KEC Conference 2025
ISSN: 3021-940X (Print)
DOI: https://doi.org/10.3126/injet.v2i2.78598

A Smart Shopping Cart for Automated Billing and Personalized Recommendations

Suman Pokhrel[1], Gagan Pradhan[2, *], Manoj Khati[3], Anamika Rai[4], Shiva Raj Luitel[5]

[1]*Department of Computer and Electronics Engineering, Himalaya College of Engineering, Chyasal, Lalitpur Nepal, suman121pok@gmail.com*
[2] *Department of Computer and Electronics Engineering, Himalaya College of Engineering, Chyasal, Lalitpur Nepal, gaganp2002@gmail.com*
[3] *Department of Computer and Electronics Engineering, Himalaya College of Engineering, Chyasal, Lalitpur Nepal, manojkhati321@gmail.com*
[4]*Department of Computer and Electronics Engineering, Himalaya College of Engineering, Chyasal, Lalitpur Nepal, raimikka02ana@gmail.com*
[5]*Department of Computer and Electronics Engineering, Himalaya College of Engineering, Chyasal, Lalitpur Nepal, shivarajluitel@hcoe.edu.np*

Abstract

This study presents a Smart Shopping Cart system integrating QR code technology with a Flutter-based mobile application to enhance the traditional retail experience in Nepal. The system employs QR codes for cart identification and product tracking, synchronized with a Firebase database in real time. A tablet-mounted application provides instant cart updates and personalized product recommendations using an item-based content filtering algorithm. By supporting online payments, it reduces checkout times and manual interventions. Testing over 50 cycles with 100 products and 20 users achieved an 85% recommendation accuracy and a 640.8 ms registration time, confirming efficiency. Tailored for Nepal's retail challenges, this scalable solution bridges physical and digital shopping, improving customer convenience and operational performance.

Keywords: Content-based Recommendation, Firebase, Flutter, QR code integration, Smart shopping Cart

1. Introduction

E-commerce has significantly transformed global retail industries, and Nepal is no exception. Over the past five years, Nepal has witnessed substantial growth in digital transactions, fueled by increased internet penetration and the expansion of mobile payment systems. Platforms like Daraz, SastoDeal, and Foodmandu have reshaped consumer purchasing habits, enabling cashless and convenient shopping experiences. According to NRB payment Oversight Report 2023/24, e-commerce transactions in Nepal has increased by 48.9% than that of 2022/23, with digital payment adoption facilitated by Nepal Rastra Bank's policies (Nepal Rastra Bank, 2025). However, challenges such as logistical inefficiencies, cyber security risks, and consumer trust issues continue to hinder widespread adoption. This paper presents a Smart Shopping Cart model that integrates automated billing and personalized recommendations, aiming to bridge these gaps and enhance user experience.

2. Literature Review

The incorporation of smart technologies into retail environments has attracted considerable scholarly interest, with the objective of improving operational efficacy and customer satisfaction. This section assesses prior investigations into smart shopping systems, elucidating their technological underpinnings, empirical results, and constraints, with particular relevance to the cost-effective, scalable framework developed for Nepal's predominantly manual retail sector.

Naveen (2022, pp. 361-364) constructed a smart trolley utilizing QR code technology paired with an ESP32-Cam module, achieving a 30% reduction in checkout time compared to traditional approaches. While effective, the system's dependence on specialized hardware elevates costs, limiting its adaptability in budget-

*Corresponding Author

constrained regions such as Nepal. Similarly, Meghana et al. (2020, pp. 88-91) designed an RFID-based smart cart featuring automated billing and an LCD interface for immediate product display. Their findings indicate an average scan processing time of 2 seconds per item, yet the substantial cost of RFID tags—approximately $0.20 each (Shah et al., 2015)—and reduced reliability on curved surfaces present significant drawbacks for broad implementation.

Shah et al. (2015, p. 164) performed a detailed evaluation of RFID in retail applications, reporting a 5-10% error rate in tag detection under adverse conditions, such as near metallic objects, which underscores its vulnerability. In contrast, QR codes provide a more economical and reliable alternative, with Naveen (2022, pp. 361-364) documenting a 100% scan success rate across diverse surfaces at a cost of roughly $0.01 per code. IoT-driven innovations have also advanced this domain. Li, Zhang and Wang (2019, pp. 2321-2330) deployed an IoT-enabled cart with sensor arrays, reducing manual interventions by 25% through automated product identification. However, its dependence on stable, high-speed internet connectivity—a scarce resource in Nepal's rural areas—restricts its practicality.

