**CHAPTER-1**

**DAY TO DAY ACTIVITIES**



**Internship Program on Python for BE-3rd Sem students**

**From 9th September to 28h September 2024 (During 3rd semester vacations).**

**Student Name: M CHAITANYA USN No: 3BR23CD052 Branch: CSE-DS**

|  |  |  |  |
| --- | --- | --- | --- |
| **INDEX PAGE** | | | |
| **Day** | **Date** | **Content Covered** | **Signature of the** |
| **faculty in-charge** |
| **1** | **09.09.24** | **Introduction to Python, Setup & Installation, First Python Program, Variables, Data Types,**  **and Basic I/O** |  |
| **2** | **10.09.24** | **Control Structures: If-else, Loops, Functions and Modules** |  |
| **3** | **11.09.24** | **Lists, Tuples, and Dictionaries, File Handling** |  |
| **4** | **12.09.24** | **Exception Handling, Practice exercises on Python basics** |  |
| **5** | **13.09.24** | **Introduction to OOP, Classes, and Objects** |  |
| **6** | **14.09.24** | **Inheritance, Polymorphism, and Encapsulation** |  |
| **7** | **15.09.24** | **Abstract Classes and Interfaces** |  |
| **8** | **17.09.24** | **Practice exercises on OOP concepts** |  |
| **9** | **18.09.24** | **Introduction to DSA, Arrays, and Linked Lists** |  |
| **10** | **19.09.24** | **Introduction to DSA,Aarray And Linkedlist** |  |
| **11** | **20.09.24** | **Introduction to Stack and queues** |  |
| **12** | **21.09.24** | **Practice Exercise on basic concept (Reduce, Lambda function, List Comprehension)** |  |
| **13** | **23.09.24** | **Introduction to Tree Data Structure** |  |
| **14** | **24.09.24** | **Introduction to Graph Data Structure** |  |
| **15** | **25.09.24** | **Searching Algorithms Project Building & Persentations** |  |
| **16** | **26.09.24** | **Project Building & Presentations** |  |
| **17** | **27.09.24** | **Project Building & Presentations** |  |
| **18** | **28.09.24** | **Project Building & Presentations** |  |

**CHAPTER-2**

**COMPANY PROFILE**

**Company Name: EZ Trainings and Technologies Pvt. Ltd.**

**Introduction:**

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

**Mission:**

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

**Services:**

**College Trainings:**

* Tailored training programs designed to enhance the employability of students.
* Industry-aligned curriculum covering technical and soft skills.
* Placement assistance and career guidance.

**Development Projects:**

* End-to-end development services, from ideation to execution.
* Expertise in diverse technologies and frameworks.
* Custom solutions to meet specific business needs.

**Locations:** Hyderabad | Delhi NCR

At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence

**CHAPTER-3**

# ABSTRACT

The Team Merchandising Revenue System is an innovative software solution designed to streamline the management of merchandise and sales transactions within a retail context. With the growing complexity of inventory and sales data in today's competitive market, businesses require efficient tools to facilitate operations and optimize revenue generation. This application empowers users to manage their merchandise effectively by providing functionalities for adding and deleting items, recording sales, and generating insightful analyses of revenue performance.

The system is built using Python and organized around three core classes: `Merchandise`, which encapsulates the properties of individual merchandise items such as name and price; `Sale`, which captures the details of each sales transaction, including the quantity sold and total revenue; and `TeamMerchandisingRevenueSystem`, which acts as the main interface for managing merchandise and sales records. This structured design promotes data integrity and simplifies the user experience by ensuring that all critical functions are easily accessible.

A user-friendly command-line interface allows for interactive engagement, where users can choose from a variety of options to perform tasks related to merchandise and sales management. The system not only tracks sales records but also analyzes revenue by product and region, enabling businesses to identify trends and make informed decisions based on real-time data.

Overall, the Team Merchandising Revenue System presents a comprehensive approach to merchandise management, offering the necessary tools for businesses to enhance their operational efficiency, improve revenue tracking, and derive actionable insights from their sales data. As retail environments continue to evolve, such a system becomes indispensable for any organization aiming to maintain a competitive edge while effectively managing their inventory and sales performance.

# CHAPTER-4

# INTRODUCTION OF THE PROJECT

In an increasingly competitive business landscape, effective merchandise management and sales tracking are paramount for success. Organizations must be able to respond quickly to market demands, optimize inventory, and analyze sales performance to maximize revenue. The Team Merchandising Revenue System is designed to address these challenges by providing a robust and user-friendly platform for managing merchandise and sales transactions.

