K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

INVENTORY MANAGEMENT FOR MUSICAL INSTRUMENT SHOP

A PROJECT REPORT

Submitted by

MOHAMMED SHAFI M(8115U23ME029)

in partial fulfillment of requirements for the award of the course

MGB1201 – PYTHON PROGRAMMING

In

DEPARTMENT OF MECHANICAL ENGINEERING

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(An Autonomous Institution affiliated to Anna University, Chennai and Approved by AICTE,New Delhi)

SAMAYAPURAM-621 112

DECEMBER- 2024

K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

TRICHY-621 112

BONAFIDE CERTIFICATE

Certified that thisproject report on **“INVENTORY MANAGEMENT FOR MUSICAL INSTRUMENT SHOP”** is the bonafide work of **MOHAMMED SHAFI M(8115U23ME029)**who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE	**SIGNATURE**
Dr. T. M. NITHYA, M.E.,Ph.D.,	**Mrs.S.RAJESWARI M.E.**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
ASSOCIATE PROFESSOR	ASSISTANT PROFESSOR
Department of CSE	Department of CSE
K.Ramakrishnan College of Engineering (Autonomous)	K.Ramakrishnan College of Engineering (Autonomous)
Samayapuram–621112.	Samayapuram–621112.

Submitted for the End Semester Examination held on…………….

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on **“INVENTORY MANAGEMENT FOR MUSICAL INSTRUMENT SHOP”**is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **MGB1201 – PYTHON PROGRAMMING**

Signature

MOHAMMED SHAFI M

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution **“K.Ramakrishnan College of Engineering (Autonomous)”**, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction

.

I whole heartily thanks to **Dr. T. M. NITHYA, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mrs.S.RAJESWARI M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE INSTITUTION

To achieve a prominent position among the top technical institutions

MISSION OF THE INSTITUTION

M1: To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.

M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.

M3: To provide education for developing high-quality professionals to transform the society.

VISION OF THE DEPARTMENT

To create eminent professionals of Computer Science and Engineering by imparting quality education.

MISSION OF THE DEPARTMENT

M1: To provide technical exposure in the field of Computer Science and Engineering through state of the art infrastructure and ethical standards.

M2: To engage the students in research and development activities in the field of Computer Science and Engineering.

M3: To empower the learners to involve in industrial and multi-disciplinary projects for addressing the societal needs.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

Our graduates shall

PEO1: Analyse, design and create innovative products for addressing social needs.

PEO2: Equip themselves for employability, higher studies and research.

PEO3: Nurture the leadership qualities and entrepreneurial skills for their successfulcareer.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Apply the basic and advanced knowledge in developing software, hardware and firmware solutions addressing real life problems.

- **PSO2:** Design, develop, test and implement product-based solutions for their career enhancement.

ABSTRACT

Inventory management is key to running a successful musical instrument shop. It involves organizing and tracking a wide range of products, such as guitars, pianos, drums, accessories, and sheet music. Since demand for these items can vary by season, trends, and customer preferences, managing inventory effectively helps ensure that popular items are always in stock, while minimizing excess or unsold inventory.

This paper explores the importance of good inventory management in a musical instrument shop, focusing on how to balance stock levels, reduce costs, and meet customer needs. It highlights basic strategies like ordering items at the right time, using software to track inventory, and keeping an eye on trends. With the right tools and methods, a shop can improve its sales, reduce waste, and ensure that customers have the products they want when they need them.

In summary, effective inventory management is essential for keeping a musical instrument shop running smoothly, increasing profits, and improving customer satisfaction.

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
Inventory management is key to running a successful musical instrument shop. It involves organizing and tracking a wide range of products, such as guitars, pianos, drums, accessories, and sheet music. Since demand for these items can vary by season, trends, and customer preferences, managing inventory effectively helps ensure that popular items are always in stock, while minimizing excess or unsold inventory. This paper explores the importance of good inventory management in a musical instrument shop, focusing on how to balance stock levels, reduce costs, and meet customer needs. It highlights basic strategies like ordering items at the right time, using software to track inventory, and keeping an eye on trends. With the right tools and methods, a shop can improve its sales, reduce waste, and ensure that customers have the products they want when they need them. In summary, effective inventory management is essential for keeping a musical instrument shop running smoothly, increasing profits, and improving customer satisfaction.	**PO1 - 1** **PO2 - 2** **PO3 - 1** **PO12 -2**	**PSO1 - 2**