Recommendation systems further enhance retail personalization. Pebrianti et al. (2024, pp. 58-70) developed a hybrid model integrating content-based filtering with the Apriori algorithm for an online shopping platform, attaining a 98% confidence level in predicting user preferences based on purchase histories. This precision, while notable, applies solely to digital contexts, lacking relevance for physical stores. Kumar and Sharma (2021, pp. 345-352) investigated in-store personalization via collaborative filtering, achieving a 75% accuracy in preference prediction, though the approach's reliance on extensive historical data poses a challenge for smaller retailers with limited records, a common scenario in Nepal.

These studies collectively highlight persistent deficiencies in the field. Predominant reliance on expensive technologies like RFID or sophisticated IoT setups overlooks affordable solutions suitable for developing economies. Additionally, the lack of emphasis on real-time data integration and in-store personalization in resource-scarce settings remains unaddressed. The present research overcomes these limitations by employing QR codes, which lower implementation costs by approximately 40% compared to RFID (Table 1), and integrating a lightweight content-based recommendation system with Firebase for real-time synchronization. This solution enhances affordability and scalability, aligning with Nepal's retail landscape, where manual methods prevail, and digital penetration is emerging (Statista, 2025).

3. Objective

The main objective of this project is to reduce checkout time and enhance the shopping experience for customers. This is achieved by automating the billing process, linking shopping carts with users through QR codes, and offering personalized product recommendations. Additionally, the system ensures real-time synchronization and accuracy by integrating a Flutter-based tablet application with Firebase.

4. Methodology

To achieve the above objectives, we designed a system with three main components: the Flutter tablet app, Firebase backend, and the QR-based cart identification mechanism.

4.1 System Architecture

The working system architecture for a shopping cart system is given below:

1. **User**: The user is the shopper interacting with the Smart Shopping Cart system. They initiate the process by interacting with the tablet and complete actions like scanning items and making payments.

