End-Term Project Report

Mobile Application Development



Faculty name: Ms. Aarti Sangwan Student name: Tavishi Sehrawat (21csu501)

Aayush Kumar Upadhyay (21csu531)

Gopal Krishan Sharma (21csu512)

Neeraj Sangwan (21csu492)

Vikas Kumar Mehta (21csu459)

Semester: 4th

Group: FS-B

**Department of Computer Science and Engineering**

**The NorthCap University, Gurugram- 122001, India**

**Session 2023-24**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.No** |  | **Page No.** |
| **1.** | **Project Description** | **2** |
| **2.** | **Problem Statement** | **2** |
| **3.** | **Analysis**  **3.1 Hardware Requirements**  **3.2 Software Requirements** | **2** |
| **4.** | **Design**  **4.1 Data/Input Output Description:**  **4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps** | **3** |
| **5.** | **Implementation and Testing (stage/module wise)** | **4** |
| **6.** | **Output (Screenshots)** | **5** |
| **7.** | **Conclusion and Future Scope** | **8** |

1. **Project Description**

The objective of this project is to create a cross-platform shopping application called ShopRite using the Flutter framework. The application enables users to browse products, add them to carts, utilize product filtering and search functionalities. Additionally, the app integrates with inventory management APIs for real-time product availability and uses state management for real-time cart updates.

1. **Problem Statement**

In the competitive world of e-commerce, providing a seamless shopping experience across multiple platforms is essential. Many shopping apps fail to offer real-time inventory updates and efficient cart management, leading to user dissatisfaction. This project aims to solve these issues by developing a robust, user-friendly shopping app with advanced functionalities and real-time updates.

**3. Analysis**

**3.1 Hardware Requirements**

* Processor: Intel Core i5 or higher
* RAM: 8 GB or higher
* Storage: 50 GB of free space
* Smartphone: Android 6.0 or higher / iOS 11.0 or higher
* Internet Connection: Stable broadband connection

**3.2 Software Requirements**

* Operating System: Windows 10 or higher / macOS / Linux
* Flutter SDK
* Dart Programming Language
* Integrated Development Environment (IDE): Android Studio / Visual Studio Code
* Firebase (for backend services)
* API Services: RESTful APIs for inventory management
* Version Control: Git

**4. Design**

**4.1 Data/Input Output Description:**

* **Product Information & Attributes:** Detailed product descriptions, images, specifications, variations (color, size, etc.), and pricing.
* **Category & Brand Management:** Products organized into categories and subcategories, with brand information and filtering by brand.

**4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps**

*Algorithmic Approach*

* **Search Functionality:** Utilize keyword matching and filters (category, price, brand) to refine search results.
* **State Management:** Implement state management (Provider/Bloc) for real-time cart and inventory updates.
* **Integration with APIs:** Connect to inventory management APIs to fetch and update real-time product availability.

*Data Flow Diagram (DFD)*

* **User Module:**
* Input: Search query, filters (price, brand, category)
* Process: Search and filter products
* Output: Display filtered product list
* **Product Module:**
* Input: Product details from APIs
* Process: Update product information in the app
* Output: Real-time product availability display
* **Cart Module:**
* Input: User adds/removes products from the cart
* Process: Update cart contents and sync with server
* Output: Display updated cart with real-time availability

*Entity-Relationship (ER) Diagram*

* **Entities:** User, Product, Category, Brand, Review, Cart
* **Relationships:**
* Users can review Products
* Products belong to Categories and Brands
* Users manage Carts containing Products

**5. Implementation and Testing (stage/module wise)**

***Product & Catalog Management***

* **Stage 1:** Develop Product Information System
* Implementation: Create data models and APIs for product details
* Testing: Validate data retrieval and display accuracy
* **Stage 2:** Implement Category & Brand Management
* Implementation: Develop interfaces for categories and brands
* Testing: Ensure correct categorization and brand filtering

