

S**ubject: CSE228(Data Structures)**

**Project Name: Inventory Management System**

**Name: Ruthvik Chandra**

**Regd No: 12204171**

**Roll No: 29**

**Section: K22UP**

**Submitted to: Waseen Ud Din Wani(63869)**

**Declaration:**

I, Ruthvik Chandra, hereby declare that this project report titled "Inventory Management System" represents my original work. All the content, research, analysis, and conclusions presented in this document are the result of my own effort and have not been previously submitted for any academic or professional evaluation.

**I further affirm that:**

The "Inventory Management System" project was developed based on the description provided, with the functionalities of managing inventory, adding items, updating quantities, and generating reports.

Any assistance or guidance received during the project is appropriately acknowledged in the acknowledgment section.

The report adheres to the formatting guidelines, including font, font size, headings, and line spacing, as specified in the project requirements.

I understand that any deviation from these principles may result in academic or professional consequences.

Date: 23-10-23

[Ruthvik Chandra]

**Acknowledgement**

I would like to express my sincere gratitude to all those who contributed to the successful completion of the "Inventory Management System" project. Their support and assistance were invaluable throughout the project's development.

I extend my heartfelt thanks to:

- Wassen Ud Din wani(638669) for their guidance, mentorship, and expertise in steering this project in the right direction.

The collaboration and assistance received from the above-mentioned individuals played a pivotal role in shaping the "Inventory Management System" project into a reality. I acknowledge and appreciate their contributions and support.

[Ruthvik Chandra]

**Abstract:**

The "Inventory Management System" is a software program designed for effectively managing inventory in various settings. Users can add, update, and generate reports for items in the inventory.

The system offers an intuitive and user-friendly interface for users to interact with inventory data, making it an ideal tool for businesses, warehouses, and retail operations.

With a robust data management system, the program ensures accurate and efficient inventory tracking and reporting, enhancing operational efficiency.

**Introduction:**

The "Inventory Management System" is a comprehensive software solution that simplifies the management of inventory. It allows users to add items, update quantities, and generate reports for efficient inventory control.

The system is designed to cater to the needs of various businesses, from retail stores to warehouses, by providing an easy-to-use interface. It manages inventory data efficiently, ensuring that businesses can track stock levels accurately and generate essential reports.

This program also includes robust error handling to detect issues such as adding duplicate items or invalid input and provides informative messages to guide users.

In summary, the "Inventory Management System" streamlines the process of managing and controlling inventory, making it an essential tool for businesses of all sizes.

**Objectives and Scope of the Project**

The primary objectives of the project include:

1. Developing a user-friendly inventory management system.

2.Implementing core features such as adding items, updating quantities, generating reports, and inventory value calculations.

3.Ensuring robust data management and error handling for reliable inventory tracking.

The scope of the project encompasses the creation of a functional inventory management system with an easy-to-use interface.

**Methodology/Flowchart or Algorithm Implementation:**

The project employs structured methodologies to ensure efficient inventory management. The algorithm consists of the following steps:

1. User can add items to the inventory, specifying the item name, quantity, and price.

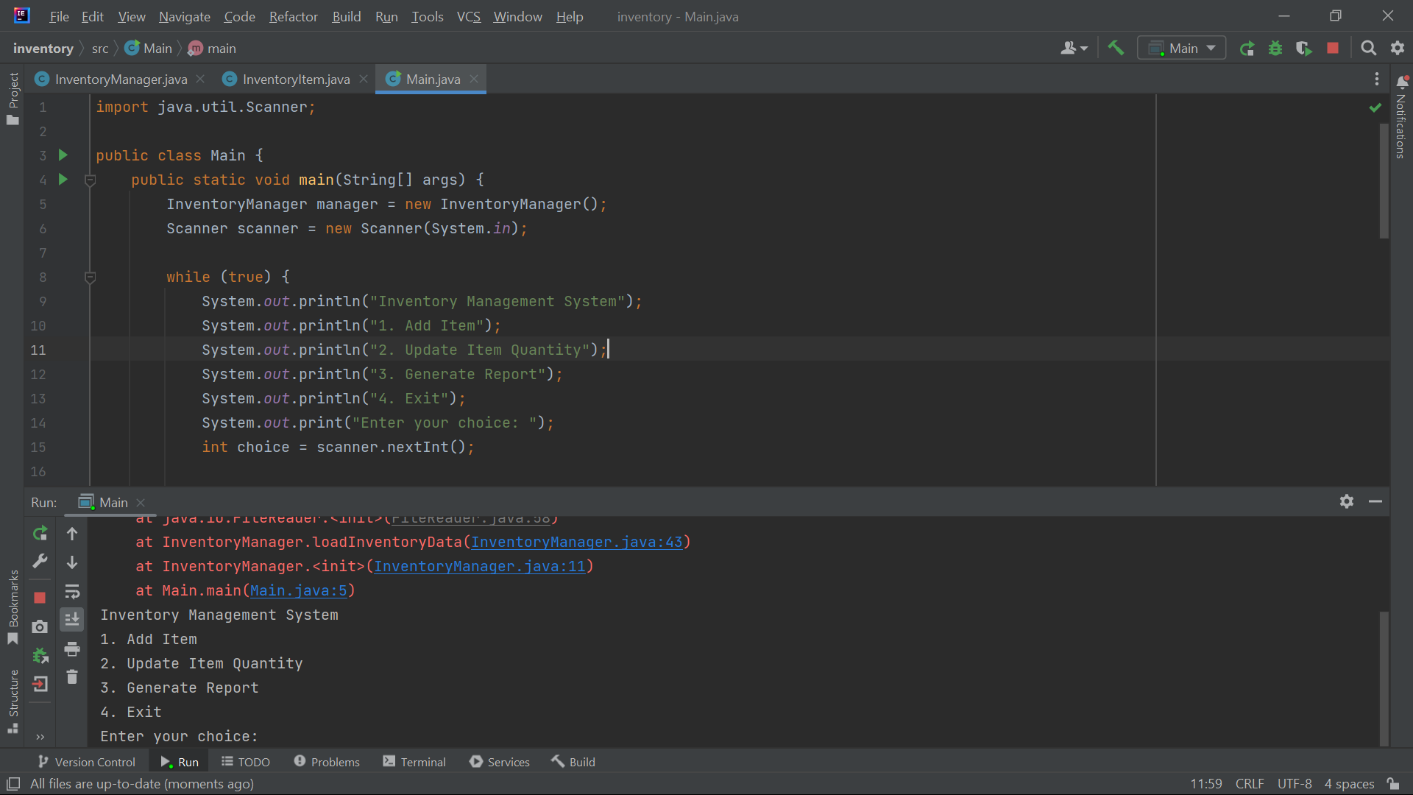
2. User can update item quantities.