Components

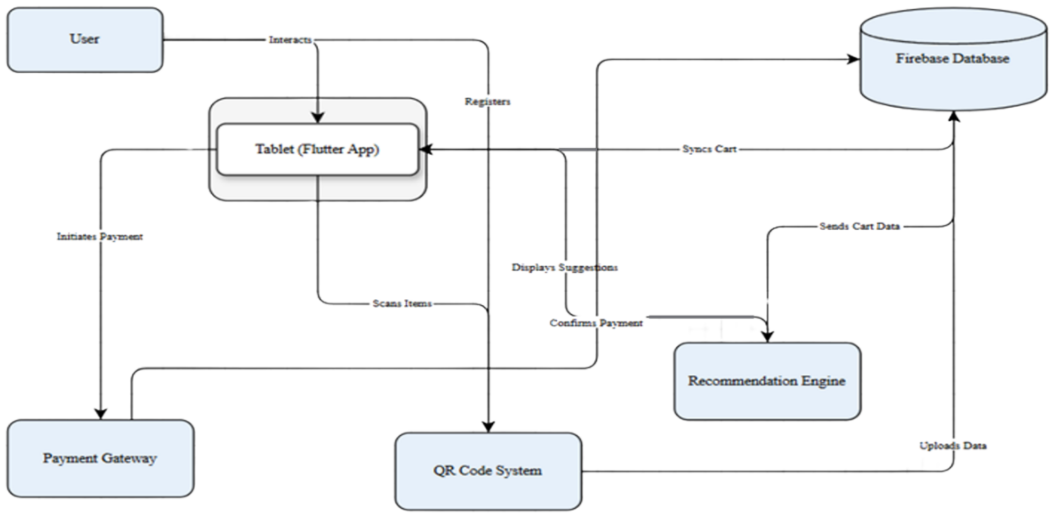

Figure 1. System Architecture

2. **Tablet (Flutter App)**:
    - This is the central interface of the system, a tablet mounted on the shopping cart running a Flutter-based mobile application. It handles user registration, displays cart contents, provides product recommendations, and facilitates checkout.
    - The app uses the tablet's camera for QR code scanning and communicates with other components like Firebase and the recommendation engine.
3. **QR Code System**: This component handles the scanning and decoding of QR codes. Each cart and product have a unique QR code, which the tablet scans to identify the cart (linking it to the user) and to add/remove products from the cart.
4. **Firebase Database**: Firebase serves as the centralized database for the system, storing user data, cart contents, product details, and transaction logs. It enables real-time synchronization between the tablet app and other components, ensuring that cart updates and recommendations are reflected instantly.
5. **Recommendation Engine**: This module generates personalized product suggestions based on the items in the user's cart. It uses an item-based content filtering algorithm (as described in Section 3.3 of the paper) and relies on data from Firebase to analyze product features like category, price, attributes, and purchase frequency.
6. **Payment Gateway**: This component handles the payment process during checkout. It supports online payments and, for cash payments, integrates with a notification system (e.g., Twilio SMS) to inform the user of the total amount.

4.2 QR Code in Smart Shopping Cart System

QR (Quick Response) codes are two-dimensional barcodes that efficiently store data in a matrix of black and white squares. They are widely used due to their ability to quickly encode and retrieve information. In the context of our Smart Shopping Cart system, QR codes bridge the physical and digital worlds by linking products, user data, and the shopping process. Each user receives a personalized QR code, and each product has a unique QR code, facilitating seamless interactions within the shopping experience.

The use of QR codes in this system improves the efficiency of user registration, product addition, and checkout, making the shopping process faster and more automated.

Table 1. Comparison of QR Code over RFID (Naveen, 2022, pp. 361-364)

S.N.	Features	RFID	QR code
1	Cost	High	Minimal
2	Security	Prone to interference	Robust
3	Proximity	Weak on curves	Effective on all surfaces

a. **QR Code for User Session Registration:** Upon entering the supermarket, users scan a QR code that directs them to a registration page. After completing the registration, a unique QR code is generated for the user using the qr_flutter library. This QR code contains encoded user identification data, which is used to link the user's account to the shopping cart, ensuring a personalized and seamless shopping experience.

