



PHP-Based Stationery Sales Information System at Ufaira Printing Sigambal, Labuhanbatu

Majidur Rahman Hasibuan^a, Farida Gultom^b, Muhammad Bobbi Kurniawan Nasution^c

Department of Informatics Engineering, Faculty of Engineering, Universitas Efarina, Pematangsiantar, Indonesia

Article Info

Keywords:

Stationery Sales
Inventory
E-Commerce

ABSTRACT

Ufaira Printing in Sigambal, Labuhanbatu previously relied on manual records to manage stationery products, inventory information, customers, and sales related administration. This practice created inconsistent stock information, risk of lost records, delayed reporting, and operational decisions based on estimates rather than verified data. This study aimed to design, implement, and functionally test a PHP based stationery sales information system supported by a MySQL database. Development followed the waterfall model through requirements analysis, system and software design, coding, testing, and maintenance planning. Requirements were obtained through observation, literature study, and examination of the store's product and sales activities. The design was documented using a registration flowchart, context diagram, data flow diagram, entity relationship diagram, database tables, and interface plans. The implemented application provides customer registration, role based authentication, product and category administration, shipping expedition data, product catalog pages, and category specific displays for stationery, ink, office automation equipment, files, and furniture. Black box testing covered registration, login, logout, and product management access. All eight nonduplicative cases reported in the thesis produced the expected responses. The findings indicate that the system can organize product information and support more consistent catalog administration. However, the current evidence does not establish complete transaction processing, automatic inventory reconciliation, payment handling, security robustness, usability across users, or long term operational impact. Future work should complete these functions and evaluate data accuracy, task completion, and user experience after deployment.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Majidur Rahman Hasibuan
Department of Informatics Engineering
Universitas Efarina
Pematangsiantar, Indonesia
Email: rmajidur549@gmail.com

1. INTRODUCTION

Small retail businesses depend on accurate product, stock, customer, and transaction information to serve buyers and control operations. Product availability must be updated when items are received, offered, or sold, while prices and categories remain consistent across records. When these activities rely on notebooks or disconnected files, employees may record an item differently, overlook changes, or prepare reports from incomplete data. Manual work also makes it difficult to trace an error because earlier entries may not be

connected with the summary used for a decision. These conditions can delay service, produce unreliable stock balances, and encourage decisions based on estimates. A structured sales information system is therefore relevant to accurate retail administration [1].

Ufaira Printing in Sigambal, Labuhanbatu began as a computer rental business and expanded into printing services and stationery sales. It increased the variety of products handled by the store, including writing instruments, ink, paper, office automation equipment, files, and furniture. A broader catalog requires more than item names because each product must have a category, price, description, stock value, and information presented to customers. As records increase, manual recording becomes less dependable for checking availability or preparing a current catalog. Separate notes can also create a gap between internal store information and information communicated to buyers. The business therefore needed an application that could organize product administration and customer access within one consistent data environment.

A web based sales system can connect customer access with store administration through an application. Customers can register, authenticate, browse categories, and view product information, while administrators maintain products, categories, expedition options, and related records. Role separation matters because public catalog functions should remain accessible without exposing controls that change stored data. Work on web based marketplace development shows that an online interface can broaden access to organized product information [2]. Research on stationery sales systems indicates that structured databases and web interfaces can reduce repeated manual work and improve catalog availability [3]. For Ufaira Printing, the approach links data maintained by administrators with information presented on customer pages instead of requiring each output to be maintained independently.

PHP and MySQL provide a technical foundation for this integration. PHP processes registration and login requests, controls page access, receives product forms, and prepares database results for presentation in a browser. MySQL stores user, product, category, expedition, and transaction data so the application can retrieve consistent records across functions. A relational structure also allows a product to reference its category without repeatedly storing the description in every interface. Sales application research shows that server logic and structured databases can support accessible information systems [4]. However, technology selection does not ensure reliable operation. Requirements must be defined, fields validated, relationships protected, and visible functions tested against expected responses before the application is considered ready for operational use.

System development must translate the store's operational problem into traceable software components. If coding begins before customer roles, product attributes, navigation, and outputs are understood, the finished pages may not support workflows. The waterfall model organizes development into requirements analysis, system and software design, coding, testing, and maintenance [5]. This sequence relates a need to a design artifact, implemented module, and verification activity. Flowcharts document operational sequences, context diagrams define external interaction, data flow diagrams describe information movement, and entity relationship diagrams represent database associations [6]. Interface plans complement these models by showing how customers and administrators reach each function. Together, the artifacts provide a path from business needs to program behavior and reduce dependence on unrecorded design assumptions.

This study aims to develop and functionally evaluate a PHP based stationery sales information system for Ufaira Printing Sigambal. The scope includes customer registration, role based login and logout, administrator product and category management, expedition data, customer catalog pages, and category specific product displays. The article emphasizes the Chapter IV interfaces and black box cases documented in the thesis. Its contribution is a foundation for organizing product information and presenting it consistently. The prototype is not claimed as a complete production commerce workflow. Stock deduction, concurrent transactions, payment confirmation, shipping calculation, order cancellation, report reconciliation, comprehensive security, and usability across multiple users require further development and testing. These limits keep the conclusions aligned with the reported functions and evidence.