3. User can generate reports based on current inventory data, including item names, quantities, and their total value.

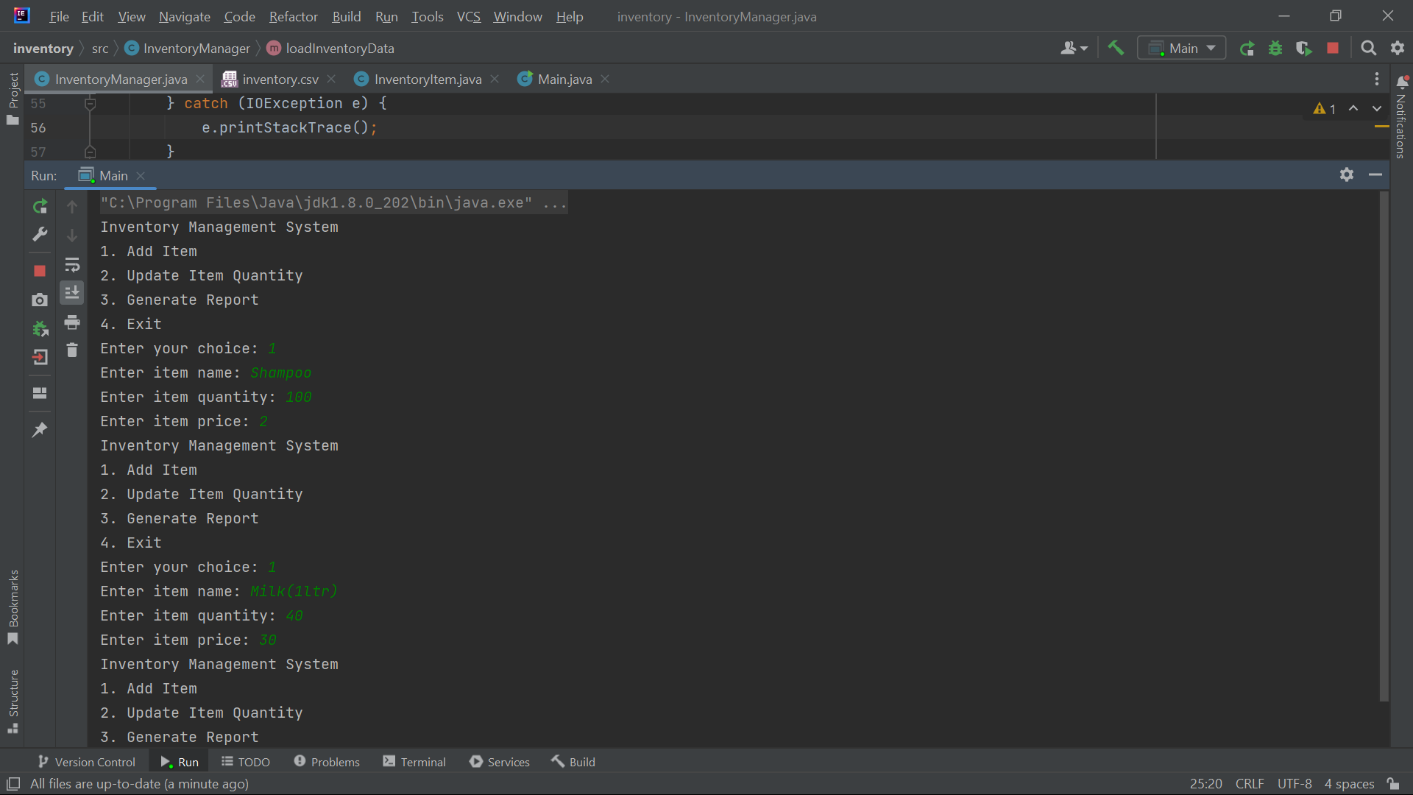
4. Robust error handling is implemented to catch common issues, such as adding duplicate items, invalid input, or accessing items that do not exist.