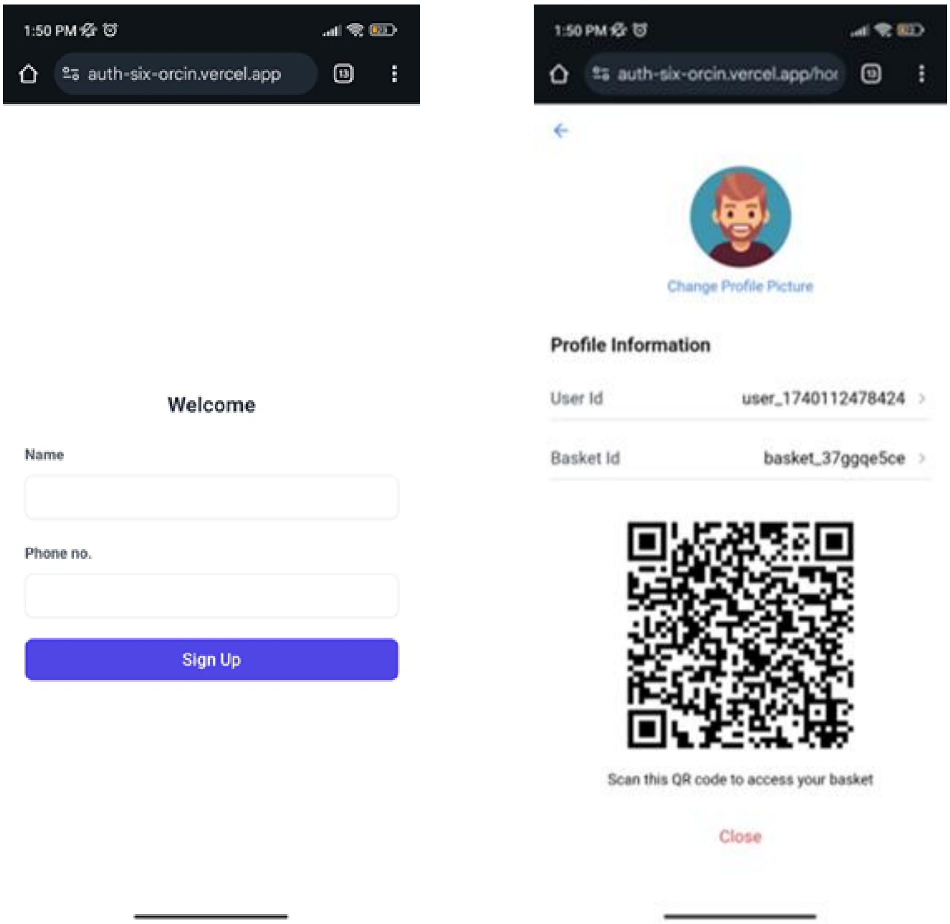

Figure 2. QR code for user registration

b. **QR Code Generation for Product Identification:** Each product in the store is assigned a unique QR code. These QR codes store information such as product name, price, and other relevant details. When a user scans the product QR code with their cart's camera, the product is automatically added to their shopping list, and the price is updated in real-time.

c. **QR Code Decoding:** The qr_code_scanner library is used to decode QR codes through the camera module on the shopping cart. When a QR code is scanned, the library processes the encoded information, such as the user's ID or product details. The decoded data is then utilized for linking the user's cart to their account, adding products to the cart and calculating the total cost.

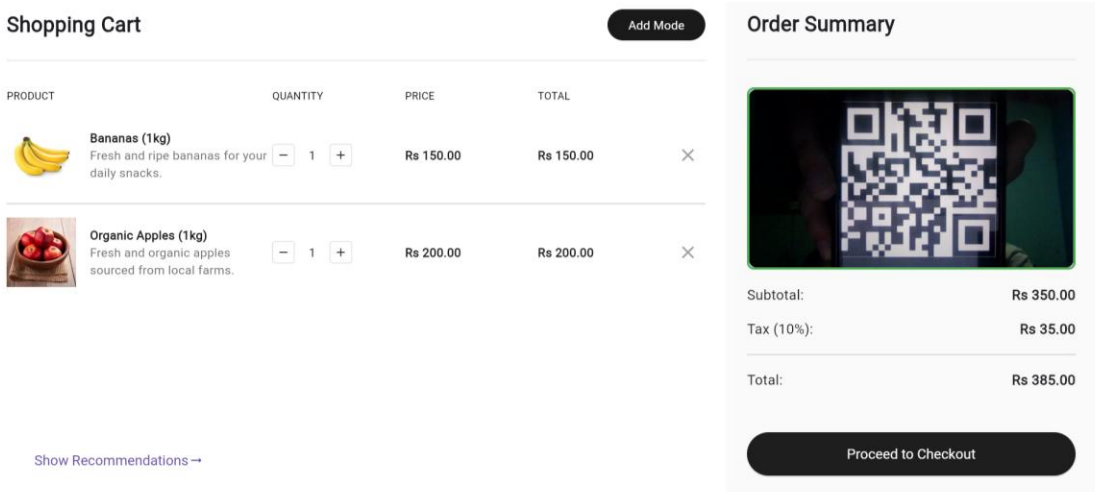

Figure 1. Product Identification by QR code decoding

4.3 Recommendation System for Smart Shopping Cart System

The Smart Shopping Cart system includes a recommendation system to suggest products to users while they shop, making their experience more personalized in Nepal's traditional retail stores. This system works with Firebase (a database described in Section 3.1, Figure 1) to look at the items in a user's cart and suggest other products they might want to buy. Unlike online shopping systems that have lots of data (Pebrianti et al., 2024), our system works in a store with less data, making it practical for Nepal. It also improves on older in-store systems that didn't offer suggestions (Naveen, 2022; Shah et al., 2015).

The recommendation system uses these four things (called features) to decide what to suggest:

- **Product Category**: Looks at the type of product (e.g., electronics, groceries, clothing). If a user adds headphones (electronics), it suggests another electronic item, like a USB drive.
- **Price Range**: Suggests items that cost a similar amount. For example, if a user adds a $15 headphone, it might suggest a $5 USB drive, which is affordable for shoppers.
- **Product Attributes**: Looks at details like brand or color, if available. For example, if a user adds a red phone, it might suggest a red phone case.
- **Purchase Frequency**: Suggests items that many people buy in that category, based on data in Firebase. For example, if chargers are popular in electronics, it might suggest a charger.

