

Team Merchandising Revenue **System**



Sumanth simha -
Chaitanya M -
Shreyas paul -
Shivamani B -
Sai Rushikesh G -

INTRODUCTION

Introduction to Merchandise Revenue :

The code is based on a Team Merchandising Revenue System, which helps manage and track the sales of merchandise for a sports team or brand. In this system, different types of merchandise, such as jerseys, caps, and accessories, are added, sold, and tracked across various regions. The system records sales data and calculates the total revenue generated from the merchandise. It also analyzes sales by product and region, giving insights into which items are popular and where they are selling the most. This system streamlines inventory management and revenue tracking for merchandising businesses

PROBLEM STATEMENT

POC: CRUD: Merchandising Revenue System

track team merchandise sales(team id): Track sales of team merchandise.

A Team Merchandising Revenue System helps sports teams like FC Barcelona or Manchester United manage merchandise sales. It tracks revenue from items like jerseys and hats, allowing creation, updates, and deletion of merchandise records. The system monitors how many products are sold over time and analyzes revenue by product, store (online vs. physical), or region. This provides insights into top-selling items and regions, helping teams optimize their sales strategies and performance.

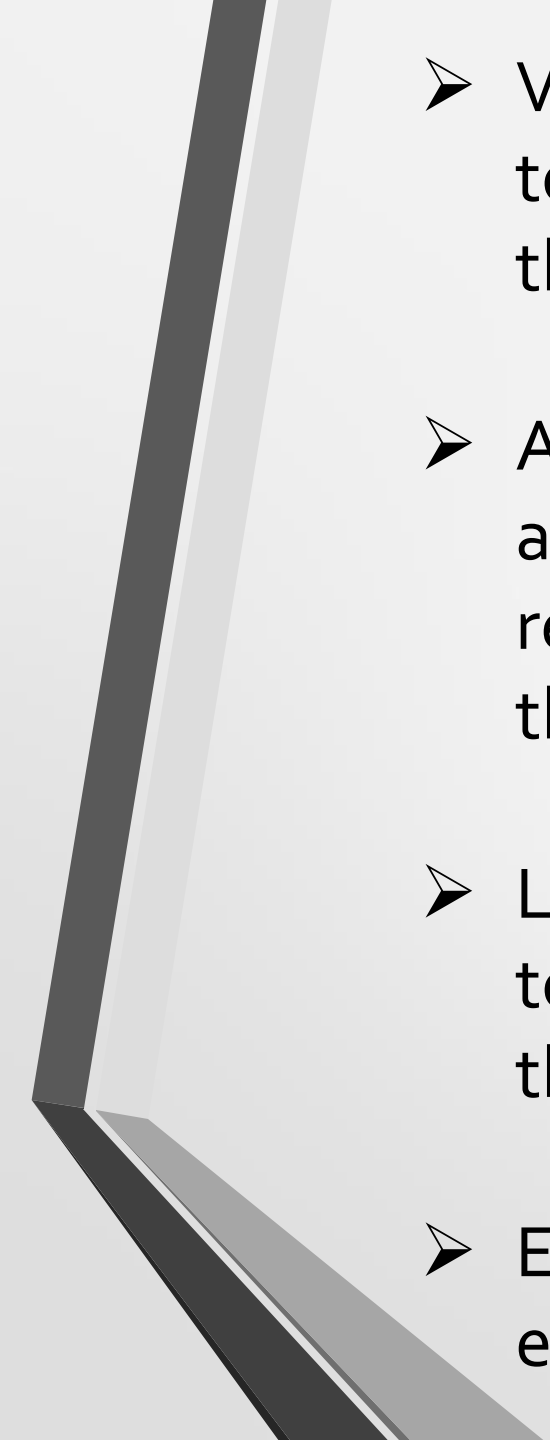
MENU

- Add Merchandise
- Delete Merchandise
- Record Sale
- Display total Revenue
- Analyze Sales (by Product and Region)
- View Merchandise List
- Exit

WORKING OF CODE

- Initialization:- The program starts by creating an instance of the **TeamMerchandisingRevenueSystem** class, which initializes two empty lists: one for merchandise (**merchandise_list**) and another for sales records (**sales_records**).
- Menu Display:- The program continuously displays a menu of options for the user to interact with: adding merchandise, deleting merchandise, recording sales, viewing total revenue, analyzing sales, or exiting the program.

- User Input & Corresponding Actions: - Add Merchandise: - The user is prompted to enter the name and price of a new merchandise item. This creates a **Merchandise** object, which is added to the system's **merchandise_list** using the **add_merchandise()** method.
- Delete Merchandise: - The user enters the name of an existing merchandise item to delete. The system removes it from **merchandise_list** using **delete_merchandise()**, updating the list by excluding the item with the given name.
- Record Sale:- The user selects an existing merchandise item by name, enters the quantity sold, and specifies the sales region. The program checks if the merchandise exists in the system. If found, it creates a **Sale** object and adds it to **sales_records** using the **record_sale()** method.

