NIBM | THE CITY UNIVERSITY

# WELCOME

A simple supermarket Point-Of-Sale (POS) system

A simple supermarket Point-Of-Sale (POS) system

# WELCOME

KUDSE24.3F - 011 | KUSHAN RANDIKA HERATH - **Project Leader/Documentation**

KUDSE24.3F - 003 | DESAN YASANDU - **Developer/ Tester**

KUDSE24.3F - 004 | PRABHASHI PRAVEENA  - **Developer**

KUDSE24.3F - 019 | Harsha Herath - **Tester/Debugger**

NIBM THE CITY UNIVERSITY

# Introduction

- Point-Of-Sale (POS) systems are essential for facilitating the daily sales process in supermarkets. A POS system is a computer system used to manage product inventory, handle customer purchases, and process payments efficiently.

- This simple POS system is designed in Java using Object-Oriented Programming (OOP) methodology. It enables the management of key functions such as product inventory, customer carts, and payment processing in a simple and fast manner.

- Using this system, the sales process in supermarkets becomes more efficient, accurate, and convenient for customers.

# Purpose

- A simple POS system for efficient warehouse management, customer cart control, and payment processing.

# Problems Addressed

- Inventory Management Issues: Errors in saving and updating product inventory.

- Payment Calculation Errors: Over or underpayments reported.

- Customer Service Delays: Slow Payment Speed.

- Information Deduction and Manual Calculation: Additional Work and Errors.

- Product ID Errors: Incorrect ID or Inventory Quantity.

# OOP Concepts Used

**Class & Object:**
Create classes like Product, Cart, CartItem, and SupermarketDriver, and create their objects.

**Interface:**

Provide common methods for product types using the AccSall interface.

**Encapsulation**:

Hide data (fields) in classes as private and control them with public getter and setter methods.

**Abstraction**:
Hide internal functionality and provide only the details needed by the user.

**Polymorphism:**

Manage different product types with the same methods using an interface.

# Further Development

**Dynamic Data Structures:**

- Release the cart size limit by using data structures like ArrayList<CartItem>.

**Database Integration:**

- To protect product inventory, sales reports, and customer information.

**Graphical User Interface (GUI):**

- To make the user interface easier to use using Swing, JavaFX, or another GUI framework.

**Barcode Scanner Integration:**

- To identify products and enter them quickly.

**User Authentication & Roles:**

- Different access control according to staff.

**Discounts, Offers & Tax Calculations:**

- Calculate and charge discounts, promotions, and taxes.

# THANK YOU….

**Git Hub Clone the repository:**
https://github.com/Kushan20070126/Point-of-sales-for-a-super-market.git

NIBM
THE CITY UNIVERSITY