This application stands out by integrating several essential functionalities that allow users to perform key operations seamlessly. At its core, the system helps users add and delete merchandise items, ensuring that inventory is always up to date. By providing a straightforward method to record sales transactions, it eliminates manual record-keeping errors, enhances data accuracy, and enables real-time tracking of revenue generation.

The necessity of analyzing sales data cannot be overstated. With the ability to break down revenue by product and region, businesses can gain invaluable insights into customer preferences and regional market trends. Such analysis facilitates strategic decision-making, allowing organizations to allocate resources more effectively and tailor their marketing strategies to specific demographics. This is particularly crucial in retail, where understanding which products are performing well in which regions can directly impact inventory decisions and promotional activities.

The system is built using Python, featuring a structured approach that encapsulates the behavior and properties of merchandise and sales. The design revolves around three main classes: `Merchandise`, which captures individual product details; `Sale`, which logs sales transactions and calculates total revenue; and `TeamMerchandisingRevenueSystem`, which acts as the interface for managing these records. This modular design promotes maintainability and allows for future enhancements.

A key aspect of the application is its interactive command-line interface, which provides a user-friendly experience. Users can navigate through various options to manage merchandise and sales without requiring extensive training or technical expertise. This simplicity empowers users to focus on what matters most—driving sales and managing inventory effectively.

In conclusion, the Team Merchandising Revenue System represents a vital tool for any retail organization aiming to improve its operational efficiency and sales performance. As the retail environment continues to evolve with new technologies and shifting consumer behaviors, such a comprehensive system becomes indispensable. By leveraging the capabilities of this application, businesses can not only streamline their merchandise management processes but also gain a competitive advantage through informed decision-making and strategic planning.

# CHAPTER-5

# MODULE DESCRIPTION

The Team Merchandising Revenue System is comprised of several interrelated modules that work together to provide a comprehensive solution for managing merchandise and sales transactions. Each module is designed to fulfill specific functionalities, ensuring that users can efficiently handle various aspects of their operations. Below is an in-depth description of each key module within the system:

1. Merchandise Class

The `Merchandise` class is a foundational component of the system, representing individual merchandise items that a business sells. Each instance of this class encapsulates critical attributes:

- \*\*Name\*\*: This attribute stores the name of the merchandise item, providing a clear identifier that is essential for sales transactions and inventory management.

- \*\*Price\*\*: This attribute holds the selling price of the merchandise, which is vital for calculating revenue during sales transactions. This price can be easily updated to reflect changes in the market or promotional pricing strategies.

The `Merchandise` class allows users to create new merchandise objects with specified attributes, facilitating straightforward management of product listings. It forms the basis for the sales tracking functionality, enabling the system to associate sales records directly with specific merchandise items.

2. Sale Class

The `Sale` class serves as a pivotal element in tracking sales transactions. It includes several key attributes that provide detailed information about each sale:

- \*\*Merchandise\*\*: This attribute links the sale to a specific merchandise item, allowing the system to track which products are being sold.

- \*\*Quantity\*\*: This attribute captures the number of units sold in a transaction, which is essential for understanding sales volume and inventory turnover.

- \*\*Total Price\*\*: This attribute is automatically calculated based on the price of the merchandise and the quantity sold, providing a quick reference for revenue generated by each transaction.

- \*\*Region\*\*: This attribute records the geographical area where the sale occurred, enabling regional analysis of sales performance.

The `Sale` class streamlines the recording of sales transactions, ensuring that all necessary data is captured accurately. This structured approach not only enhances data integrity but also supports effective reporting and analysis.

3. TeamMerchandisingRevenueSystem Class

The `TeamMerchandisingRevenueSystem` class is the heart of the application, integrating the functionalities of both the `Merchandise` and `Sale` classes. It provides a cohesive interface for users to manage their operations effectively. Key methods within this class include:

- \*\*Add Merchandise\*\*: This method allows users to introduce new merchandise items into the system. It enhances inventory management by ensuring that all products are accounted for.

- \*\*Delete Merchandise\*\*: This functionality enables users to remove merchandise from the system, which is crucial for maintaining an accurate inventory, especially when items are discontinued or sold out.

- \*\*Record Sale\*\*: This method allows users to log sales transactions, associating them with specific merchandise and calculating total revenue. It ensures that sales data is accurately captured for future analysis.

- \*\*Total Revenue Calculation\*\*: This method aggregates total revenue from all recorded sales, providing users with an overview of financial performance. This information is critical for assessing the effectiveness of sales strategies.