Note: 1- Low, 2-Medium, 3- High

SUPERVISORHEAD OF THE DEPARTMENT

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of inventory management for a musical instrument shop is to ensure a seamless balance between product availability and operational efficiency while meeting customer demands. Effective inventory management aims to maintain optimal stock levels, ensuring popular instruments, accessories, and spare parts are always available without overstocking or incurring excessive holding costs. It minimizes the risks of stockouts that can result in lost sales and customer dissatisfaction, as well as prevents obsolescence of slow-moving items.

By leveraging data-driven insights, inventory management allows the shop to forecast demand, identify sales trends, and adapt to seasonal or market changes. It also supports efficient ordering and replenishment processes, reducing manual errors and enhancing operational productivity. A well-organized inventory ensures the shop can cater to diverse customer needs, from beginners to professionals, while optimizing storage space and maintaining product quality.

Ultimately, the goal is to create a system that aligns inventory levels with business objectives, supports promotional campaigns, and enhances customer satisfaction, driving overall profitability and growth.

1.2 Overview

Python is a versatile and powerful programming language that can be used to develop an efficient inventory management system for a musical instrument shop. By leveraging Python, you can create tools to monitor stock levels, generate purchase orders, track sales, and forecast demand. Python’s libraries and frameworks, such as Pandas, NumPy, SQLite3, and Matplotlib, enable the development of data-driven applications to handle inventory data effectively.

1.3 Python Programming Concepts

1. Dictionaries

The inventory dictionary is used to store the inventory items. Each item is represented as a key (the name of the item), and its details (quantity and price) are stored as a nested dictionary.

2. Functions

Functions are used to encapsulate operations on the inventory, making the code reusable and modular.

Functions in the program:

Add_item: Adds an item to the inventory.

Remove_item: Removes an item from the inventory.

Update_item: Updates the quantity and/or price of an item.

Show_inventory: Displays all items in the inventory.

3. Default Parameters

The update_item function uses default parameters (quantity=None, price=None) to allow updating either the quantity, the price, or both.

4. Conditional Statements

If-else statements are used to check conditions, such as whether an item exists in the inventory or whether the inventory is empty.

5. Loops

A for loop is used in the show_inventory function to iterate through the dictionary and display all items.

6. String Formatting

The program uses f-strings to format output messages dynamically.

7. Dynamic Data Updates

The program updates the dictionary dynamically by modifying or removing items based on user input or function calls.

8. Code Reusability

Functions make the code reusable, allowing operations like adding, removing, and updating items without rewriting logic.

9. Basic Program Flow

- items to the inventory.
- Displaying the inventory.
- Updating and removing items.
- Displaying the updated inventory.

CHAPTER 2

PROJECT METHODOLOGY

2.1Proposed Work

- Efficiently manage the inventory of musical instruments.
- Allow the addition, deletion, and updating of inventory items.
- Provide functionality to view the current stock and instrument details.
- Ensure data persistence using a database or file storage.

1. **Requirement Analysis:**

- Identify the types of instruments (e.g., guitars, pianos, drums).
- Define the attributes for each item (e.g., name, category, price, stock).

2. **System Design:**

- Develop a menu-driven interface for user interaction.
- Create functions for inventory operations (add, update, delete, view)**.**

3. **Implementation:**

- Use Python for coding the inventory system.
- Employ dictionaries, classes, or database integration for data storage.

4. **Testing:**

- Test functionalities for different inputs (e.g., invalid inputs, missing data).
- Validate the system’s ability to handle edge cases.

