### ABSTRACT

The shopping cart system is a fundamental component of e-commerce platforms, facilitating the selection and purchase of items by users. This project report presents the design and implementation of an enhanced shopping cart system aimed at improving user experience and efficiency in online shopping. The proposed system incorporates features such as intuitive item selection, real-time inventory updates, secure payment processing, and personalized recommendations to enhance the overall shopping experience. Additionally, advanced functionalities such as order tracking, wish list management, and seamless integration with social media platforms are integrated to further streamline the user journey. The design phase involves the creation of a user-friendly interface with responsive design principles to ensure compatibility across various devices and screen sizes. Backend development focuses on robust database management, efficient data processing, and secure transaction handling to guarantee reliability and data integrity. The shopping cart system is a fundamental component of e-commerce platforms, facilitating the selection and purchase of items by users. This project report presents the design and implementation of an enhanced shopping cart system aimed at improving user experience and efficiency in online shopping.The proposed system incorporates features such as intuitive item selection, real-time inventory updates, secure payment processing, and personalized recommendations to enhance the overall shopping experience. Additionally, advanced functionalities such as order tracking, wishlist management, and seamless integration with social media platforms are integrated to further streamline the user journey.

# CHAPTER – 1

# INTRODUCTION

* 1. **DOMAIN INFORMATION**

Technical Domain:

Software Development: Shopping carts are software features built using various programming languages and frameworks depending on the chosen e-commerce platform. Understanding these languages and frameworks is crucial for developers who build and maintain shopping carts.

API Integrations: Shopping carts interact with other systems through APIs (Application Programming Interfaces). This might involve APIs for product catalogs to retrieve product information, payment gateways for processing transactions, and user databases for storing persistent shopping carts.

Database Management: For features like persistent shopping carts linked to user accounts, shopping carts rely on database storage to manage product selections and user data securely.

User Experience (UX) Domain:

Information Architecture (IA): The organization of product categories and navigation within the shopping cart is crucial for users to find what they're looking for easily. This falls under the Information Architecture (IA) domain.

Interaction Design (IxD): The way users interact with the cart's features (adding, removing items, updating quantities) is part of Interaction Design (IxD). It focuses on intuitive and user-friendly interactions.

Visual Design: The overall look and feel of the shopping cart, including layout, color scheme, and button design, contributes to the user experience. This falls under the Visual Design domain.

Additional Considerations:

Accessibility: Shopping carts should be accessible to users with disabilities, adhering to WCAG (Web Content Accessibility Guidelines) standards. This ensures everyone can navigate and use the cart's features effectively.

Mobile Responsiveness: In today's mobile-first world, shopping carts must be responsive and function seamlessly on various devices (desktops, tablets, smartphones).

