Order Handling: Streamline purchase and sales order workflows with detailed tracking.**
- **User Role Management: Secure the system with role-based access for administrators, managers, and staff.**
- **Data-Driven Insights: Provide actionable data through real-time reports and audit logs.**
- **Scalability: Enable seamless integration of additional modules as the system grows.**

3. Application Architecture

3.1 User Interface

The IMS interface is developed using WPF, employing modern design principles for usability and aesthetics:

- **Color Scheme: Navy blue, sky blue, and silver for a professional appearance.**
- **Navigation:**
  - **Welcome Page: Features Login and Signup buttons.**
  - **Dashboard: Displays an overview of sales, inventory levels, and pending orders.**
  - **Management Modules:**
    - **Inventory: View, add, and edit product details.**
    - **Orders: Manage purchase and sales orders.**

- **Reports: Generate summaries of key metrics.**

- **Responsive Design: Ensures consistent functionality across devices and resolutions.**

3.2 Database Integration

- **SQL Server, hosted on DESKTOP-8BL3MIG, serves as the backend database.**
- **Tables and relationships are designed to support complex workflows and ensure data integrity.**
- **CRUD operations are seamlessly integrated with the WPF application for real-time data manipulation.**

4. Database Design

The IMS database, named InventoryManagement10, consists of the following tables:

1.

Products:

2.

- **Tracks product information, including Name, SKU, Category, Quantity, and UnitPrice.**
- **Sample data includes items like gaming consoles, desks, and books.**

3.

Suppliers:

4.

- **Stores supplier details such as ContactName, Phone, and Address.**

5.

Categories:

6.

- **Organizes products into groups like Electronics, Furniture, and Books.**

7.

PurchaseOrders & PurchaseOrderDetails:

8.

- **Tracks supplier orders and their respective product details.**

9.

SalesOrders & SalesOrderDetails:

10.

- **Manages customer orders, including statuses like Pending, Shipped, and Cancelled.**

11.

StockMovements:

12.

- **Logs changes in stock, categorized as IN, OUT, or ADJUSTMENT.**

13.

Users:

14.

- **Handles user credentials and roles (Admin, Manager, Staff).**

15.

AuditLogs:

16.

- Tracks system activities, detailing actions, affected tables, and timestamps.

---

5. Current Progress

5.1 Database Development

- **Database schema fully implemented with relationships and constraints.**
- **Sample data pre-loaded into key tables for testing and validation.**
- **Foreign keys ensure integrity between related tables like Products, Orders, and Users.**

5.2 Application Development

- **Visual Studio solution (IMS.MainMenu.sln) created with the following progress:**
  - **UI Development:**
    - **Initial designs for Login, Welcome Page, and Dashboard completed.**
    - **Consistent use of the defined color scheme.**
  - **Backend Integration:**
    - **Basic CRUD operations connected to the SQL database.**

---

6. Future Development

1.

UI Enhancements:

2.

- **Design consistent forms for product, supplier, and order management.**
- **Add visual elements like charts and graphs for data representation.**

3.

Advanced Features:

4.

- **Implement notification alerts for low stock or overdue orders.**
- **Allow bulk upload and export of inventory data.**

5.

Testing and Debugging:

6.

- **Conduct unit tests for individual modules.**
- **Perform integration testing to ensure seamless operation.**

7.

Security Improvements:

8.

- **Encrypt sensitive information such as passwords.**
- **Add user session tracking for enhanced security.**

9.

Deployment:

10.

- **Develop an installer for easy deployment.**
- **Configure multi-user support for larger teams.**

---

7. Conclusion

The IMS project has made significant progress, with the database schema fully designed and the initial UI implemented. Future developments will focus on enhancing usability, integrating advanced features, and ensuring the system’s reliability. Once completed, IMS will be a comprehensive tool for inventory and order management, capable of scaling with organizational needs.