



Author: Keshav Thapa

Student ID: 25126463

Date: 24/01/2026

Course: Database and Web Application Module

Code: DIG5127

Word count: 1800 Words

Page count: 11 Pages

Report: ERM Report

Electronic Store Management System

A Web-Based E-Commerce Platform

Executive Summary

This project presents a complete web-based solution for managing an electronics retail business. The system was built to handle the daily operations of an online electronics store, including product listings, customer orders, and payment processing. Through careful planning and development, we created a platform that serves three distinct user types: customers who shop online, sellers who manage inventory, and administrators who oversee the entire operation.

1. Introduction

1.1 Background

The retail electronics industry has seen tremendous growth in online sales over recent years. Traditional brick-and-mortar stores are increasingly moving towards digital platforms to reach wider audiences and provide convenient shopping experiences. However, managing an online electronics store involves numerous challenges - from maintaining accurate inventory records to processing payments securely and tracking orders efficiently.

1.2 Project Motivation

We developed this system after observing the difficulties faced by small to medium-sized electronics retailers who struggle with managing their online presence. Many existing solutions are either too expensive or too complex for these businesses. Our goal was to create an accessible, user-friendly platform that addresses the core needs of electronics retail without unnecessary complications.

1.3 Project Objectives

The main objectives we set out to achieve were:

- Build a secure user authentication system supporting multiple user roles
- Develop an intuitive product browsing and shopping experience
- Implement reliable order tracking from placement to delivery
- Create a flexible payment system supporting multiple payment methods
- Provide comprehensive management tools for sellers and administrators
- Ensure data integrity through proper database design and relationships

2. System Overview

2.1 Core Functionality

The Electronic Store Management System operates as a multi-user platform where different types of users have access to specific features based on their roles. Customers browse products organized by categories, add items to their shopping cart, and complete purchases through their preferred payment method. Sellers manage their product listings, update inventory levels, and track sales performance. Administrators maintain overall control, managing user accounts, monitoring transactions, and ensuring smooth operations.

2.2 User Roles and Responsibilities

Customers (Buyers) Customers represent the primary end-users of the system. After creating an account, they can browse through various electronics categories, search for specific products, view detailed specifications and pricing, and make purchases. The system maintains their order history and allows them to track current orders from placement through delivery.

Sellers Sellers are users who list and manage products on the platform. They have access to inventory management tools where they can add new products, update existing listings, adjust prices, and monitor stock levels. Sellers can view orders for their products and track payment statuses.

Administrators Administrators have the highest level of access and responsibility. They manage user accounts across all roles, oversee product categories, monitor all transactions and payments, handle customer service issues, and generate reports on system performance.

3. Database Design

3.1 Database Architecture

The database follows a relational model with six primary tables: USERS, CATEGORY, PRODUCTS, ORDERS, ORDER_ITEMS, and PAYMENTS. This structure was chosen to minimize data redundancy while maintaining flexibility for future expansion.

3.2 Table Structures

USERS Table This table stores information about everyone who uses the system. Each user has a unique identifier (user_id), along with their full name, username, email address, and encrypted password. We store phone numbers and addresses for delivery purposes. The role field determines whether someone is a buyer, seller, or admin. We also include security questions and answers for account recovery. Additional fields track when accounts were created, last updated, and when users last logged in.

CATEGORY Table Categories help organize products into logical groups like Mobile Phones, Laptops, Gaming equipment, and Home Appliances. Each category has a unique ID, name, description, and associated image. This makes it easier for customers to browse products and for the system to maintain organized inventory.

PRODUCTS Table The products table contains detailed information about every item available for purchase. Each product links to a category and optionally to the seller who listed it. We store the product name, description, price, and current stock quantity. Product images are referenced through URLs. The `is_active` flag allows sellers to temporarily hide products without deleting them.

ORDERS Table When a customer makes a purchase, we create an order record that captures all relevant information - who placed the order, when it was placed, where it should be delivered, and the total amount. Orders progress through different statuses: pending, processing, shipped, delivered, cancelled, or returned. This allows everyone involved to track the order's current state.

ORDER_ITEMS Table Since a single order can contain multiple products, we use a separate table to track individual items within each order. This table records which products were ordered, in what quantities, and at what price (we store the price at purchase time since product prices can change later).

