## **Hackathon submission template (level -1-solution)**

**Use case Title: Smart stock- Realtime inventory intelligence for scalable retail operation**

**Student name –** Anuja k

**Register number –** 411823243003

**Institution –** RRASE college of engineering

**Department –** B. Tech (AI&DS)

**Date of submission -** 20.05.2025

1. **Problem Statement:**

Retailers face significant challenges in managing inventory efficiently across multiple locations. Manual stock checks, delayed updates, and lack of real-time visibility often lead to overstocking, stockouts, and lost sales opportunities. These inefficiencies increase operational costs, reduce customer satisfaction, and limit the scalability of retail businesses. There is a pressing need for a solution that provides accurate, real-time inventory data and actionable insights to optimize stock levels and streamline retail operations.

1. **Proposed Solution:**

Smart Stock is an AI-powered real-time inventory intelligence platform designed for scalable retail operations. The system uses IoT sensors, computer vision, and cloud analytics to continuously monitor stock levels across stores and warehouses. Key features include:

* **Real-time Stock Monitoring:** Automated tracking of inventory using sensors and image recognition.
* **Predictive Replenishment:** AI-driven forecasts to optimize reorder points and prevent stockouts or overstocking.
* **Centralized Dashboard:** Unified view of inventory status across locations, accessible via web and mobile apps.
* **Automated Alerts:** Instant notifications for low stock, anomalies, or discrepancies.
* **Analytics & Reporting:** Actionable insights on sales trends, shrinkage, and inventory turnover.

1. **Technologies & Tools Considered:**

* **IoT Devices:** RFID tags, weight sensors, and smart shelves for real-time data capture.
* **Computer Vision:** Cameras with AI models for visual stock counting and anomaly detection.
* **Cloud Platforms:** AWS, Azure, or Google Cloud for data storage and processing.
* **Programming Languages:** Python
* **Frameworks & APIs:** TensorFlow/PyTorch (AI), React (frontend), RESTful APIs for system integration.

1. **Solution Architecture & Workflow**

**Major Components:**

* **Edge Devices:** IoT sensors and cameras installed in stores/warehouses.
* **Data Aggregation Layer:** Collects and preprocesses data from all edge devices.
* **Cloud Analytics Engine:** Processes data, runs AI models for forecasting and anomaly detection.
* **Central Dashboard:** Visualizes real-time inventory data and analytics for users.
* **Notification System:** Sends alerts via email/SMS/app notifications.
* **Sensors and data collection:** smart sensors track stock levels in real time.
* **Ai processing:** machine learning models analyze trends and forecast needs.

**Workflow:**

* Inventory data is captured by IoT sensors and cameras.
* Data is transmitted securely to the cloud.
* AI models analyze the data for stock levels, trends, and anomalies.
* Results are displayed on a centralized dashboard.

**Flowchart:**

Start system

Inventory monitoring

Real time inventory update

Data collection & preprocessing

Security &anomaly detection

Threshold check

End / reset

Dashboard &alerts

1. **Feasibility & Challenges:**

**Feasibility:**

The solution leverages mature technologies (IoT, AI, cloud computing) that are widely available and proven in retail environments. Integration with existing POS and ERP systems is possible via APIs, making deployment practical for both small and large retailers.