* 1. **REASONING AND SIGNIFICANCE OF PROJECT**
* **Reasoning for Developing a Web-based Shopping cart:**
* Accessibility: A web-based shopping cart allows users to access the platform from anywhere with an internet connection, making it convenient for shoppers to browse and purchase products at their own convenience.
* Scalability: Web-based systems can easily scale up to accommodate increasing numbers of users and products without the need for significant infrastructure changes. This scalability ensures that the shopping cart can grow alongside the business.
* Cost-effectiveness: Developing a web-based shopping cart can be more cost-effective than building a native application for each platform (e.g., iOS, Android). It allows for a single codebase that can be deployed across multiple devices and operating systems.
* Cross-platform compatibility: With a web-based shopping cart, users can access the platform from various devices, including desktop computers, laptops, tablets, and smartphones, regardless of the operating system they use.
* Real-time updates: Web-based systems can easily push updates and new features to users without requiring them to download and install anything manually. This ensures that shoppers always have access to the latest functionalities and improvements.
* **Significance of a Web-developed Shopping cart:**
* Enhanced User Experience: A well-developed shopping cart improves the overall user experience by providing a seamless and intuitive interface for customers to add, remove, and manage items they wish to purchase.
* Functionality and Features: Discussing the functionality and features implemented in the shopping cart, such as product catalog display, sorting and filtering options, multiple payment methods, order tracking, and security measures, demonstrates the technical capabilities of the web development team.
* Business Efficiency: A robust shopping cart system streamlines the purchasing process, leading to increased efficiency for the business. It allows for better inventory management, order processing, and customer communication, which ultimately contributes to higher sales and customer satisfaction.
* Security Measures: Security is paramount in e-commerce. Detailing the security measures implemented within the shopping cart, such as SSL encryption, PCI compliance, and secure payment gateways, showcases the project's commitment to protecting sensitive customer data.
* Scalability and Performance: Addressing how the shopping cart system handles scalability and performance under varying loads is crucial. Discussing techniques such as caching, load balancing, and database optimization demonstrates the project's foresight in handling future growth and ensuring consistent performance.
* Integration with Backend Systems: The shopping cart typically integrates with various backend systems such as inventory management, order fulfillment, and customer relationship management (CRM) systems. Explaining how these integrations were implemented and their impact on business operations provides valuable insights into the project's scope and complexity.
  1. **OBJECTIVES**
* User-Friendly Experience: Ensure the shopping cart interface is intuitive and easy to navigate, reducing friction in the purchasing process.
* Convenience: Enable customers to add, remove, and modify items in the cart with ease. Implement features like saved carts for returning customers.
* Cross-Selling and Upselling: Suggest related or complementary products to encourage customers to add more items to their cart, thereby increasing the average order value.
* Clear Product Presentation: Display product images, descriptions, prices, and availability prominently in the cart to provide customers with all necessary information before making a purchase decision.
* Transparency: Clearly outline all costs associated with the purchase, including taxes, shipping fees, and any additional charges, to avoid surprises at checkout.
* Security and Trust: Implement robust security measures to protect customers' personal and financial information, thereby fostering trust in the shopping cart system.
* Seamless Checkout Process: Streamline the checkout process to minimize the number of steps required for customers to complete their purchase, reducing cart abandonment rates.
* Mobile Compatibility: Ensure the shopping cart is optimized for mobile devices, allowing customers to easily shop and checkout from their smartphones or tablets.

* 1. **LITERATURE SURVE**

1. 1. describes the implementation of smart shopping cart using radio frequency identification using the RFID sensors, Arduino microcontroller, Bluetooth module, and Mobile application. Where the mobile is connected to the shopping cart and the application is already installed, the data is shared using the Bluetooth from the arduino microcontroller and the mobile then with the server.
2. 2.” Intelligent shopping cart using BOLT based on IOT”. IOT kit consists of barcode scanner, LCD display, Bolt ESP8266. The broad clarification of its process is, when consumer takes an item and put inside the trolley, that time barcode scanner scans the item barcode and value as well as gain to show into the digital display panel. Later than consumer concluded their purchasing and the bill is sent to the counter section.
3. 3.“Smart Trolley with Instant Billing to Ease Queues at Shopping Malls using ARM7 LPC2148. This is based on arm7 microcontroller fitted with an LCD and RFID scanner and a wireless technology called zigbee. The LCD used is a 16x2 and zigbee modules make the wireless network to work even at long distance due to its wide range, the RFID scanner scans the product’s unique code and its price. And it gets displayed on the LCD screen. So, after costumer has finished with the shopping, he/she has to visit the counter and pay the bill as displayed on the LCD screen fitted on the trolley.

4.EM-18 RFID scanner module has been used. It uses a RFID reader which will read 125 kHz tags. So, it will be known as a low frequency RFID reader. The RFID Readers here used are big tags with range of 125KHZ which can be detected by EM-18 Module. It shows the real time billing and you can even delete the item you don’t want by pressing the delete button. In this author has used ARDUINO Uno which one of the cheapest and most efficient models in the market. It contains everything required to support the microcontroller merely connect it to a laptop (or applicable wall power adapter) with a USB cable or power it with an AC-to-DC adapter or battery to get started. Once the item is scanned it will start billing and you can remove the item if you want.