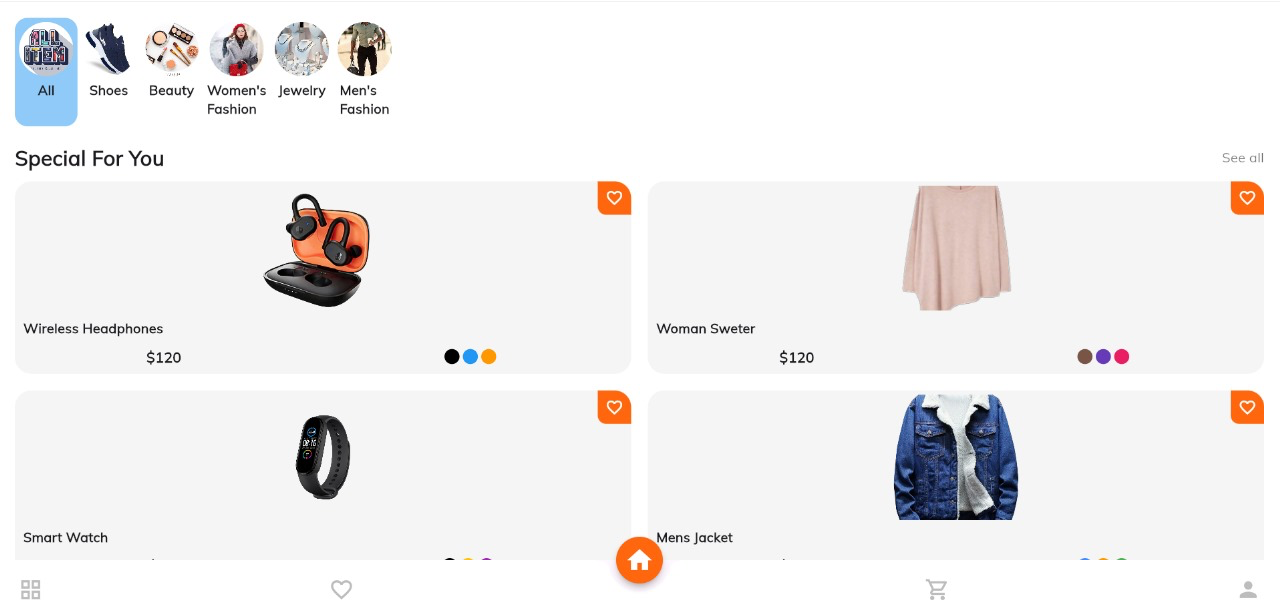
***Search & Cart Management***

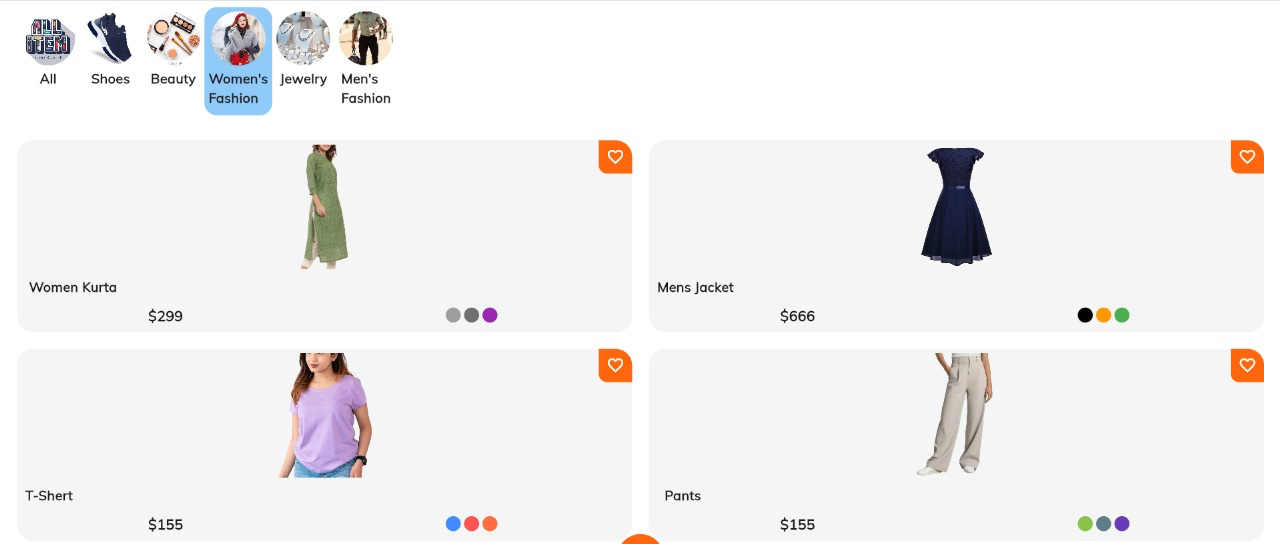
* **Stage 3:** Develop Search Functionalities
* Implementation: Integrate search algorithms and filtering options
* Testing: Validate search results and performance
* **Stage 4:** Implement State Management for Cart
* Implementation: Use Provider/Bloc for state management
* Testing: Ensure real-time cart updates and synchronization