- \*\*Sales Analysis\*\*: The system provides detailed analysis capabilities, breaking down sales data by product and region. This analysis helps identify trends, assess performance, and make informed business decisions.

- \*\*Merchandise Retrieval\*\*: Users can easily retrieve and view the list of available merchandise, allowing for quick reference and efficient inventory management.

The `TeamMerchandisingRevenueSystem` class encapsulates the entire operational workflow, ensuring that all processes are interconnected and streamlined.

4. Interactive Menu

The interactive command-line interface serves as the front end of the system, enabling users to engage with its functionalities effortlessly. The menu-driven approach allows for easy navigation, presenting users with clear options for various tasks such as adding merchandise, recording sales, and generating reports. This user-centric design minimizes the learning curve, making it accessible even to those with limited technical skills.

Each option within the menu is linked to specific methods in the `TeamMerchandisingRevenueSystem` class, ensuring that user actions trigger appropriate responses in the system. The interface is designed to guide users through their tasks, providing prompts and feedback to enhance the overall experience.

# CHAPTER-6

# ALGORITHM

 **Initialize the System**:

* Create an instance of TeamMerchandisingRevenueSystem.

 **Display Menu**:

* Show the user a list of options (Add Merchandise, Delete Merchandise, Record Sale, etc.).

 **Get User Choice**:

* Accept input from the user to determine which operation to perform.

 **Perform Operations Based on Choice**:

* If choice is 1:
  + Prompt for merchandise name and price.
  + Create Merchandise object and add it to the list.
* If choice is 2:
  + Prompt for merchandise name to delete.
  + Remove the merchandise from the list.
* If choice is 3:
  + Prompt for merchandise name, quantity sold, and region.
  + Check if merchandise exists, create Sale object and record it.
* If choice is 4:
  + Calculate and display total revenue from recorded sales.
* If choice is 5:
  + Analyze sales by product and region and display results.
* If choice is 6:
  + Display the list of merchandise with their prices.
* If choice is 7:
  + Exit the program.
* If invalid choice, prompt user to try again.

 **Loop Until Exit**:

* Repeat the process until the user decides to exit

# CHAPTER-7

# FLOWCHART

[Start]

|

[Initialize System]

|

[Display Menu]

|

[Get User Choice]

|

[Decision: User Choice?]

/ | | | | | \

Choice 1 Choice 2 Choice 3 Choice 4 Choice 5 Choice 6 Choice 7

| | | | | | |

[Add] [Delete] [Record] [Display] [Analyze] [View] [Exit]

| | | | | | |

[Confirm] [Confirm] [Confirm] [Total] [Product] [List] [End]