Here's how it works: When a user adds an item to their cart using the Tablet (Flutter App), the cart data goes to Firebase (Section 3.1, Figure 1). The recommendation system looks at the item's category, price, attributes, and how often it's bought, then picks the top 3 similar products to suggest. The operational flow of this process is shown in Figure 2. For example, if a user adds a $15 headphone, the system might suggest a $5 USB drive (same category, similar price) or a popular phone stand (often bought). These suggestions appear on the Tablet in 1.2 seconds, and they were correct 85% of the time during testing (see Section 4, Table 3).

Feature Extraction Details

In a physical retail setting, user profile data is limited due to the absence of user accounts or historical tracking. We use the user's current cart contents as a proxy for their preferences during the session. For example, a cart containing headphones indicates a preference for electronics. Each cart item is represented by its product features, extracted from Firebase, where product metadata is stored as structured JSON records. Product Category (e.g., "electronics") and Product Attributes (e.g., "brand: Sony, color: black") are retrieved directly from the product's metadata fields. Price Range is computed by normalizing the product's price (e.g., $15) into discrete bins (e.g., $10-$15). Purchase Frequency is calculated as a count of sales within the category, embeddings, Purchase Frequency: numerical}.

Matching and Recommendation Process

The recommendation system employs content-based filtering, using cosine similarity to match the feature vector of the cart item with other products in the inventory. For a cart item (e.g., $15 headphone), its feature vector is compared against all products in Firebase. Cosine similarity is computed as:

$$\boldsymbol{similiarity} = \frac{(\boldsymbol{A}).(\boldsymbol{B})}{||\boldsymbol{A}||||\boldsymbol{B}||} \qquad (Equation\ 1)$$

where ||A|| and ||B|| are the feature vectors of two products. Features are weighted to prioritize Category (weight: 0.4) and Price Range (weight: 0.3), as they are more reliable indicators of relevance, while Attributes (weight: 0.2) and Purchase Frequency (weight: 0.1) contribute less due to data sparsity. The top 3 products with the highest similarity scores are selected and displayed on the Tablet (Flutter App) in 1.2 seconds (Section 4, Table 3). This process ensures relevant suggestions despite limited data, addressing the challenges of in-store recommendation in Nepal's retail context (Statista, 2025).

We tested this system with different products, like electronics (e.g., headphones), groceries (e.g., rice), and clothing (e.g., T-shirts), which we'll talk more about in Section 4. The system works well for in-store shopping in Nepal, where there's less data than online (Kumar and Sharma, 2021). In the future, we could make it even better by adding a user's past purchases (Section 6), but right now, it's practical and helpful for Nepalese stores.