**Challenges:**

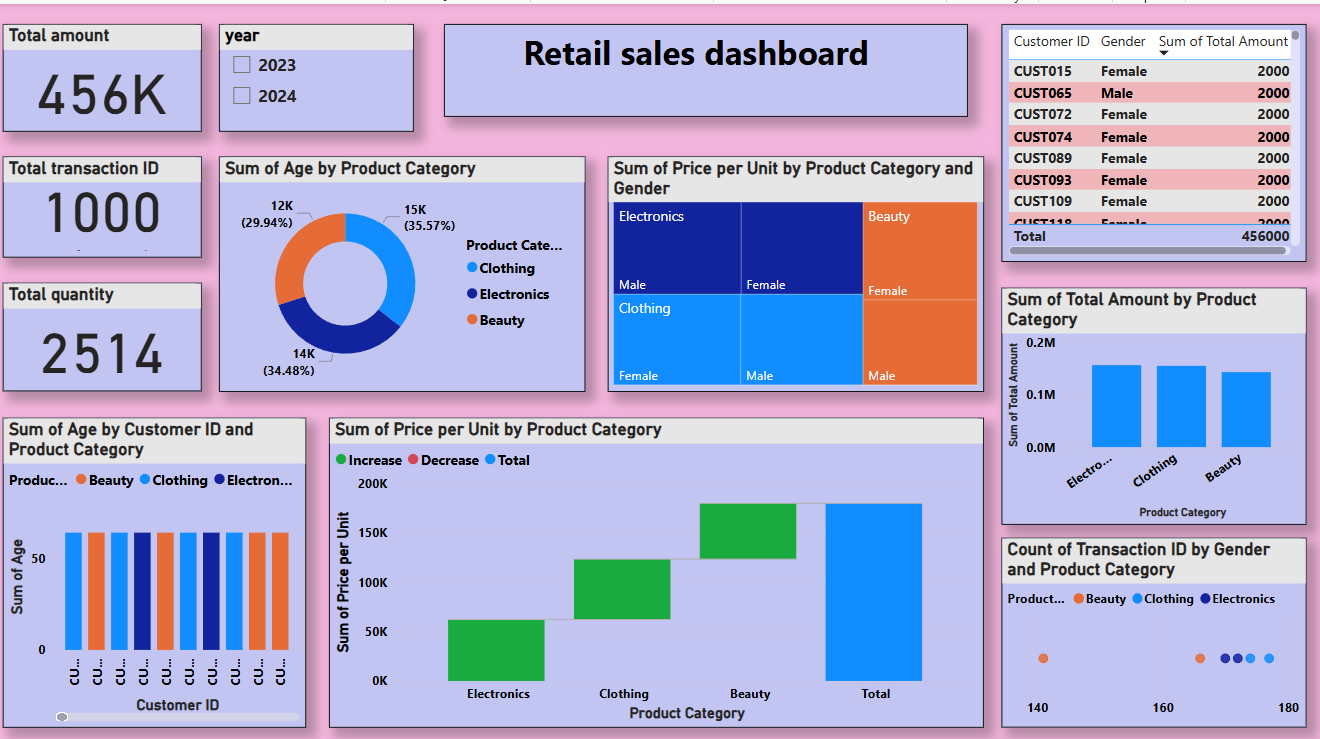
* **Hardware Costs:** Initial investment in IoT devices and cameras.
* **Data Privacy:** Ensuring secure handling of sensitive inventory and sales data.
* **Integration Complexity:** Connecting with legacy systems may require custom development.
* **Scalability:** Managing large data volumes and real-time processing as the business grows.

**Mitigation Strategies:**

* Use phased implementation to manage costs.
* Employ robust encryption and access controls.
* Develop modular APIs for easier integration.

1. **Expected Outcome & Impact:**

* **Reduced Stockouts/Overstocking:** Improved inventory accuracy and timely replenishment.
* **Operational Efficiency:** Less manual intervention, freeing staff for higher-value tasks.
* **Cost Savings:** Lower carrying costs and minimized lost sales.
* **Enhanced Customer Experience:** Higher product availability and satisfaction.
* **Scalability:** Supports growth across multiple locations with minimal additional overhead.
* **Dashboard:**

****

* **GITHUB link:** [**https://github.com/anuja2244/anuja**](https://github.com/anuja2244/anuja%20)

1. **Future Enhancements:**

* **Integration with Supplier Systems:** Automate purchase orders and restocking.
* **Advanced Analytics:** Incorporate demand sensing and external data (weather, events).
* **Mobile App:** Enable on-the-go inventory management.
* **Self-Learning AI:** Continuously improve forecasts based on new data.
* **Sustainability Metrics:** Track and optimize for waste reduction and eco-friendly practices.