# CHAPTER-8

# SOURCE CODE

# class Merchandise:

# def \_\_init\_\_(self, name, price):

# self.name = name

# self.price = price

# class Sale:

# def \_\_init\_\_(self, merchandise, quantity, region):

# self.merchandise = merchandise

# self.quantity = quantity

# self.total\_price = merchandise.price \* quantity

# self.region = region

# class TeamMerchandisingRevenueSystem:

# def \_\_init\_\_(self):

# self.merchandise\_list = []

# self.sales\_records = []

# # CRUD operations for merchandise

# def add\_merchandise(self, merchandise):

# self.merchandise\_list.append(merchandise)

# def delete\_merchandise(self, merchandise\_name):

# self.merchandise\_list = [item for item in self.merchandise\_list if item.name != merchandise\_name]

# def get\_merchandise(self):

# return self.merchandise\_list

# # Add a sales record

# def record\_sale(self, sale):

# self.sales\_records.append(sale)

# # Track and display total revenue

# def total\_revenue(self):

# return sum(sale.total\_price for sale in self.sales\_records)

# # Unified analysis by product, store, and region

# def analyze\_sales(self):

# product\_sales = {}

# region\_sales = {}

# for sale in self.sales\_records:

# # Track sales by product

# product = sale.merchandise.name

# if product not in product\_sales:

# product\_sales[product] = 0

# product\_sales[product] += sale.total\_price

# # Track sales by region

# region = sale.region

# if region not in region\_sales:

# region\_sales[region] = 0

# region\_sales[region] += sale.total\_price

# return product\_sales, region\_sales

# # Function to display the menu

# def display\_menu():

# print("\n1. Add Merchandise")

# print("2. Delete Merchandise")

# print("3. Record Sale")

# print("4. Display Total Revenue")

# print("5. Analyze Sales")

# print("6. View Merchandise List")

# print("7. Exit")

# # Main interactive function

# if \_\_name\_\_ == "\_\_main\_\_":

# system = TeamMerchandisingRevenueSystem()

# 

# while True:

# display\_menu()

# choice = int(input("Enter your choice: "))

# 

# if choice == 1:

# # Add Merchandise

# name = input("Enter merchandise name: ")

# price = float(input("Enter merchandise price: "))

# merchandise = Merchandise(name, price)

# system.add\_merchandise(merchandise)

# print(f"Merchandise '{name}' added.")

# 

# elif choice == 2:

# # Delete Merchandise

# name = input("Enter merchandise name to delete: ")

# system.delete\_merchandise(name)

# print(f"Merchandise '{name}' deleted.")

# 

# elif choice == 3:

# # Record Sale

# merchandise\_name = input("Enter merchandise name: ")

# merchandise = next((m for m in system.get\_merchandise() if m.name == merchandise\_name), None)

# if merchandise:

# quantity = int(input("Enter quantity sold: "))

# region = input("Enter region name: ")

# sale = Sale(merchandise, quantity, region)

# system.record\_sale(sale)

# print(f"Sale recorded for {merchandise\_name}.")

# else:

# print(f"Merchandise '{merchandise\_name}' not found.")

# 

# elif choice == 4:

# # Display Total Revenue

# print(f"Total Revenue: ₹{system.total\_revenue()}")

# 

# elif choice == 5:

# # Unified Sales Analysis

# product\_sales, region\_sales = system.analyze\_sales()

# 

# print("\nSales by Product:")

# for product, revenue in product\_sales.items():

# print(f"{product}: ₹{revenue}")

# 

# print("\nSales by Region:")

# for region, revenue in region\_sales.items():

# print(f"{region}: ₹{revenue}")

# 

# elif choice == 6:

# # View Merchandise List

# print("\nMerchandise List:")

# for item in system.get\_merchandise():

# print(f"{item.name} - ₹{item.price}")

# 

# elif choice == 7:

# # Exit

# print("Exiting program.")

# break

# 

# else:

# print("Invalid choice. Please try again.")

# CHAPTER-9

# OUTPUT

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: 6

Merchandise List:

jersey - ₹250.0

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: 6

Merchandise List:

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.

**CHAPTER-10**

# CONCLUSION

The **Team Merchandising Revenue System** is a comprehensive solution designed to manage and analyze merchandise sales effectively. It offers functionalities for adding and deleting merchandise, recording sales, tracking total revenue, and conducting sales analyses by product and region. Here’s a summary of its key features and implications:

1. **User-Friendly Interface**: The interactive menu allows users to navigate various functions easily, making it accessible for both beginners and experienced users.
2. **Merchandise Management**: The system supports CRUD (Create, Read, Update, Delete) operations, allowing for efficient management of merchandise. Users can easily add new items or remove outdated ones, ensuring the merchandise list remains relevant.
3. **Sales Recording**: The ability to record sales transactions provides valuable data for tracking performance. Users can enter sales details, including quantity and region, which enhances the depth of the sales data collected.
4. **Revenue Tracking**: The total revenue feature provides a quick overview of financial performance. This metric is crucial for businesses to understand their financial health and make informed decisions.
5. **Sales Analysis**: The system allows for in-depth analysis of sales data, breaking down revenues by product and region. This information is invaluable for identifying trends, understanding market dynamics, and making strategic decisions to boost sales.
6. **Scalability and Adaptability**: The architecture of the system can be easily expanded to include more features, such as customer management or advanced reporting capabilities. This adaptability ensures that the system can grow with the business's needs.
7. **Impact on Decision-Making**: By providing comprehensive sales data and insights, the system empowers management to make data-driven decisions, optimize inventory, and tailor marketing strategies effectively.

**Future Enhancements**

To further improve the system, consider the following enhancements:

* **User Authentication**: Implement user roles and authentication to ensure data security and privacy.
* **Data Visualization**: Introduce graphs and charts to visualize sales trends and performance metrics, making analysis more intuitive.
* **Integration with E-Commerce Platforms**: Allow integration with online sales channels to automate sales data entry and updates.
* **Detailed Reporting**: Develop customizable reports that can be exported for further analysis or presentations.

**Final Thoughts**

In summary, the Team Merchandising Revenue System is a robust tool for managing merchandise sales. Its design focuses on usability, flexibility, and data analysis, making it an essential resource for businesses looking to optimize their merchandising strategies and improve overall revenue performance. As the business landscape evolves, this system can adapt to meet future challenges and opportunities in the merchandising domain.

# CHAPTER-11

# REFERENCES

* Google
* ChatGPT
* Training