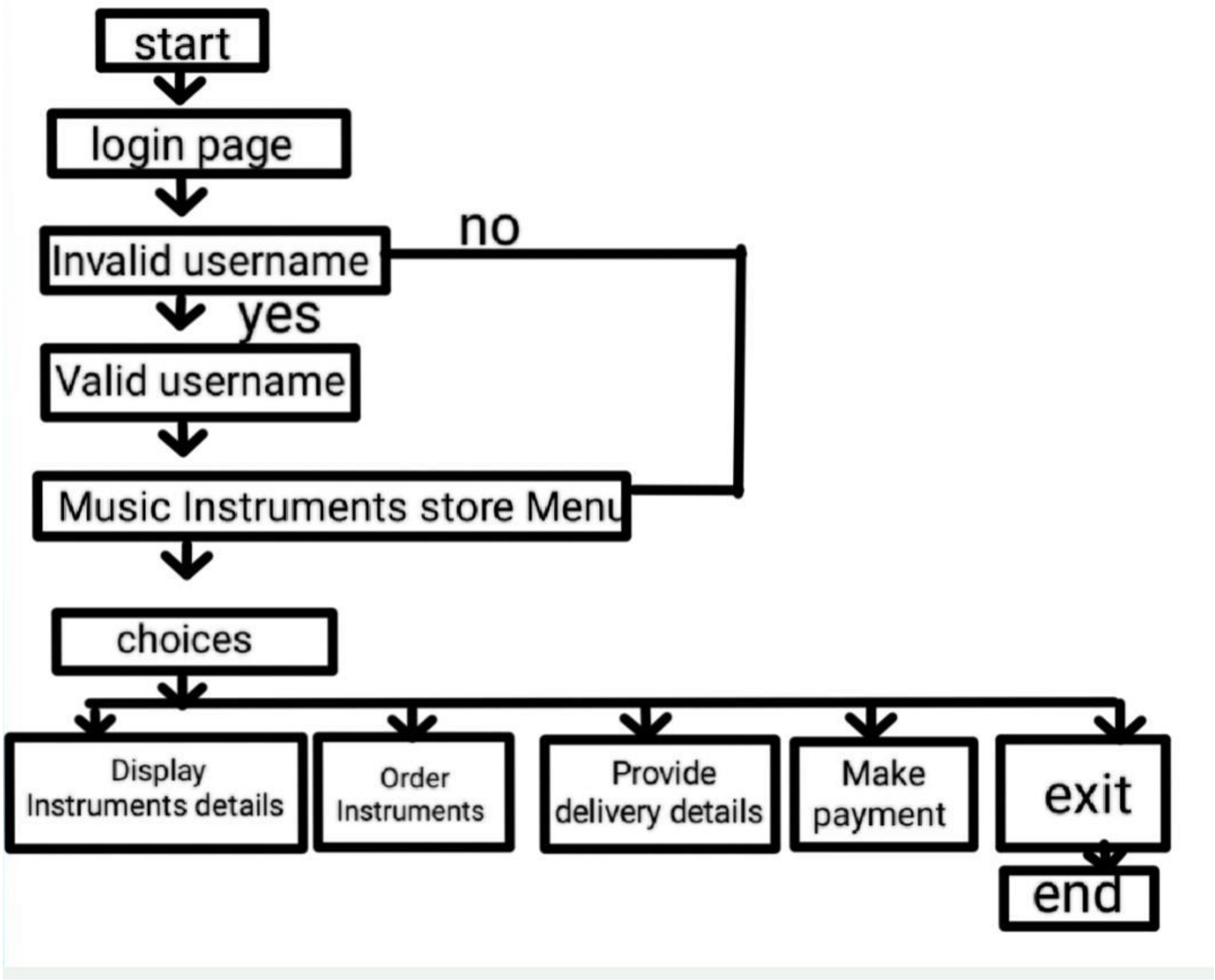
5. **Data Persistence:**

Store inventory data in a file (CSV/JSON) or a database (e.g., SQLite).

6. **Future Enhancements:**

- Add sales tracking, graphical inventory analysis, and multi-user support.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1. Add_item(name, quantity, price)

- Adds an instrument to the inventory with its quantity and price.
- Updates the inventory dictionary with the new item.

3.2. Remove_item(name)

- Deletes an instrument from the inventory if it exists.
- Prints a success or error message based on the item’s presence.

3.3. Update_item(name, quantity=None, price=None)

- Updates the quantity and/or price of an existing item in the inventory.
- Ensures the item exists before updating and handles invalid entries.

3.4. Show_inventory()

- Displays all the instruments in the inventory.
- Lists the instrument name, quantity, and price. If the inventory is empty, it notifies the user.