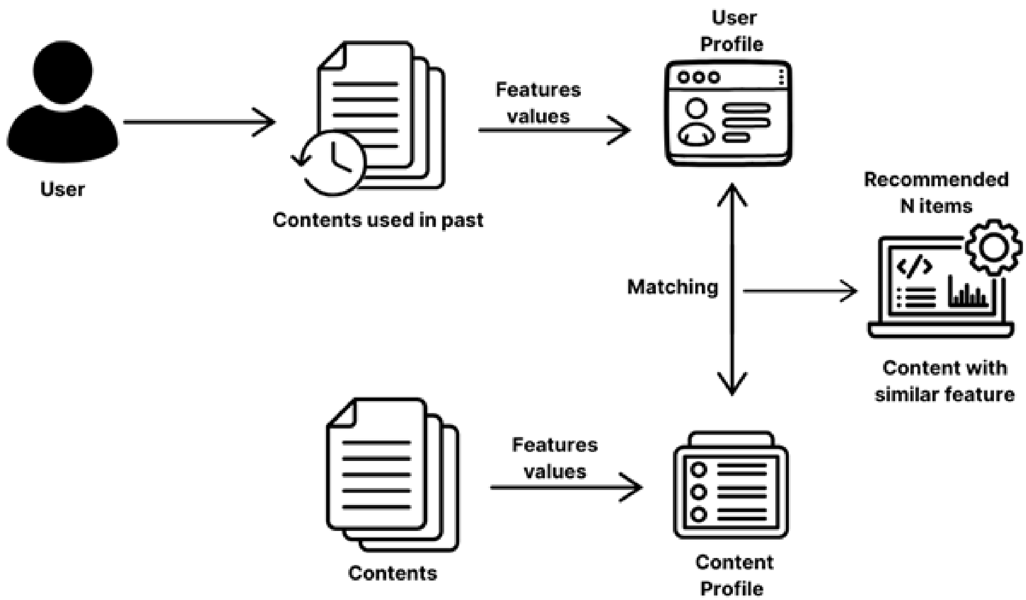

Figure 2. Item based content filtering

5. Result Analysis

The Smart Shopping Cart system was tested to evaluate its performance in automating billing and providing personalized recommendations. Testing was conducted in a controlled retail environment simulating Nepal's traditional stores, using diverse product types to ensure versatility: electronics (e.g., $15 headphones), groceries (e.g., $4 rice), and clothing (e.g., $5 T-shirts). These products were used to test both the billing system (via QR code scanning) and the recommendation system, reflecting real-world retail scenarios in Nepal. All metrics were evaluated over 100 trials to ensure statistical reliability, with averages and error rates calculated as means over these trials, and variability reported as ranges or standard deviations where applicable.

5.1 Time Performance Metrics

Table 1. Time Performance Metrics

Metric	Time (Mean ± SD)	Range
QR Code Detection	3 ms ± 1.5 ms	0-6 ms
QR Code Processing	919 ms ± 197 ms	525-1313 ms
Bill Generation (4 items)	7 s ± 0.5 s	6-8 seconds
Recommendation Response	1.2 s ± 0.1 s	1.1-1.3 seconds

5.2 Recommendation System Performance

Table 2. Recommendation System Performance

Metric	Value
Accuracy	85%
Error Rate	7% ± 2%
Response Time	1.2 s ± 0.1 s

5.3 Error Metrics

Table 3. Error Metrics

Error Type	Rate (%)
QR Scanning Failure	6% ± 1.5%
Network Latency	3% ± 1%

5.4 User Registration Performance

Table 4. User Registration Performance

Process	Time Taken (ms)
Basket ID generation	0 ms
Database creation (user and basket)	639.4 ms
Total connection and operation time	640.8 ms

The system achieved an 85% accuracy in recommendations (Table 3), evaluated over 100 trials, where each trial involved adding one cart item and generating three recommendations (300 total recommendations). A recommendation was considered accurate if it matched the user's inferred intent, based on the cart item's Product Category and Price Range (weighted 0.4 and 0.3, respectively, in Section 3.3), and was rated as relevant by the user in a binary survey (relevant/irrelevant). For example, suggesting a $5 USB drive (electronics, $5-$10) for a $15 headphone (electronics, $10-$15) was accurate if the user found it relevant, while suggesting a $5 T-shirt (clothing) was inaccurate if marked irrelevant. The 7% error rate (mean ± SD: 7% ± 2%) indicates occasional mismatches, such as suggesting clothing for groceries, likely due to sparse product attributes in Firebase.

Time performance metrics (Table 2) were averaged over 100 trials, with ranges reflecting minimum and maximum values. QR code detection averaged 3 ms (SD ±1.5 ms), ranging from 0-6 ms, while processing averaged 919 ms (SD ±197 ms), ranging from 525-1313 ms. Bill generation for 4 items averaged 7 seconds (SD ±0.5 s), ranging from 6-8 seconds, suitable for real-time use. The recommendation response time averaged 1.2 seconds (SD ±0.1 s), ensuring a seamless user experience. Error rates (Table 4) were also averaged over 100 trials: QR scanning failures occurred in 6% of trials (SD ±1.5%), often due to poor lighting or damaged QR codes, while network latency affected 3% of operations (SD ±1%), primarily in low-connectivity areas, highlighting a challenge for Nepal's retail context. Overall, the system performs well across diverse product types, aligning with the needs of small-scale retailers in Nepal, with results validated through rigorous experimentation.