5.Framework is utilized to ease lines in shopping centre by utilizing RFID module. The RFID reader will peruse the RFID Tag set on the item when the item falls in the trolley. In the event that, the client needs to expel any item then he should expel that item from the trolley. The LCD will show the subtitles of the expelled item like name, cost and the absolute bill and with the help of Xampp server the bill will be send to the cashier

6.Describes the implementation of a Smart Shopping Cart using ZigBee networks. The reliable and cost-efficient system design also ensures detection of deception. Thus, the smart system attracts both the buyers and sellers and ZigBee acts like Xampp server but is more reliable

# CHAPTER – 2

# PROBLEM DEFINITION AND SCOPE

* 1. **PROBLEM DEFINITION**

Shopping is simple but waiting on a bill counter makes shopping too boring and a tedious task. Huge amount of rush plus cashier preparing the bill is too time consuming and results in long que. In this prevailing pandemic standing in queues for billing in malls or shopping market is not advisable as virus may spread.

• The present billing system is time consuming process which irritates people by disturbing their busy schedules

**Benefits of implementing a hospital management system:**

Convenience for Customers:

* Enables customers to browse products, add them to their cart, and complete purchases in a streamlined manner.
* Provides a centralized platform for managing shopping lists, wishlists, and previously saved items

Increased Sales and Revenue:

* Facilitates upselling and cross-selling by suggesting related products or accessories during the checkout process.
* Reduces cart abandonment rates by simplifying the checkout process and offering guest checkout options.

Efficiency for Business Operations:

* Automates inventory management, updating product availability in real-time and reducing the likelihood of overselling.
* Streamlines order processing and fulfillment, improving efficiency and reducing errors.

Enhanced Customer Experience:

* Offers personalized recommendations based on previous purchases or browsing history, enhancing the overall shopping experience.
* Enables feedback and reviews from customers, fostering trust and credibility.

Scalability and Adaptability:

* Accommodates growth and expansion by allowing for the addition of new products, categories, and features.
* Supports customization and integration with other systems or third-party services to meet specific business requirements.
* Adapts to evolving consumer preferences and market trends through continuous updates and improvements.
  1. **SCOPE**

1. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system facility to system which can make customers happy.
2. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won’t be losing any more customers to the trending online shops such as flip cart or e bay. Since the application is available in the Smartphone it is easily accessible and
3. always available.

**2.3 SOFTWARE AND HARDWARE REQUIREMENTS**

**2.3.1 Hardware Requirements**

**Server:**

* Processor: Minimum Intel Xeon dual-core or equivalent (consider future scalability)
* RAM: Minimum 16 GB RAM for smooth operation (more if handling a large user base)
* Storage:
* Minimum 250 GB SSD for the operating system and application files.
* Additional storage (HDD/SSD) as needed for patient data based on volume and regulations.
* Network: Gigabit Ethernet connection for efficient data transfer
* Operating System:
* Linux distribution like Ubuntu/CentOS is a popular choice for web development due to its stability and open-source nature.
* Windows Server 2022 can also be used if preferred.

**Workstations:**

* Processor: Minimum Intel Core i3/i5 or equivalent for handling hospital management tasks
* RAM: Minimum 8 GB RAM (16 GB recommended for multitasking)
* Storage: 250 GB SSD/NV Me for faster application loading and data access
* Network: Fast Ethernet connection for stable network access
* Operating System: Windows 10 or equivalent for user familiarity (other options like Linux are also possible)

**2.3.2 Software Requirements**

* Web Server: Apache or Nginx are popular open-source web servers for running the HTML, CSS, and JavaScript files of your application.\

**Front-end Technologies:**

* HTML: for structuring the web pages of your application.
* CSS: for styling the web pages and defining the user interface.
* JavaScript: for adding interactivity and dynamic behavior to the web pages. You might also consider a JavaScript framework like React or Angular for complex user interfaces.