MODULE FUNCTION

Module 1: Instrument Catalog

- Centralized repository to store and manage details of all instruments
- Search and filter functionality to easily access specific instruments.

Module 2: Inventory Management

- Tracks the stock levels of instruments.
- Features include stock-in, stock-out, reorder level alerts, and inventory audit trails.

Module 3: Order Management

- Facilitates creation, tracking, and processing of orders.
- Includes order history, status updates, and notifications for pending or completed orders.

Module 4: Supplier Management

- Maintains supplier profiles and contact information.
- Supports supplier evaluations, contract management, and communication tracking.

Module 5: Reporting and Analytics

- Generates reports on inventory levels, order history, supplier performance, a.
- Includes data visualization tools like charts, graphs, and dashboards.

Module 6: User Management

- Handles user roles, permissions, and authentication.
- Provides secure access control and activity logs.

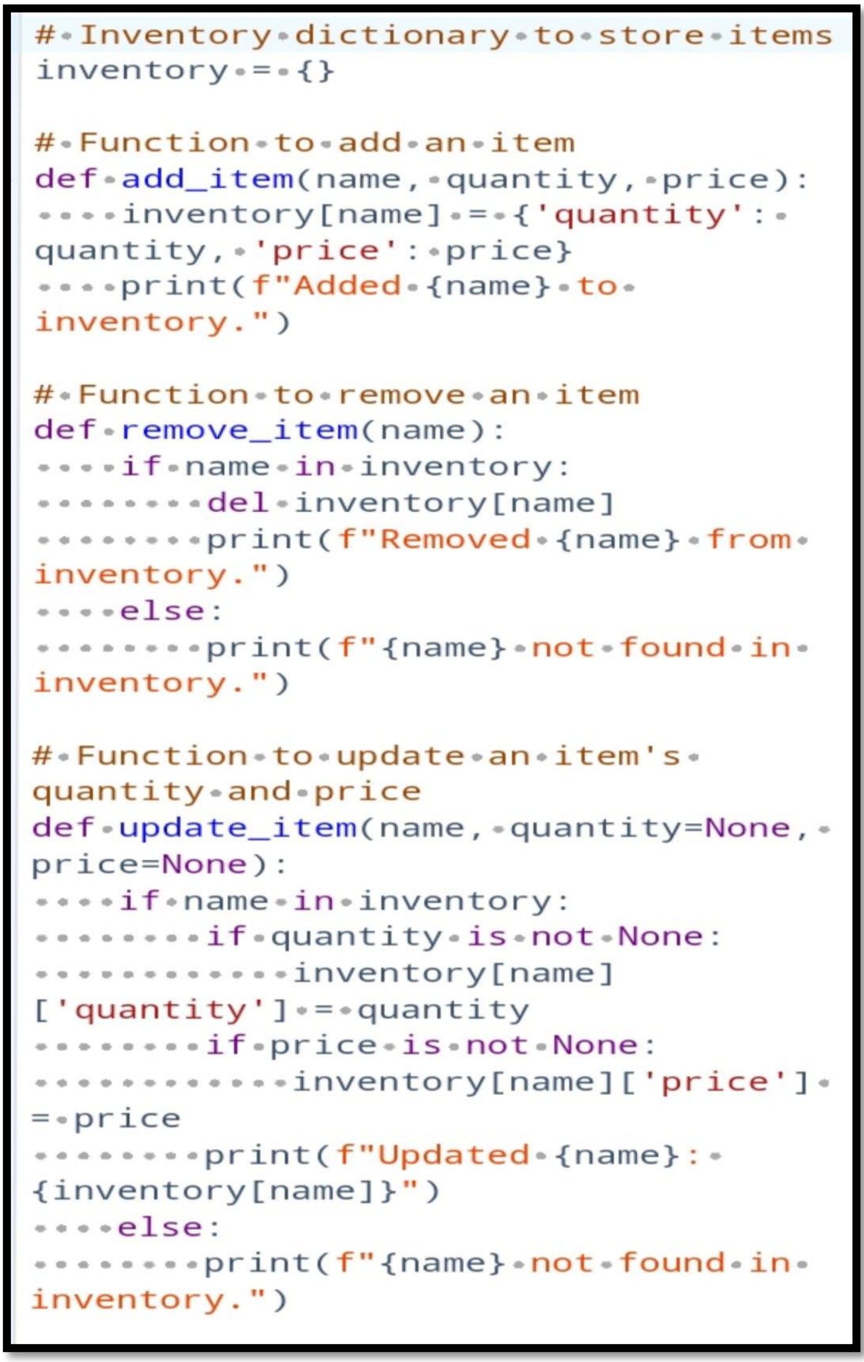
Module 7: Integration

- Connects the system with platforms such as ERP, CRM, or e-commerce.