- 
- View Total Revenue:- The program calculates and displays the total revenue by summing up the total prices from all sales using the **total_revenue()** method.
 - Analyze Sales:- The system analyzes the sales data by product and by region using the **analyze_sales()** method. It tracks total revenue per product and per region, then displays the results to the user.
 - Looping:- After each action, the program re-displays the menu to prompt the user for another action. This loop continues until the user selects the option to exit.
 - Exit:- Once the user selects "**Exit**", the program terminates, ending the session.

1. User Chooses to Add Merchandise (Option 1)

•Input:

Enter merchandise name: jersey Enter merchandise price: 250

•**Action:** The user adds a merchandise item named "jersey" priced at ₹250.

•Output:

Merchandise 'jersey' added.

•**Explanation:** The system creates a **Merchandise** object and adds it to the merchandise list.

2. User Chooses to Record a Sale (Option 3)

•Input:

Enter merchandise name: jersey Enter quantity sold: 200 Enter region name: karnataka

•**Action:** The user records a sale for 200 "jersey" items in the region "karnataka."

•Calculation:

•Total Price = Price of jersey (₹250) × Quantity (200) = ₹50000.

•Output:

Sale recorded for jersey.

•**Explanation:** A **Sale** object is created with the merchandise, quantity, total price, and region. This object is added to the sales records.

3. User Chooses to Display Total Revenue (Option 4)

- **Action:** The user requests to see the total revenue generated from sales.
- **Output:**

Total Revenue: ₹50000.0

- **Explanation:** The system sums up the total price of all recorded sales, which in this case is just the sale of 200 jerseys.

4. User Chooses to Analyze Sales (Option 5)

- **Action:** The user selects option 5 to analyze sales by product and region.
- **Output:**

Sales by Product: jersey: ₹50000.0 Sales by Region: karnataka: ₹50000.0

- **Explanation:** The system compiles total sales figures for each product and each region. Since there was only one sale recorded, the revenue for both "jersey" and "karnataka" reflects the total revenue from that single sale.

5. User Chooses to Delete Merchandise (Option 2)

• **Input:**

Enter merchandise name to delete: jersey

• **Action:** The user deletes the "jersey" merchandise item from the merchandise list.

• **Output:**

Merchandise 'jersey' deleted.

• **Explanation:** The system removes the merchandise from the list, ensuring it can no longer be sold or recorded.

6. User Chooses to Exit the Program (Option 7)

• **Output:**

Exiting program.

• **Explanation:** The user chooses to exit, ending the interaction with the system.

1. Add Merchandise
2. Delete Merchandise
3. Record Sale
4. Display Total Revenue
5. Analyze Sales
6. View Merchandise List
7. Exit

Enter your choice: 1

Enter merchandise name: jersey

Enter merchandise price: 250

Merchandise 'jersey' added.

1. Add Merchandise
2. Delete Merchandise
3. Record Sale
4. Display Total Revenue
5. Analyze Sales
6. View Merchandise List
7. Exit

Enter your choice: 3

Enter merchandise name: jersey

Enter quantity sold: 200

Enter region name: karnataka

Sale recorded for jersey.

1. Add Merchandise
2. Delete Merchandise
3. Record Sale
4. Display Total Revenue
5. Analyze Sales
6. View Merchandise List
7. Exit

Enter your choice: 4

Total Revenue: ₹50000.0

1. Add Merchandise
2. Delete Merchandise
3. Record Sale
4. Display Total Revenue
5. Analyze Sales
6. View Merchandise List
7. Exit

Enter your choice: 5

Sales by Product:

jersey: ₹50000.0

Sales by Region:

karnataka: ₹50000.0

1. Add Merchandise
2. Delete Merchandise
3. Record Sale
4. Display Total Revenue
5. Analyze Sales
6. View Merchandise List
7. Exit

Enter your choice: 2

Enter merchandise name to delete: jersey

Merchandise 'jersey' deleted.

1. Add Merchandise
2. Delete Merchandise
3. Record Sale
4. Display Total Revenue
5. Analyze Sales
6. View Merchandise List
7. Exit

Enter your choice: 7

Exiting program.

CONCLUSION

The system effectively manages merchandise, sales tracking, and revenue analysis in a simple and accessible way.

It provides real-time insights into product performance and regional sales trends.

The lack of input validation and data persistence limits its robustness and scalability.

Overall, the program is functional for basic use but would benefit from enhancements to improve usability and long-term data storage.