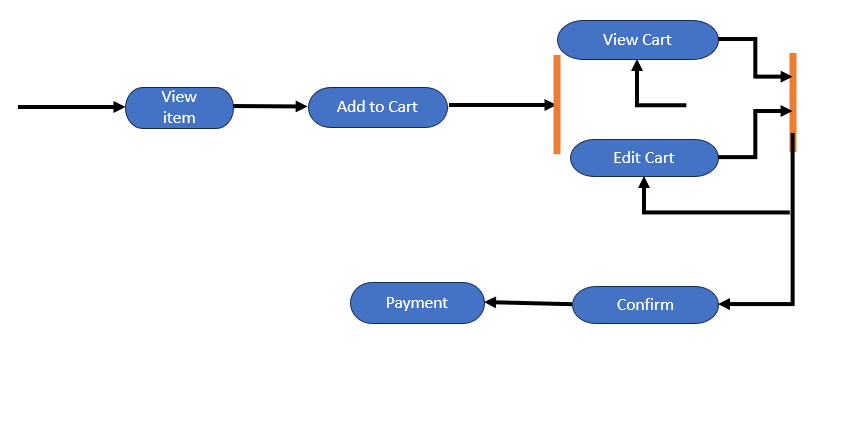
**Back-end Development Language:**

* JavaScript (Node.js) can be used for server-side

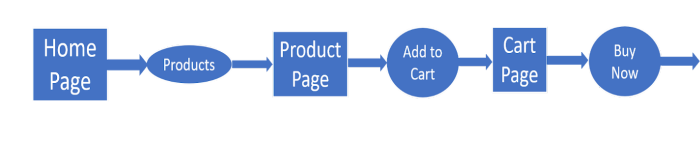
# CHAPTER – 3

# DESIGN

* 1. **SYSTEM ARCHITECTURE**

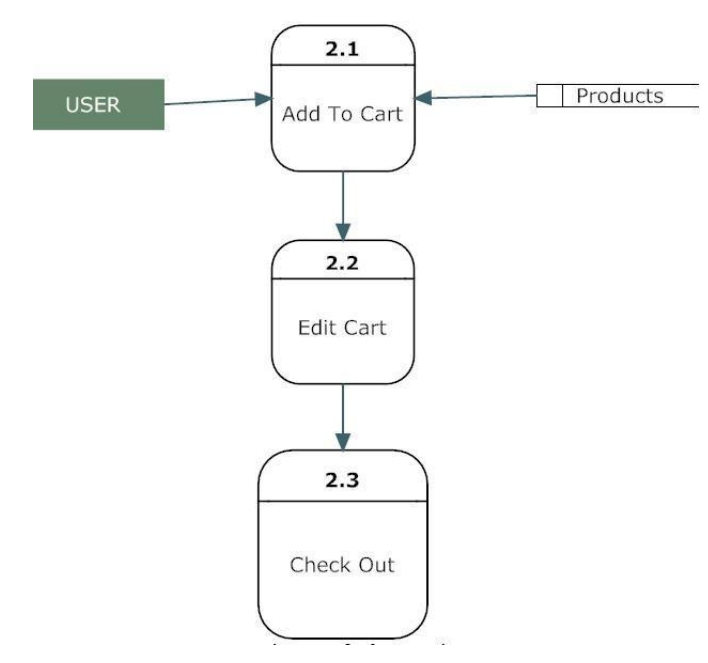


Data Flow Diagram

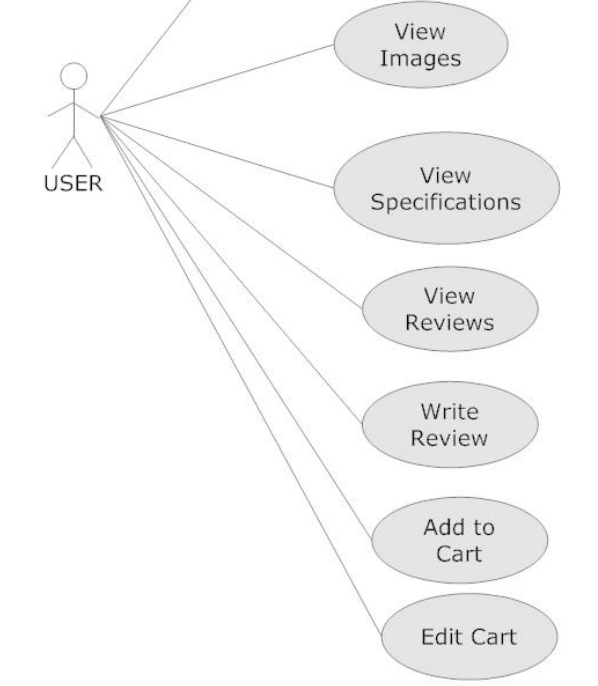


**LEVEL 1**

**LEVEL 2**

****

Use Case



# CHAPTER – 4

# IMPLEMENTATION AND TESTING

* 1. **IMPLEMENTATION DETAILS**
     1. **Model 1**

Module 1: Core Functionality

Data Structures:

Product: A data structure to store information about each product (e.g., ID, name, price, description, image URL, stock availability).

Cart Item: A data structure to represent an item in the cart, including a reference to the product, quantity selected, and potentially additional details like chosen size or color.

Shopping Cart: A data structure to manage the collection of cart items, including methods to add, remove, update quantities, and calculate totals.

Functions:

Add to Cart: Takes a product and quantity as input, validates availability, and adds a new "Cart Item" to the "Shopping Cart" data structure.

Remove from Cart: Takes a product ID or a reference to the "Cart Item" as input and removes the item from the shopping cart.

Update Quantity: Takes a product ID or a reference to the "Cart Item" and a new quantity as input, validates the updated quantity against stock, and updates the quantity in the shopping cart.

Get Cart Items: Returns a list of all "Cart Item" objects currently in the shopping cart.

Get Cart Total: Calculates the total cost of all items in the shopping cart considering quantities and product prices.

* + 1. **Model 2**

Module 2: Persistence and Integrations

Data Persistence:

Session Storage: Stores cart items temporarily in the user's browser session for short-term persistence. Useful for anonymous users or quick purchases.