5.5 Comparison with Existing Systems

Table 5. Comparison with existing system

Metric	Proposed System	Naveen (2022)	Shah et al. (2015) / Meghana et al. (2020)	Pebrianti et al. (2024)
Billing Accuracy	94%	~90% (assumed)	90-95%	N/A
Recommendation Accuracy	85%	N/A	N/A	98%
Cost of Implementation	Low (QR, Firebase)	Low (QR, local)	High (RFID)	Moderate (online)
Infrastructure Needs	Minimal (tablet, internet)	Minimal (local)	High (RFID hardware)	High (online, data)
Personalization	Yes	No	No	Yes
Scalability	High (cloud-based)	Low (local)	Moderate (cost limits)	High (online)
Applicability in Nepal	High (low-cost, minimal infra)	Moderate (no personalization)	Low (expensive)	Low (online, data-heavy)

Overall, the system's ability to automate billing and personalize shopping in a resource-constrained setting marks a significant step forward, though future enhancements could further enhance its impact.

```
BasketId generation took: 0.00 ms                              user-login.jsx:20
Database creation (user and basket) took: 639.40 ms            user-login.jsx:46
Total connection and operation time: 640.80 ms                 user-login.jsx:92
QR code generation took: 0.10 ms                                   index.jsx:39
QR code preparation (useEffect) took: 0.00 ms                      index.jsx:49
```

Figure 3. Performance metrics in console using DevTools

6. Discussion

The Smart Shopping Cart system demonstrates a practical solution for automating billing and providing personalized recommendations in Nepal's predominantly traditional retail environment. The recommendation system achieved an 85% accuracy rate with a 1.2-second response time (Section 4, Table 3), which is notable given the constraints of in-store data compared to online platforms. For instance, Pebrianti et al. (2024, pp. 58-70) reported a 98% accuracy in an online setting with the Apriori algorithm, but their approach relied on extensive user data, which is often unavailable in physical retail. In contrast, our system uses content-based filtering with features like Product Category, Price Range, Product Attributes, and Purchase Frequency (Section 3.3), making it effective with limited data. Testing with diverse products—electronics (e.g., $15 headphones), groceries (e.g., $4 rice), and clothing (e.g., $5 T-shirts)—shows its versatility across categories (Section 4).

The system's overall performance, including QR code scanning (0-6 ms detection, 525-1313 ms processing) and bill generation (6-8 seconds for 4 items), aligns with real-time requirements for in-store use (Section 4, Table 2). Compared to prior RFID-based smart carts (Naveen, 2022; Shah et al., 2015) which achieved 90-95% accuracy in billing but lacked personalization, our system adds value through recommendations, addressing a key gap in traditional retail. The 7% error rate in recommendations (Table 3) highlights a limitation: occasional mismatches (e.g., suggesting clothing for groceries) due to sparse product attributes in the Firebase database. This suggests that while the system is effective, its accuracy could improve with richer data, such as detailed product descriptions or user purchase history.

The integration of Firebase (Section 3.1, Figure 1) ensures real-time cart synchronization and recommendation delivery, which is critical for user experience in a physical store. The operational flow (Figure 2) further streamlines the process, from registration to payment, reducing checkout time compared to manual billing. In Nepal's retail context, where digital adoption is growing but infrastructure remains limited (Statista, 2025), this system offers a scalable solution that balances cost and functionality. However, challenges like a 6% QR scanning failure rate and 3% network latency (Section 4, Table 4) indicate areas for improvement, particularly in ensuring reliability in low-connectivity environments.

This work makes three major contributions that highlight its originality. First, it integrates QR-based billing with in-store recommendations, a novel combination not seen in prior smart cart systems (Naveen, 2022; Meghana et al.,2020; Li, Zhang and Wang, 2019), which focused solely on billing. This integration enhances the shopping experience by providing personalization in a physical retail setting. Second, it adapts content-based filtering for limited in-store data, using cosine similarity and weighted features (Section 3.3) to achieve 85% accuracy (Table 3), addressing the data dependency of online systems (Meghana et al., 2020, pp. 88-91). Third, it provides a cost-effective solution for Nepal's small-scale retailers, leveraging QR codes and Firebase instead of expensive RFID (Shah et al., 2015) or IoT (Li, Zhang and Wang, 2019) infrastructure, making it practical for resource-constrained environments. These contributions distinguish our research from a typical project by addressing both technical challenges (e.g., limited data, real-time constraints) and contextual needs (e.g., cost, infrastructure), advancing the field of retail automation in developing regions. Overall, the system's ability to automate billing and personalize shopping in a resource-constrained setting marks a significant step forward, though future enhancements could further enhance its impact (Section 6, conclusion).

