INTERNSHIP PROJECT-8 DOCUMENT

ON "E-COMMERCE CART SIMULATION (JAVA- GUI)"

Submitted by:

Navpreet Singh (INTERN)

Submitted To:-

Kanduri Abhinay (FOUNDER)

RITHIN VERMA (CTO)

INDEX

INTRODUCTION:-

The E-commerce Cart Simulation project models a simple online shopping cart where users can select products and specify quantities to purchase. The program mimics the behavior of an e-commerce platform's cart functionality, calculating the final bill based on user input. It uses object-oriented programming concepts by implementing product and cart classes, and employs collections to manage multiple items efficiently. This simulation helps understand key software development principles and provides a basis for real-world application development.

SOFTWARE REQUIREMENTS:-

The project is implemented in Java, requiring the Java Development Kit (JDK) for compilation and execution. A text editor or an Integrated Development Environment (IDE) such as IntelliJ IDEA or Eclipse is recommended for writing and debugging code. The program uses standard Java libraries for input handling (java.util.Scanner), collections (java.util.ArrayList), and file handling (java.io.BufferedWriter, java.io.FileWriter) to save the final bill.

DESIGN:-

The system is designed around two core classes: Product and Cart. The Product class encapsulates the product's attributes, including name, price, and quantity. The Cart class manages a collection of products added by the user. The design follows a modular approach where each class handles its responsibility, improving maintainability and extensibility. File saving is handled separately to persist purchase details, enhancing usability.

INPUT:-

```
Welcome to E-commerce Cart Simulation
Product List:
1. Laptop - ₹50000.00
2. Smartphone - ₹20000.00
3. Headphones - ₹1500.00
4. Keyboard - ₹700.00
5. Mouse - ₹500.00
Enter product number to add to cart (0 to finish):
```

IMPLEMENTATION:-

- Created a Product class to encapsulate product name, price, and quantity.
- Created a Cart class to manage a collection of products selected by the user.

- Provided a predefined catalogue of products with prices.
- Used Scanner to take user input for product selection and quantities.
- Added input validation for correct product number and positive quantities.
- Calculated individual product total and overall final bill dynamically.
- Displayed a detailed final bill including product list, quantities, unit prices, and totals.
- Used BufferedWriter and FileWriter to persist final bill details to a text file named "FinalBill.txt".
- Ensured resource management with try-with-resources for file operations.
- Closed Scanner object to release system resources.

TESTING:-

- Tested valid product selection and quantity input.
- Verified accurate calculation of product subtotals and final bill.
- Checked handling of invalid product numbers and non-positive quantities.
- Tested repeated addition of different products to the cart.
- Confirmed generated bill output matches input quantities and calculations.
- Validated proper file saving of the bill with correct content.
- Tested program behavior on file write errors.
- Tested program termination by entering zero (finish ordering).

ADVANTAGES:-

- Demonstrates core Java programming concepts: classes, objects, collections.
- Provides practical experience with console input and output.
- Introduces file handling for data persistence in Java applications.
- Illustrates modular and scalable design principles.
- Validates user input ensuring smooth user experience.
- Can be easily extended to support more complex e-commerce features.
- Serves as foundational knowledge for real-world web/app platform development.

CONCLUSION:-

The E-commerce Cart Simulation successfully demonstrates the essential components of building an online shopping experience. By implementing classes, collections, and persistent file storage, the project integrates fundamental Java concepts with practical application development. This foundational project is a stepping stone for more complex e-commerce systems and provides valuable hands-on experience for aspiring software developers.

REFERENCES:-

Oracle Java SE Documentation

Oracle Java Tutorials - Collections

Oracle Java Tutorials - Basic I/O

Bloch, Joshua. Effective Java. Addison-Wesley Professional, 2018.

Java File I/O Overview

Java Scanner Class - Oracle