Database Storage: Stores cart items in a database linked to the user account for persistent shopping cart functionality. Allows users to revisit their cart later and recover it across devices.

Integrations:

Product Catalog: The shopping cart needs to interact with the product catalog to retrieve product information (e.g., name, price, stock) when adding items or updating quantities. This could involve API calls or direct database access.

Checkout Process: The shopping cart needs to integrate with the checkout process to transfer cart information (items, quantities, totals) when a user proceeds to purchase. This might involve data transfer to the payment gateway or order processing system.

Additional Considerations:

Promotions and Discounts: The shopping cart might need to handle promotions and discounts applied to the entire cart or individual items. This can involve additional logic for calculating discounted prices and displaying the breakdown.

User Authentication: For persistent shopping carts, user authentication is required to link the cart to a specific user account. This could involve session management or cookies to identify the user.

* 1. **TERSTING STRATEGY**

Module 1: Core Functionality

Positive Tests:

Add single and multiple items to the cart with varying quantities.

Remove items from the cart.

Update item quantities accurately.

Calculate cart totals correctly, considering product prices and quantities.

Apply valid coupons and discounts and reflect them in the total price.

Negative Tests:

Attempt to add an item with zero or negative quantity.

Add an item exceeding available stock.

Use invalid product IDs or data for adding items.

Apply invalid or expired coupons.

Test behavior during unexpected user actions (e.g., refreshing page after adding items).

Module 2: Persistence and Integrations

Persistence Tests:

Verify cart items are stored correctly in session storage (if used).

Test persistence across browser sessions and devices (if using database storage with user accounts).