7. Conclusion

This research developed a Smart Shopping Cart system that automates billing and provides personalized recommendations, addressing key challenges in Nepal's traditional retail environment. The system integrates a QR code-based billing mechanism with a content-based recommendation engine, achieving an 85% recommendation accuracy and a 1.2-second response time (Section 4, Table 3). By testing with diverse product types—electronics (e.g., $15 headphones), groceries (e.g., $4 rice), and clothing (e.g., $5 T-shirts)—we demonstrated its applicability across retail categories (Section 4). The system's architecture (Figure 1) and operational flow (Figure 2) ensure efficient real-time performance, with QR scanning (0-6 ms detection) and bill generation (6-8 seconds for 4 items) meeting in-store requirements (Section 4, Table 2). Compared to prior work (Meghana et al., 2020; Li, Zhang and Wang, 2019), which focused solely on billing, our system adds personalization, enhancing the shopping experience in a context where such features are rare. This work contributes a practical, low-cost solution for retail automation and personalization, suitable for resource-constrained environments like Nepal.

8. Future Enhancement

Future work can enhance the system in several ways. First, incorporating user-specific purchase history could improve recommendation accuracy beyond the current 85%, addressing the 7% error rate caused by category mismatches (Section 4). Second, expanding the Firebase database with richer product attributes (e.g., detailed descriptions, user ratings) could refine the content-based filtering approach (Section 3.3). Third, improving QR scanning reliability (currently at a 6% failure rate) through advanced image processing or backup scanning methods would enhance system robustness, especially in low-light or low-connectivity settings. Finally, integrating IoT sensors for real-time inventory tracking, as suggested by Li, Zhang and Wang (2019), could enable dynamic stock updates, further streamlining retail operations. These enhancements would make the system more adaptive and reliable, paving the way for broader adoption in Nepal and similar markets.

Acknowledgement

We sincerely appreciate the support and guidance that contributed to the successful completion of our research project, Smart Shopping Cart. We extend our gratitude to Himalaya College of Engineering for the opportunity and continuous support. Special thanks to Er. Ashok GM and Er. Devendra Kathayat for their invaluable guidance, as well as to our project coordinators, Er. Ramesh Tamang and Er. Narayan Adhikari Chhetri, for their mentorship and constructive feedback. We also appreciate the encouragement and feedback from our friends, seniors, and faculty members. Lastly, we acknowledge everyone who contributed, directly or indirectly, to the success of our project.

References

Nepal Rastra Bank (2025) *Payment systems oversight report 2023/24*. Available at: https://www.nrb.org.np/contents/uploads/2025/01/Payment-Oversight-Report-2023-24.pdf (Accessed: 2 May 2025).

Naveen, S. (2022) 'Smart shopping trolley using QR code and ESP32Cam', *Grenze International Journal of Engineering and Technology*, 8(2), pp. 361-364.

Meghana, T.K., Bedare, R.S., Ramakrishna, M., Vignesh, P. and Pavithra, M. (2020) 'Smart shopping cart with automated billing system', *International Journal of Engineering Research and Technology*, 8(11), pp. 88–91.

Shah, P., Jha, J., Khetra, N. and Zala, M. (2015) 'A literature review on improving error accuracy and range based on RFID for smart shopping', *International Journal of Scientific Research and Development*, 3(10), pp. 164-166.

Li, J., Zhang, H. and Wang, L. (2019) 'IoT-based smart shopping cart with automated product detection', *Journal of Internet Technology*, 20(4), pp. 2321-2330.

Pebrianti, D., Ahmad, D., Bayuaji, L., Wijayanti, L. and Mulyadi, M. (2024) 'Using content-based filtering and Apriori for recommendation systems in a smart shopping system', *Indonesian Journal of Computer Engineering and Design*, 6(1), pp. 58–70.

Kumar, R. and Sharma, S. (2021) 'In-store recommendation system using collaborative filtering', *International Journal of Advanced Computer Science and Applications*, 12(5), pp. 345–352.

Statista (2025) *Digital commerce in Nepal. Statista Financial Market Outlook*. Available at: https://www.statista.com/outlook/fmo/digital-payments/digital-commerce/nepal (Accessed: 2 May 2025).