PAYMENTS Table The payments table tracks financial transactions for orders. We support multiple payment methods including cash on delivery, eSewa, Khalti, Fonepay, IME Pay, card payments, and bank transfers. Each payment has a status indicating whether it's pending, pending verification (for screenshot-based payments), paid, failed, or rejected. For digital payments, customers can upload payment screenshots which administrators review and verify.

3.3 Relationships Between Tables

Users and Orders (One-to-Many) A single customer can place many orders over time, but each order belongs to exactly one customer. This relationship lets us maintain complete purchase histories and provide personalized shopping experiences.

Categories and Products (One-to-Many) Each category can contain numerous products, but every product belongs to one category. This organizational structure makes inventory management and product browsing more efficient.

Orders and Order Items (One-to-Many) An order typically contains multiple products, so we connect them through order items. Each order item represents one product type with its quantity and price in that particular order.

Products and Order Items (One-to-Many) Popular products appear in many different orders. This relationship helps us track which products sell best and manage inventory based on demand.

Orders and Payments (One-to-One) Every order must have exactly one payment record, ensuring that all transactions are properly documented and tracked.

4. Key Features

4.1 User Management System

The registration process requires users to provide their full name, a unique username, email address, strong password, phone number, and delivery address. Passwords must be at least 8 characters long and include capital letters and special characters. We hash all passwords before storage for security.

During login, users can authenticate using either their username or email along with their password. The system verifies credentials and grants access based on the user's role. Users can update their profile information, manage delivery addresses, and view their complete order history through their account dashboard.

4.2 Product and Category Management

Administrators create and manage product categories to keep the store organized. Each category can contain an unlimited number of products. Sellers add products by providing names, detailed descriptions, pricing, stock quantities, category assignments, and product images.

Customers browse products through an intuitive interface that allows filtering by category, searching by keywords, sorting by price or popularity, and viewing detailed product information including specifications and availability.

4.3 Shopping Cart and Checkout

When customers find products they want to purchase, they add them to their shopping cart. The cart allows reviewing items, adjusting quantities, and removing unwanted products before checkout. During checkout, customers confirm their delivery address and see a complete order summary including the total amount.

4.4 Order Processing

After placing an order, customers receive confirmation and can track its status through their account. Orders move through several stages: pending (just placed), processing (being prepared), shipped (on the way), delivered (completed), cancelled (customer or admin cancelled), or returned (customer sent back).

Sellers see orders for their products and can update order statuses as they process and ship items. Administrators have visibility into all orders and can intervene when issues arise.

4.5 Payment System

We support diverse payment methods to accommodate different customer preferences:

- Cash on Delivery (COD): Customers pay when they receive the order
- Digital Wallets (eSewa, Khalti, Fonepay, IME Pay): Popular payment apps in Nepal
- Card Payments: Credit and debit cards

- Bank Transfers: Direct bank-to-bank payments

For digital payments where customers pay through apps, they can upload payment screenshots. These screenshots go into a verification queue where administrators review and approve or reject them. Once verified, orders are marked as paid and proceed to fulfillment.

5. Technical Implementation

5.1 Security Measures

Security was a primary concern throughout development. We implement password hashing using bcrypt with a cost factor of 10, making it computationally expensive to crack passwords even if the database is compromised. SQL injection attacks are prevented through prepared statements and parameterized queries. User input is sanitized and validated on both client and server sides.

Session management uses secure, randomly generated session tokens. We implement role-based access control to ensure users only access features appropriate to their role. Payment screenshots and sensitive data are stored securely with restricted access.

5.2 Data Integrity

Foreign key constraints maintain referential integrity throughout the database. When a user is deleted, their orders are removed automatically (CASCADE). When a seller account is deleted, their products remain but with the seller_id set to NULL (SET NULL). This prevents orphaned records and maintains data consistency.

We use transactions for critical operations like order placement and payment processing. If any step fails, the entire transaction rolls back, preventing partial updates that could leave the database in an inconsistent state.

6. System Benefits

6.1 For Customers

Customers enjoy convenient 24/7 shopping access from anywhere with internet connectivity. They can compare products easily, read detailed specifications, check current stock availability, and choose their preferred payment method. Order tracking keeps them informed about delivery status, and their purchase history helps with reordering and warranty claims.