**ScreenShots**

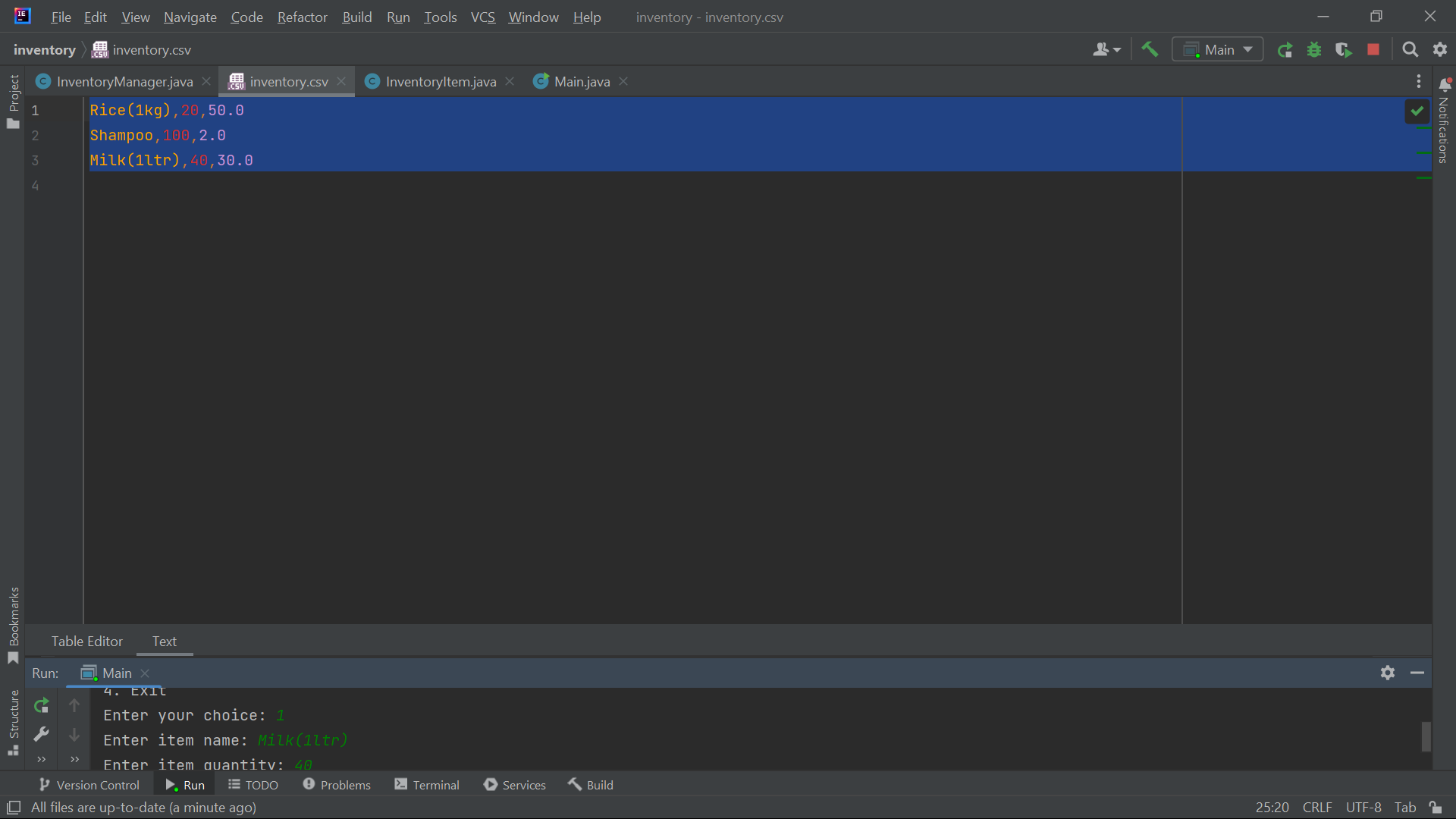
On running the code



**On choosing 1st option**



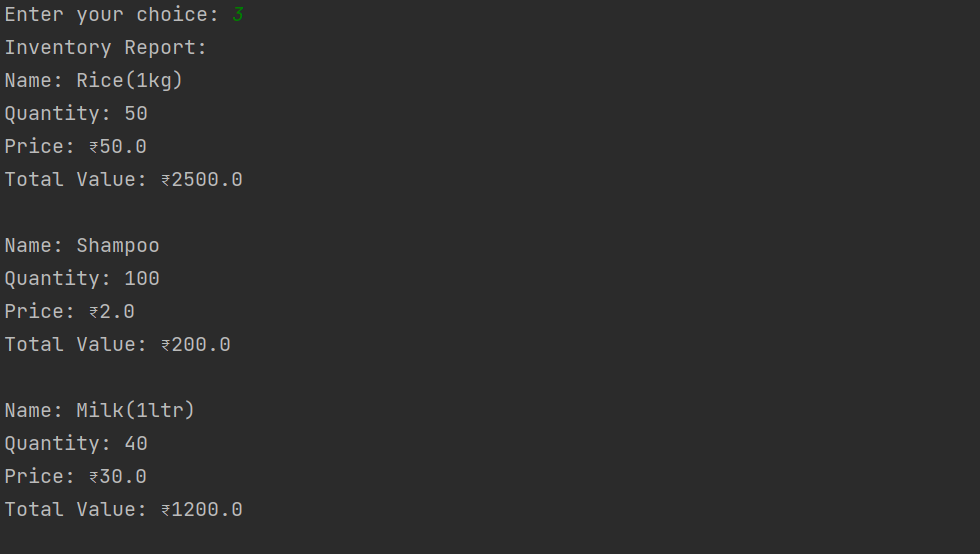
**The list will be saved in the csv file**



**On Choosing the 2nd option**



**On Choosing the 3rd option**



**On choosing the 4th option**



**Source code:**

**Main class**

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 InventoryManager manager = new InventoryManager();  
 Scanner scanner = new Scanner(System.*in*);  
  
 while (true) {  
 System.*out*.println("Inventory Management System");  
 System.*out*.println("1. Add Item");  
 System.*out*.println("2. Update Item Quantity");  
 System.*out*.println("3. Generate Report");  
 System.*out*.println("4. Exit");  
 System.*out*.print("Enter your choice: ");  
 int choice = scanner.nextInt();  
  
 switch (choice) {  
 case 1:  
 System.*out*.print("Enter item name: ");  
 String name = scanner.next();  
 System.*out*.print("Enter item quantity: ");  
 int quantity = scanner.nextInt();  
 System.*out*.print("Enter item price: ");  
 double price = scanner.nextDouble();  
 manager.addItem(name, quantity, price);  
 break;  
 case 2:  
 System.*out*.print("Enter item name: ");  
 name = scanner.next();  
 System.*out*.print("Enter new quantity: ");  
 quantity = scanner.nextInt();  
 manager.updateItemQuantity(name, quantity);  
 break;  
 case 3:  
 manager.generateReport();  
 break;  
 case 4:  
 System.*exit*(0);  
 default:  
 System.*out*.println("Invalid choice. Try again.");  
 }  
 }  
 }  
}