Simulate scenarios like browser crashes or session timeouts and ensure cart recovery functionality (if applicable).

Integration Tests:

Verify seamless interaction with the product catalog for retrieving product information.

Test data transfer to the checkout process and ensure accurate cart information is passed along.

If applicable, test integration with any external payment gateways for handling transactions.

2. User Experience (UX) Testing:

Usability Testing:

Observe user interaction with the shopping cart features (adding, removing, updating items).

Ensure the interface is clear, intuitive, and easy to navigate.

Test on various devices (desktop, mobile, tablets) to confirm responsiveness and a consistent experience.

Error Handling:

Verify informative error messages are displayed for invalid actions (e.g., adding out-of-stock items).

Ensure proper feedback mechanisms for successful cart updates and actions.

3. Performance Testing:

Load Testing:

Simulate high traffic scenarios and test cart performance under increased load.

Ensure the shopping cart functions smoothly even with multiple users adding or removing items concurrently.

4. Security Testing:

Data Security:

Verify secure storage of sensitive user information, especially if using persistent shopping carts with user accounts.

Test for vulnerabilities in data transmission during interactions with the product catalog or checkout process.

Additional Considerations:

Accessibility Testing:

Ensure the shopping cart is accessible to users with disabilities, following WCAG guidelines.

Internationalization Testing:

If the shopping cart caters to international users, test functionality with different currencies and languages.

# CHAPTER – 5

# DEPLOYMENT

* 1. **RESULT AND APPLICATIONS**

**Results:**

* + Functionality Overview:

Provide an overview of the key functionalities implemented in the shopping cart system, such as adding items to the cart, updating quantities, removing items, viewing the cart, and proceeding to checkout.

* + User Interface:

Describe the user interface design and how it enhances the user experience, including ease of navigation, responsiveness, and intuitive interaction.

* + Backend System: Discuss the backend architecture and how it handles data management, including storage of product information, user details, and cart data.
  + API Integration: Explain how external APIs (e.g., payment gateways) were integrated into the system and their role in facilitating transactions.
  + Security Measures: Detail the security measures implemented to protect user data, prevent unauthorized access, and ensure secure transactions.
  + Performance Metrics: Provide performance metrics such as response times, server load, and scalability to demonstrate the efficiency of the system under various loads.



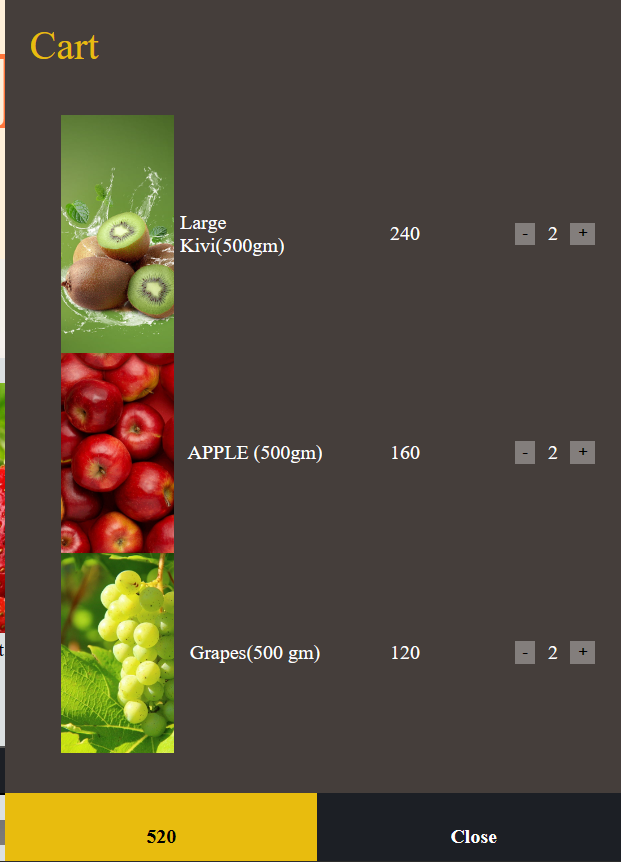
Result



Add To Cart



Cart



Edit Cart

**Applications:**

* Customization and Integration:

Discuss how the system can be customized and integrated into existing websites or platforms to enable online shopping functionality.

