**RFID-Based Inventory Management System**

1. **Project Overview**

This project involves the design and development of an RFID-based Inventory Management System that integrates both online and offline purchase methods. The system allows customers to register on the website, place orders online, or directly buy from a physical store. It ensures that inventory data is updated in real-time, providing accurate stock levels on the website and preventing discrepancies.

Additionally, the system incorporates Firebase for real-time UID updates when RFID tags are scanned. Although the integration of Firebase with the website is not complete, two separate systems have been developed:

A local host/website using PHP and MySQL for customer and inventory management.

A system that pushes scanned RFID tag UIDs to Firebase.

1. **Hardware Components**

The following hardware components are used in the project:

1. **RAIN RFID / RC522 RFID Module**

Scans RFID tags attached to items in the store for inventory management.

1. **Round Tag**

An RFID tag attached to each item to uniquely identify the product.

1. **ESP32**

A microcontroller used to interface the RFID module with the system and send UIDs to Firebase.

1. **Buck Converter**

Regulates the power supply to the ESP32 and other components from the battery source.

1. **Battery Source**

Provides the necessary power to the system.

1. **Software Stack**

The software components used in this project include:

**Frontend**: HTML, CSS, JavaScript

Provides the user interface for the website where customers can register, view products, and place orders.

**Backend**: PHP

Manages customer registration, order processing, and communication with the database.

**Database**: MySQL (Local) & Firebase (for UID handling)

MySQL stores customer data, inventory details, and order information.

Firebase receives and processes UIDs from scanned RFID tags for future integration with the website.

1. **Workflow**

**4.1. Online Order Workflow**

1. Customer Registration:

Customers register on the website by providing details like name, contact information, and shipping address.

1. Placing an Order:

Customers browse the product catalog on the website and place orders for desired items.

1. Order Verification:

The distributor verifies and confirms orders. Upon confirmation, the database updates the order details, and the website reflects changes in the inventory.

1. Stock Management:

The ordered quantity is deducted from the stock, and the updated inventory is displayed. Out-of-stock items trigger an appropriate notification on the website.

**4.2. Store Purchase Workflow**

1. RFID Tag Scanning:

Items purchased in-store are scanned using the RFID reader module. The scanned UID is sent to Firebase.

1. Billing:

The system retrieves item details from the database, and the billing process is initiated.

1. Inventory Update:

The inventory in the database is updated and reflected on the website. Notifications are displayed for out-of-stock items.

1. **Features**
2. **Real-Time Stock Updates:**

Inventory levels are dynamically updated for both online and in-store purchases.

1. **Customer Registration:**

Users can register, track orders, and manage their profiles.

1. **Stock Alerts:**

Notifications are automatically sent for out-of-stock items.

1. **Order Management:**

Orders placed online are verified by a distributor before confirmation.

1. **UID Integration with Firebase:**

RFID tag UIDs are pushed to Firebase for real-time updates (integration with the website is ongoing).

1. **Future Enhancements**
2. **Firebase and Website Integration:**

Full synchronization of Firebase data with the website for seamless inventory updates.

1. **Blockchain Integration:**

To enhance the security of inventory and order tracking.

1. **Mobile App Development:**

A mobile app for customers to place orders and track inventory on the go.

1. **Conclusion**

The RFID-Based Inventory Management System offers an efficient solution for managing inventory across both online and offline sales channels. While currently divided into two separate systems, the planned Firebase integration will enable seamless real-time updates. The system ensures accurate stock tracking, real-time updates, and a smooth customer experience, with potential for future expansions like blockchain and mobile app support.