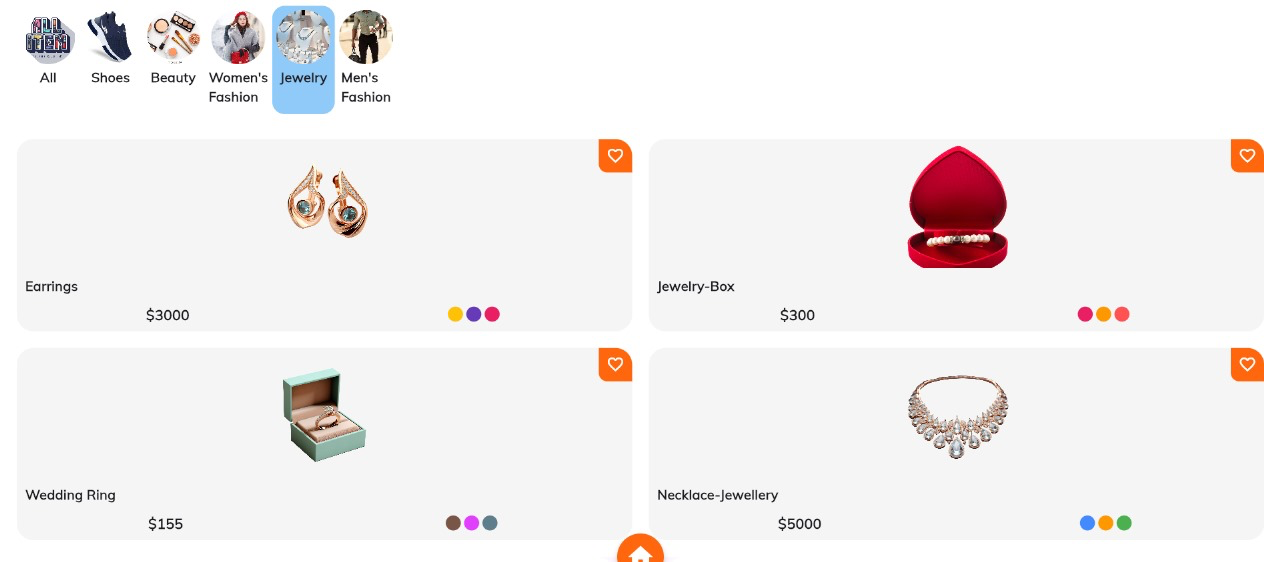
***API Integration***

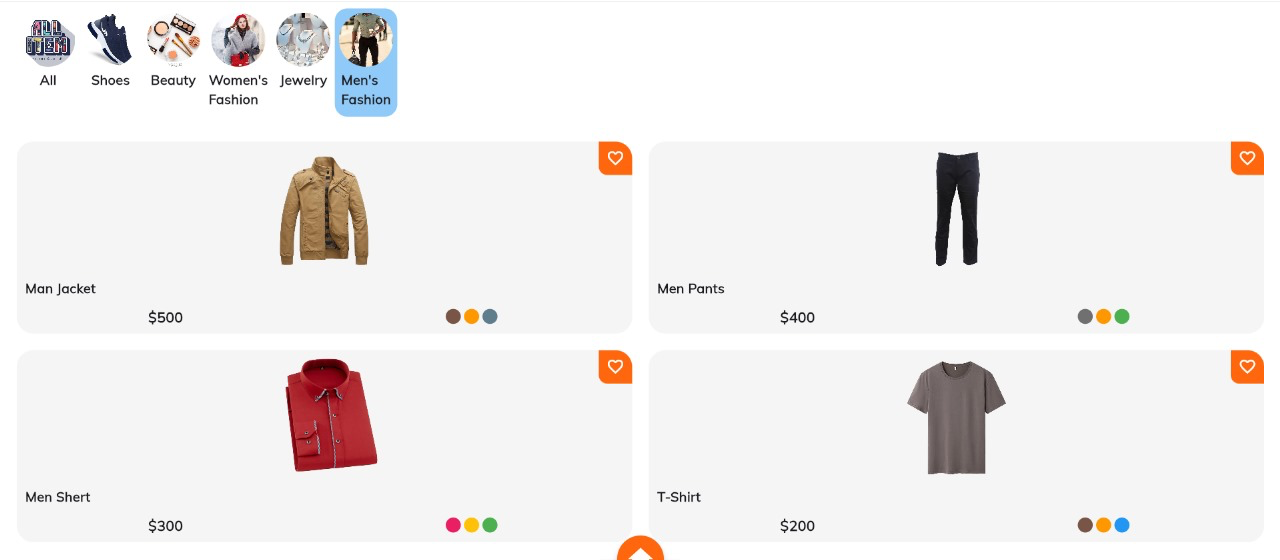
* **Stage 5:** Integrate with Inventory Management APIs
* Implementation: Connect to external APIs for real-time inventory updates
* Testing: Validate real-time product availability updates