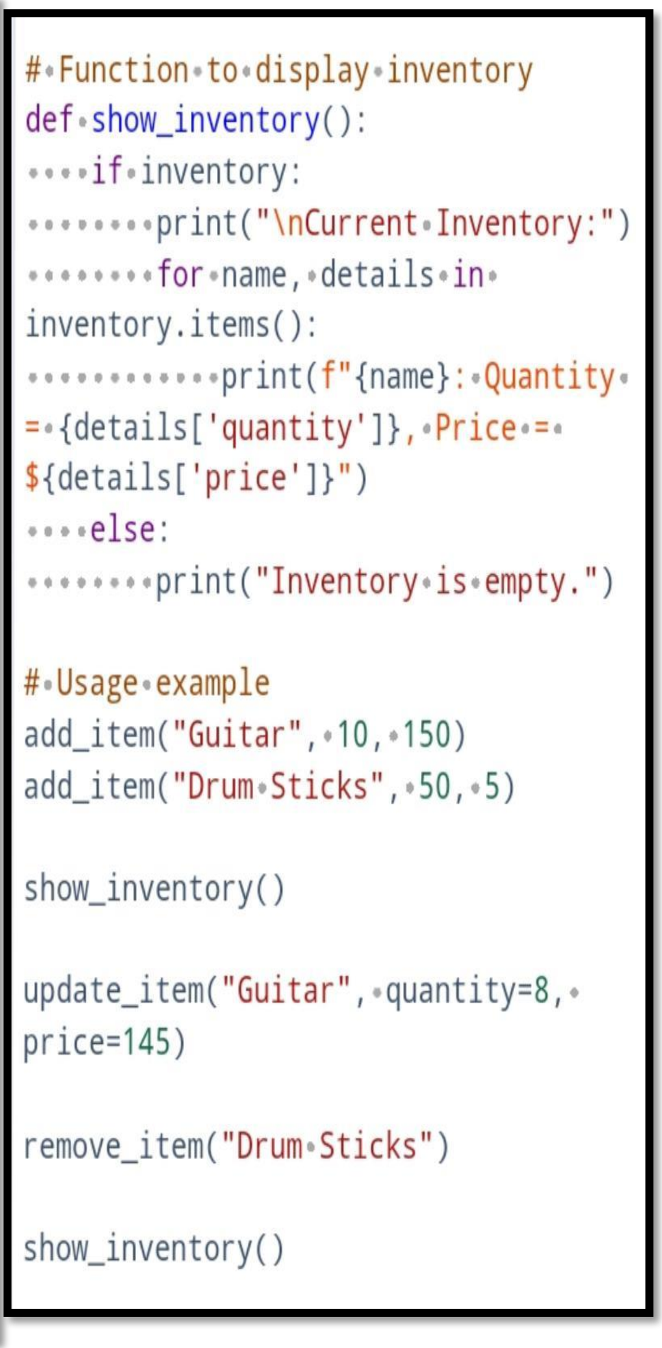
CHAPTER 4

RESULTS AND DISCUSSION

PROGRAM

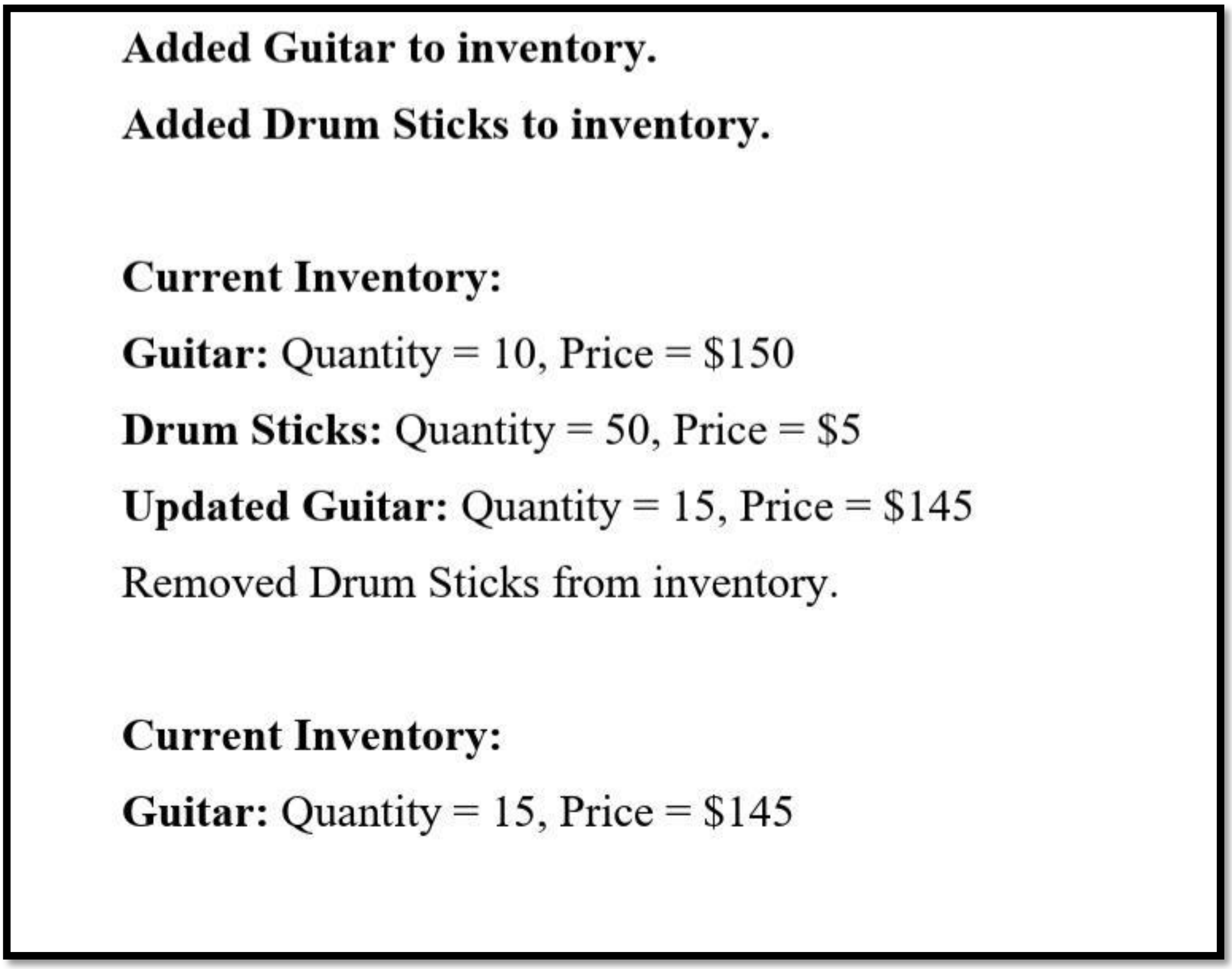


```
# Inventory dictionary to store items
inventory = {}

# Function to add an item
def add_item(name, quantity, price):
    inventory[name] = {'quantity': 
quantity, 'price': price}
    print(f"Added {name} to 
inventory.")

# Function to remove an item
def remove_item(name):
    if name in inventory:
        del inventory[name]
        print(f"Removed {name} from 
inventory.")
    else:
        print(f"{name} not found in 
inventory.")

# Function to update an item's 
quantity and price
def update_item(name, quantity=None, 
price=None):
    if name in inventory:
        if quantity is not None:
            inventory[name]
['quantity'] = quantity
        if price is not None:
            inventory[name]['price'] 
= price
        print(f"Updated {name}: 
{inventory[name]}")
    else:
        print(f"{name} not found in 
inventory.")
```



```
# Function to display inventory
def show_inventory():
    if inventory:
        print("\nCurrent Inventory:")
        for name, details in 
inventory.items():
            print(f"{name}: Quantity 
= {details['quantity']}, Price = 
${details['price']}")
    else:
        print("Inventory is empty.")

# Usage example
add_item("Guitar", 10, 150)
add_item("Drum Sticks", 50, 5)

show_inventory()

update_item("Guitar", quantity=8, 
price=145)

remove_item("Drum Sticks")

show_inventory()
```