**Inventory Manager Class**

import java.io.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class InventoryManager {  
 private List<InventoryItem> inventory;  
 private static final String *INVENTORY\_FILE* = "inventory.csv";  
  
 public InventoryManager() {  
 inventory = new ArrayList<>();  
 loadInventoryData();  
 }  
  
 public void addItem(String name, int quantity, double price) {  
 InventoryItem item = new InventoryItem(name, quantity, price);  
 inventory.add(item);  
 saveInventoryData();  
 }  
  
 public void updateItemQuantity(String name, int quantity) {  
 for (InventoryItem item : inventory) {  
 if (item.getName().equals(name)) {  
 item.setQuantity(quantity);  
 saveInventoryData();  
 return;  
 }  
 }  
 System.*out*.println("Item not found in inventory.");  
 }  
  
 public void generateReport() {  
 System.*out*.println("Inventory Report:");  
 for (InventoryItem item : inventory) {  
 System.*out*.println("Name: " + item.getName());  
 System.*out*.println("Quantity: " + item.getQuantity());  
 System.*out*.println("Price: $" + item.getPrice());  
 System.*out*.println("Total Value: $" + item.getQuantity() \* item.getPrice());  
 System.*out*.println();  
 }  
 }  
  
 private void loadInventoryData() {  
 try (BufferedReader reader = new BufferedReader(new FileReader(*INVENTORY\_FILE*))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(",");  
 if (parts.length == 3) {  
 String name = parts[0];  
 int quantity = Integer.*parseInt*(parts[1]);  
 double price = Double.*parseDouble*(parts[2]);  
 InventoryItem item = new InventoryItem(name, quantity, price);  
 inventory.add(item);  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void saveInventoryData() {  
 try (PrintWriter writer = new PrintWriter(new FileWriter(*INVENTORY\_FILE*))) {  
 for (InventoryItem item : inventory) {  
 writer.println(item.getName() + "," + item.getQuantity() + "," + item.getPrice());  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

**Class InventoryItem**

public class InventoryItem {  
 private String name;  
 private int quantity;  
 private double price;  
  
 public InventoryItem(String name, int quantity, double price) {  
 this.name = name;  
 this.quantity = quantity;  
 this.price = price;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
  
 public void setQuantity(int quantity) {  
 this.quantity = quantity;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double price) {  
 this.price = price;  
 }  
}

**Features:**

1. \*\*Item Management:\*\* The "Inventory Management System" offers robust item management capabilities. Users can easily add new items to the inventory, including specifying the item name, quantity, and price.

2. \*\*Quantity Updates\*\* The system allows users to update item quantities. This feature is critical for keeping the inventory data accurate and up-to-date. Users can increase or decrease item quantities as stock levels change.

3. \*\*Report Generation\*\* The program supports the generation of detailed inventory reports. Users can request reports that provide a comprehensive overview of all items in the inventory. These reports include item names, quantities, and their total value.

4. \*\*User-Friendly Interface\*\* The system offers a user-friendly command-line interface that is easy to navigate. Users can interact with the program without the need for extensive technical knowledge.

5. \*\*Data Validation\*\* Robust error handling is a fundamental feature of this system. It detects and gracefully handles issues such as:

- Adding items with the same name to prevent duplicates.

- Invalid input, ensuring that only valid numeric values are accepted.

- Attempts to access items that do not exist in the inventory.

6. \*\*Data Accuracy\*\* The program guarantees data accuracy. It ensures that all inventory items are uniquely identified, preventing the addition of duplicate items with the same name. This commitment to data accuracy is essential for making informed business decisions.

7. \*\*Inventory Value Calculation\*\* For each item in the inventory, the system calculates its total value by multiplying the quantity by the price