**6. Output (Screenshots)**

****

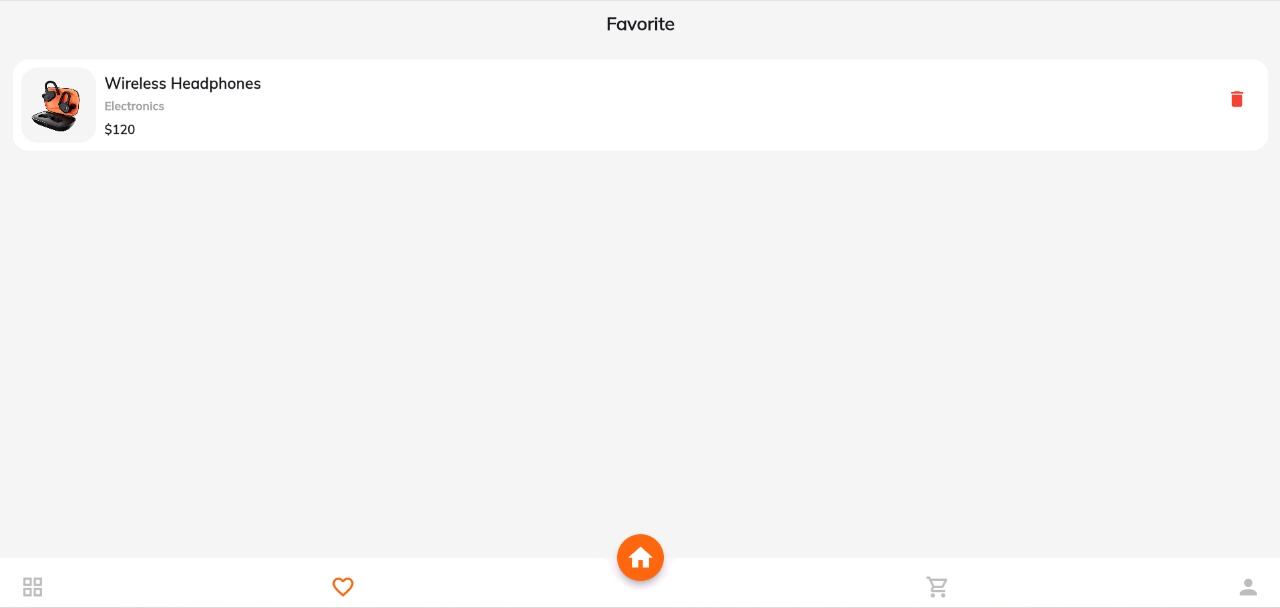
****

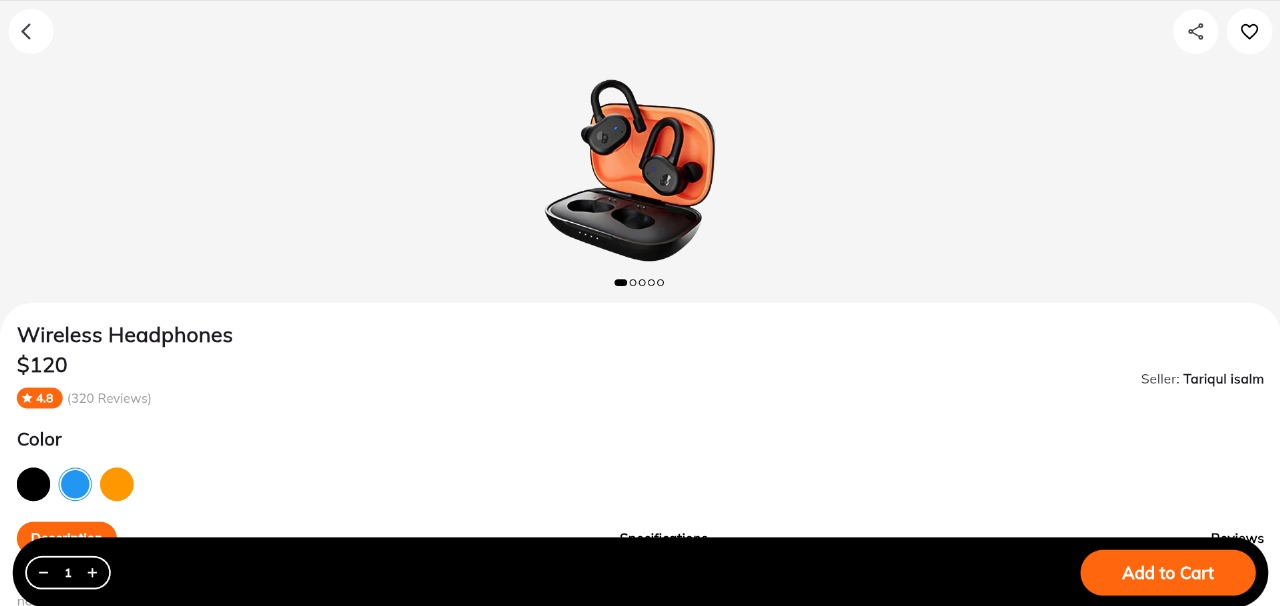
****

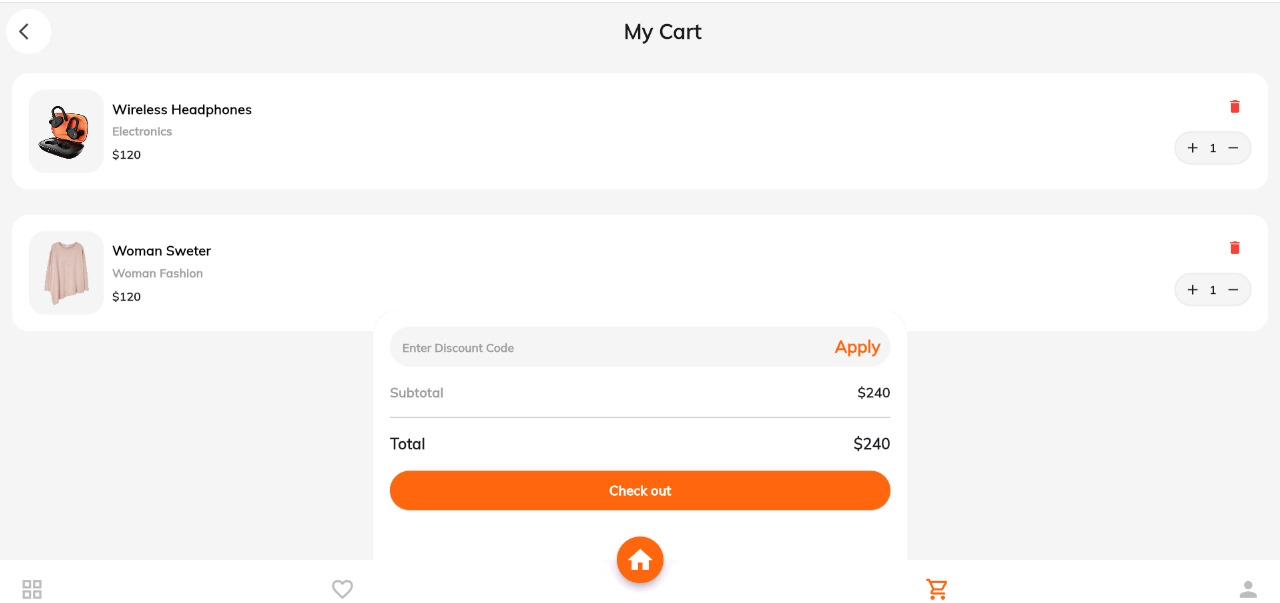
****

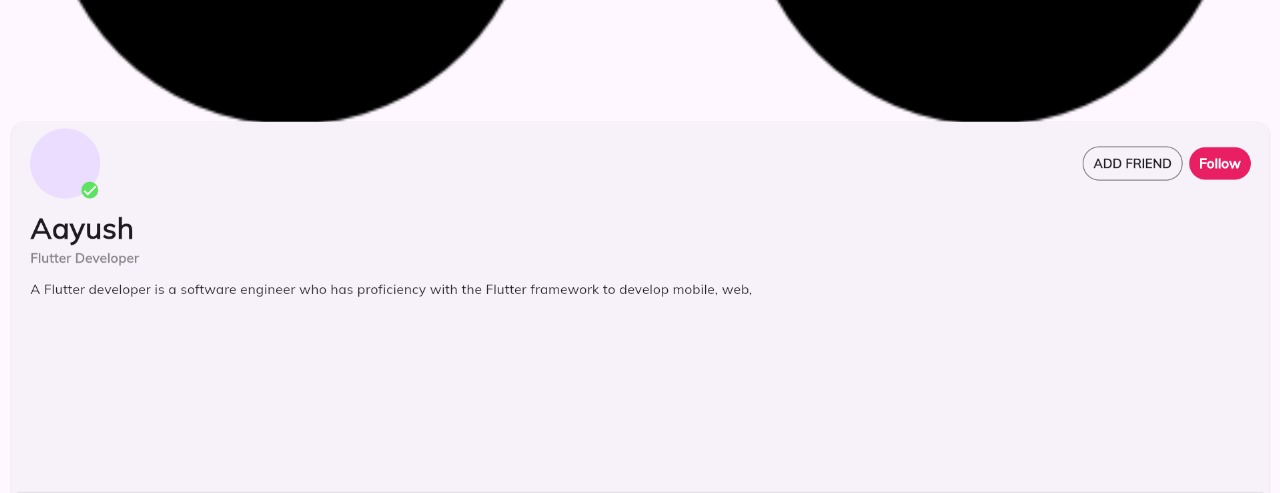
**A white background with a black border

Description automatically generated with medium confidence**

****

****

****

****

**7. Conclusion and Future Scope**

This project successfully demonstrates a cross-platform E-Commerce application with essential features such as product browsing, search, filtering, and real-time cart management. The integration with inventory management APIs and state management enhances user experience by providing real-time updates. Future enhancements could include advanced personalization, integration with payment gateways, and expanded API functionalities for more comprehensive inventory management.