OUTPUT



```
Added Guitar to inventory.
Added Drum Sticks to inventory.

Current Inventory:
Guitar: Quantity = 10, Price = $150
Drum Sticks: Quantity = 50, Price = $5
Updated Guitar: Quantity = 15, Price = $145
Removed Drum Sticks from inventory.

Current Inventory:
Guitar: Quantity = 15, Price = $145
```

CHAPTER 5

CONCLUSION

In conclusion, effective inventory management for a musical instrument shop is crucial for maintaining a balanced stock, reducing overhead costs, and ensuring customer satisfaction. By implementing strategies such as demand forecasting, categorizing instruments based on popularity and seasonality, and utilizing inventory management software, the shop can minimize stockouts and overstock situations. Regular stock audits, as well as establishing strong supplier relationships, can also help maintain a smooth flow of products. Additionally, offering a diverse range of products and keeping track of sales trends can help the shop make informed decisions about purchasing and stocking items, ultimately boosting profitability and customer loyalty.

REFERENCES:

1. "Learn Python Programming" on w3schools.com– This online tutorial provides interactive lessons and examples for learning Python programming from scratch.
2. "Introduction to Python Programming" on Coursera.org – This online course provides a structured introduction to Python programming, covering basic concepts and practical exercises.
3. "Python.org Documentation" – The official Python documentation website offers comprehensive information on Python programming language features, syntax, and usage.
4. "Inventory Management: Principles, Concepts, and Techniques" by John Toomey
5. "The Art of Inventory Management for Small Businesses" by Tim Berry (Entrepreneur.com)
6. "Inventory Management Best Practices" by Colin O'Rourke
7. "Retail Inventory Management" by American Retail Supply
8. Shopify Blog – "Inventory Management for Small Businesses"
9. "Inventory Control for Musical Instrument Retailers" by Sound Retail Solutions
10. Chatgpt

APPENDIX

(Coding)

```
# Inventory dictionary to store items
Inventory = {}

# Function to add an item
Def add_item(name, quantity, price):
    Inventory[name] = {'quantity': quantity, 'price': price}
    Print(f'Added {name} to inventory.")

# Function to remove an item
Def remove_item(name):
    If name in inventory:
        Del inventory[name]
        Print(f'Removed {name} from inventory.")
    Else:
        Print(f"{name} not found in inventory.")

# Function to update an item's quantity and price
Def update_item(name, quantity=None, price=None):
    If name in inventory:
        If quantity is not None:
            Inventory[name]['quantity'] = quantity
        If price is not None:
            Inventory[name]['price'] = price
        Print(f'Updated {name}: Quantity = {inventory[name]['quantity']}, Price = ${inventory[name]['price']}")
    Else:
        Print(f"{name} not found in inventory.")

# Function to display inventory
Def show_inventory():
    If inventory:
        Print("\nCurrent Inventory:")
        For name, details in inventory.items():
            Print(f"{name}: Quantity = {details['quantity']}, Price = ${details['price']}")
    Else:
        Print("Inventory is empty.")

# Usage example
Add_item("Guitar", 10, 150)
Add_item("Drum Sticks", 50, 5)
Show_inventory()
Update_item("Guitar", quantity=15, price=145)
Remove_item("Drum Sticks")
Show_inventory()
```

APPENDIX

(OUTPUT)

```
Added Guitar to inventory.
Added Drum Sticks to inventory.

Current Inventory:
Guitar: Quantity = 10, Price = $150
Drum Sticks: Quantity = 50, Price = $5
Updated Guitar: Quantity = 15, Price = $145
Removed Drum Sticks from inventory.

Current Inventory:
Guitar: Quantity = 15, Price = $145
```