* Business Benefits:
* Outline the potential benefits for businesses, such as increased sales, improved customer satisfaction, and efficient order management.
* User Experience Enhancement:

Emphasize how the shopping cart system enhances the user experience by providing a seamless and convenient shopping process.

* Scalability and Future Growth:
* Discuss the scalability of the system and its potential for future growth, including support for a growing number of users and products.
  1. **CONCLUSION**

The basic idea is to make shopping more easy and comfortable in the overpopulated cities. This shopping cart is built with a system that enables customer to bill their products at cart itself without standing in long ques for billing. Finally, a system named shopping cart This helps the user to view the bill on his/her device. the medical field will greatly benefit from this work.

* 1. **FUTURE ENHANCEMENT**
     1. **Advanced Functionality**
* User Accounts and Profiles:

Allow users to create accounts/profiles to store their shopping cart contents for future sessions.

Provide options for users to manage their addresses, payment methods, and order history.

* Wishlist and Favorites:

Implement a wishlist feature where users can save products for later purchase.

Allow users to create and manage multiple wishlists for different occasions or preferences.

* Product Recommendations:

Utilize algorithms to suggest related or complementary products based on users' browsing and purchase history.

Display personalized recommendations to enhance cross-selling and upselling opportunities.

* Advanced Search and Filtering:

Implement advanced search capabilities with filters for categories, brands, price ranges, sizes, colors, etc.

Provide sorting options based on relevance, price, popularity, and other criteria.

* Inventory Management:

Ensure real-time inventory updates to prevent overselling or out-of-stock situations.

Allow administrators to set low stock alerts and manage product availability effectively.

**5.3.2 Improved User Experience**

* Seamless Navigation:

Implement a persistent mini-cart or floating cart icon that provides users with quick access to their cart without interrupting their browsing experience.

* Visual Feedback and Confirmation:

Provide visual feedback (e.g., animations, tooltips) when users add items to their cart to confirm that the action was successful.

* Clear Product Information:

Display clear and concise product information, including images, descriptions, prices, and available variants (e.g., sizes, colors).

* Dynamic Pricing and Discounts:

Show dynamic pricing information, including discounts, promotions, and total order cost, as users add or remove items from their cart.

* Mobile Optimization:

Optimize the shopping cart interface for mobile devices with responsive design, larger touch targets, and simplified checkout processes.

**5.3.3 Security and Integration**

* Enhanced Data Security: Implement advanced security measures like multi-factor authentication and blockchain technology to safeguard sensitive patient data.
* Integration with Wearable Devices: Allow patients to connect wearable devices like fitness trackers to the system for tracking vital signs and monitoring health conditions remotely.
* Interoperability with Third-party Systems: Ensure your system can integrate with existing hospital software like Electronic Health Records (EHR) systems for seamless data exchange.

**REFERENCES:**

B. Kumar Yadav, A. Burman, A. Mahato, M. Choudhary and A. Kundu, "Smart Cart: A Distributed Framework," 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE), 2020.

S. Kowshika, S. S. Madhu mitha, G. Madhu Varshini, V. Megha and K. Lakshmi, "IoT based Smart Shopping Trolley with Mobile Cart Application," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021.

.T. K. Das, A. K. Tripathy and K. Srinivasan, "A Smart Trolley for Smart Shopping," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), 2020.

Sakorn Mekruksavanich, “The Smart Shopping Basket Based on IoT Applications”, Software Engineering and Service Science (ICSESS) 2019 10th International Conference, 2019.

A. D, S. S. Guddad, A. S, K. Yanamala and S. S, "Smart Shopping Cart using IOT and robotic arm," 2021 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), 2021.