6.2 For Sellers

Sellers gain access to tools that simplify inventory management. They can quickly update product information, adjust prices based on market conditions, track which products sell best, monitor stock levels, and prevent overselling through automated inventory updates. The system helps them reach more customers without the overhead of physical stores.

6.3 For Administrators

Administrators benefit from centralized control over the entire platform. They can manage user accounts, resolve disputes, monitor transactions, identify trends through sales data, ensure payment verification, and maintain system integrity. The dashboard provides real-time insights into business performance.

7. Challenges and Solutions

7.1 Payment Verification

One significant challenge was handling payments in contexts where direct API integration wasn't feasible. We solved this by implementing a screenshot upload and verification system. While this requires manual review, it provides flexibility for customers using various payment methods while maintaining transaction records.

7.2 Inventory Management

Keeping inventory accurate when multiple customers might purchase the same product simultaneously required careful transaction handling. We implemented database locks during checkout to prevent overselling and ensure stock quantities remain accurate.

7.3 Role-Based Access

Creating a system that appropriately limits features based on user roles while maintaining code simplicity required thoughtful design. We implemented middleware that checks user roles before allowing access to specific functions, providing security without duplicating code.

8. Future Enhancements

Several improvements could enhance the system further:

Advanced Analytics: Implementing detailed sales reports, customer behavior analysis, inventory forecasting, and revenue tracking would help sellers and administrators make better business decisions.

Customer Reviews: Allowing customers to rate and review products would help others make informed purchasing decisions and provide sellers with valuable feedback.

Wishlist Functionality: Customers could save products they're interested in for future purchase, improving user experience and potentially increasing sales.

Email Notifications: Automated emails for order confirmations, shipping updates, and payment reminders would keep customers informed and reduce support queries.

Multi-Currency Support: Adding support for different currencies would enable international expansion.

Mobile Application: A dedicated mobile app would provide better user experience on smartphones and tablets.

9. Conclusion

The Electronic Store Management System successfully addresses the core requirements of online electronics retail. Through careful database design, we ensured data integrity and scalability. The multi-role architecture provides appropriate tools for customers, sellers, and administrators while maintaining security and privacy.

The system demonstrates how thoughtful planning and execution can create practical solutions for real-world business challenges. By focusing on essential features and user experience, we developed a platform that's both powerful and accessible.

This project provided valuable experience in database design, web application development, security implementation, and user interface design. The challenges encountered and solved will inform future development efforts. While there's always room for improvement and additional features, the current system provides a solid foundation for electronic commerce operations.

The success of this project validates the approach of building focused, user-centric systems that solve specific problems effectively rather than trying to include every possible feature. As the electronics retail landscape continues evolving, this system is positioned to adapt and grow with changing business needs.

Appendix A :

ER DIAGRAM ELECTRONIC STORE MANAGEMENT SYSTEM

PRODUCTS		
int	product_id	PK
int	seller_id	FK
int	category_id	FK
varchar	product_name	
varchar	slug	
text	description	
decimal	price	
int	stock	
varchar	product_image	
tinyint	is_active	
datetime	created_at	
datetime	created_at	

CATEGORY		
int	category_id	PK
varchar	category_name	
varchar	description	
varchar	category_image	
datetime	created_at	

ORDERS		
int	order_id	PK
int	user_id	FK
datetime	order_date	FK
decimal	total_amount	
text	shipping_address	
num	status	

USERS		
int	user_id	PK
varchar	full_name	
varchar	username	
varchar	email	
varchar	password	
varchar	phone_no	
text	address	
varchar	profile_picture	
enum	role	
varchar	security_question	
varchar	security_answer	
tinyint	is_active	
datetime	created_at	

ORDER_ITEMS		
int	order_item_id	PK
int	order_id	FK
int	product_id	FK
int	quantity	
decimal	unit_price	
datetime	created_at	

text	notes	
datetime	created_at	
datetime	updated_at	
int	product_id	
string	contact_no	

datetime	updated_at	
datetime	last_login	

PAYMENTS		
int	payment_id	PK
int	order_id	FK
decimal	amount	
enum	payment_method	
enum	payment_status	
varchar	payment_screenshot	
varchar	transaction_id	
datetime	payment_date	
text	notes	
datetime	created_at	
timestamp	verified_at	
int	verified_by	
text	rejection_reason	

Appendix B :

