Online Shopping Cart System in C

1. Introduction:

The C-based Online Shopping Cart System is a command line application meant to fill the gap between offline and online sales and online ease for small enterprises. As an educational project, it illustrates how fundamental C programming principles—such as file handling, structures, and dynamic memory can generate functional applications. In contrast to the internet dependent web-based systems, this minimalistic solution runs offline, hence suitable for local traders with minimal technical facilities.
This solution solves these issues by offering:
□ The project has two purposes: as an educational platform for learning C programming basics, and as an affordable digital improvement tool for micro enterprises. Its modularity enables future extension With the addition of GUI interfaces (through GTK) or multi-user support.

2. Problem Statement:

Many students often struggle with:
1. Applying theoretical concepts to real projects
2. Understanding program flow in complete systems
3. Managing data without databases
4. Creating interactive menus with user input
5. Our solution provides:

✔ Hands-on experience with arrays, structures, and functions
✔ Visual program flow through simple algorithms
✔ File-based data persistence (optional)
✔ Menu-driven interface for better UX

Algorithm:

```
DO:
  1. Display Menu:
      [1] View Products
      [2] View Cart
      [3] Add to Cart
      [4] Remove from Cart
      [5] Checkout
      [6] Exit

  2. Read user choice (input).

  3. SWITCH (choice):
      CASE 1:
          - Display all products (loop through `products[]`).
          - Format: ID | Name | Price.

      CASE 2:
          - IF cart is empty:
              Print "Cart is empty."
          ELSE:
              - Display cart items (ID, Name, Quantity, Subtotal).
              - Calculate/display grand total.

      CASE 3:
          - Prompt for Product ID and Quantity.
          - Validate ID exists and Quantity > 0.
          - Add item to cart (update quantity if already present).




      CASE 4:
          - Prompt for Product ID to remove.
          - IF item exists in cart:
              Remove it or decrement quantity.
          ELSE:
              Print "Item not found in cart."

      CASE 5:
          - Calculate total cost.
          - Generate receipt (print to console/file).
          - Optional: Apply discounts (e.g., 10% off for totals > $50).
          - Clear cart after checkout.

      CASE 6:
          - Exit program.

      DEFAULT:
          - Print "Invalid choice. Try again."

WHILE (choice != 6)
```

4. Features:

Core features(must have)

□ Product Catalog

1) Predefined list of 5-10 products
2) Display with IDs, names, prices

□ Shopping Cart

1) Add/remove items
2) Quantity adjustment
3) Real-time total calculation

□ User Interference:

1) Numbered menu system
2) Clear error messages
3) Simple navigation

□ Learning Extensions(Optional)

1) File operations- save / load products
2) Search Function- find product by name
3) Receipt Generation- print order summary
4) Discount system-percentage based offers

5. Objectives:

1. Understand Fundamental Concepts
□ Arrays and structures
□ Functions and modular programming
□ Control flow and loops

2. Develop Practical Skills
□ Menu system implementation
□ User input handling
□ Basic data management
3. Build Confidence

□ Complete a functional project
□ Create an expandable codebase
□ Prepare for more advanced projects

1. For Beginners
□ Minimal code complexity
□ Through code comments
□ Step-by-step implementation
□ No advanced libraries required

2. For educators
□ Perfect teaching tool
□ Demonstrates theory in practice
□ Easy to grade and evaluate

7. Conclusion:
This online cart system implementation offers the perfect combination of educational values and real-world application with its emphasis on basic programming principles with a full Working system provided, it is:
□ A great project for students learning C
□ A tutorial for basic programming
□ A foundation for further advanced features.
The simplicity of the project allows rapid understanding,with structures that enable easy progression to more advanced programming challenges.