2. RESEARCH METHOD

2.1. Research setting and data collection

The study was conducted at Ufaira Printing, Jl. Kancil, Sigambal, Rantau Selatan District, Labuhanbatu Regency, North Sumatra. Data collection used observation, literature study, and examination of the store's product and sales activities.

The requirements analysis focused on customer registration, role-based login, product and category maintenance, shipping-expedition data, catalog presentation, and report-related information. The database named "toko" contained user, product, category, and transaction tables.

2.2. Waterfall procedure

The development procedure followed five connected waterfall stages: requirements analysis, system and software design, coding, testing, and maintenance. Each stage used the output of the preceding stage so that the application could be developed from documented store activities rather than from assumptions about customer or

administrator needs. The sequence also provided a clear basis for checking whether each implemented page corresponded to an identified requirement.

The first stage was requirements analysis. Observation, literature study, and examination of the store's product and sales activities were used to identify the information that the system needed to handle. The analysis covered customer registration, authentication according to user role, product and category maintenance, expedition data, catalog presentation, database storage, and report related information. The result was a functional scope that distinguished customer access from protected administrator operations and defined the principal data required by both sides of the application.

The second stage was system and software design. The requirements were translated into a registration flowchart, context diagram, level zero data flow diagram, entity relationship diagram, database tables, and interface plans. The process models described interactions among customers, administrators, application functions, and stored data. The database design organized users, products, categories, expeditions, and transaction related records, while the interface plans arranged navigation, forms, actions, and catalog outputs. These artifacts established how information would move and how each user would access the required functions before coding began.

The third stage was coding or implementation. The approved design was converted into PHP pages connected to the MySQL database [7]. Customer modules handled registration, login, and catalog access, whereas administrator modules supported product, category, expedition, and related data management. Form values were linked with database operations, and stored product records were transformed into catalog pages grouped by category. Implementation remained focused on the functions documented during analysis and design so that visible behavior could later be compared with the original requirements.

The fourth stage was testing. Black box testing was used to examine externally visible responses without relying on the internal program structure. The documented cases covered correct, incorrect, and completed customer registration; valid, invalid, and empty login input; logout; and access to the product management list. A case was considered successful when the observed page or message matched the expected result. This approach demonstrated the connection of the reported functions, although it did not replace broader security, performance, transaction, inventory, or usability testing.

The final stage was maintenance planning. After the tested functions were prepared for use, later corrections and adaptations were recognized as continuing responsibilities. Maintenance may include repairing defects, refining validation, adjusting product or category structures, strengthening authentication and authorization, improving reports, and responding to changes in store procedures. Database backup, recovery, and review of user feedback are also important because the usefulness of the system depends on accurate records over time. Treating maintenance as part of the development cycle helps the application remain aligned with operational needs after its initial implementation.

2.3. Requirements and implementation environment

The system includes customer-facing and administrator-facing functions. Table 1 summarizes the principal requirements derived from the thesis.

Table 1. Functional requirements of the stationery sales system

No.	Actor	Required Function	Expected Output
1	Customer	Register and authenticate	A reusable customer account and authorized session
2	Administrator	Add and update product data	Current product catalog records
3	Administrator	Manage categories and expedition options	Organized catalog and shipping information
4	Customer	Browse product categories	Accessible stationery and office-product information
5	System	Store users, products, categories, and transactions	Structured relational records
6	System	Display processed information	Readable catalog and administrative output

The software and hardware environment is summarized in Tables 2 and 3. The application was authored in Adobe Dreamweaver CS6 and executed through a local XAMPP/MySQL environment.

Table 2. Software environment

No.	Software	Version or Purpose
1	Microsoft Windows	Windows 11 Home Single, 64-bit
2	Adobe Dreamweaver	CS6; code authoring
3	MySQL / XAMPP	Version 3.3.0; database and local web server

Table 3. Hardware environment

No.	Hardware	Specification
1	Laptop	HP
2	Processor	Intel Celeron N4120, 1.10 GHz
3	Memory	4 GB RAM
4	Storage	238 GB SSD
5	Display	15-inch LCD

2.4. Verification approach

Black-box testing examined externally visible responses without inspecting internal code paths [8]. The thesis documented cases for customer registration, login and logout, and product-management access. A case passed when the observed behavior matched the expected result.

The registration cases used correct, incorrect, and completed input. Login/logout cases checked valid, invalid, and empty credentials and the logout action. Product-management testing checked whether the product list could be opened. Ambiguous duplicate login rows in the source were not counted as independent product-management evidence.

3. RESULTS AND DISCUSSION

3.1. Implemented user interfaces

The implemented system contains input interfaces for customers and administrators and output pages for the product catalog. Six representative Chapter IV screens are reproduced individually below.

Figure 1 shows the customer registration page. The form collects identity, contact, address, email, password, and profile information and stores a reusable account so the customer can return without repeating registration.

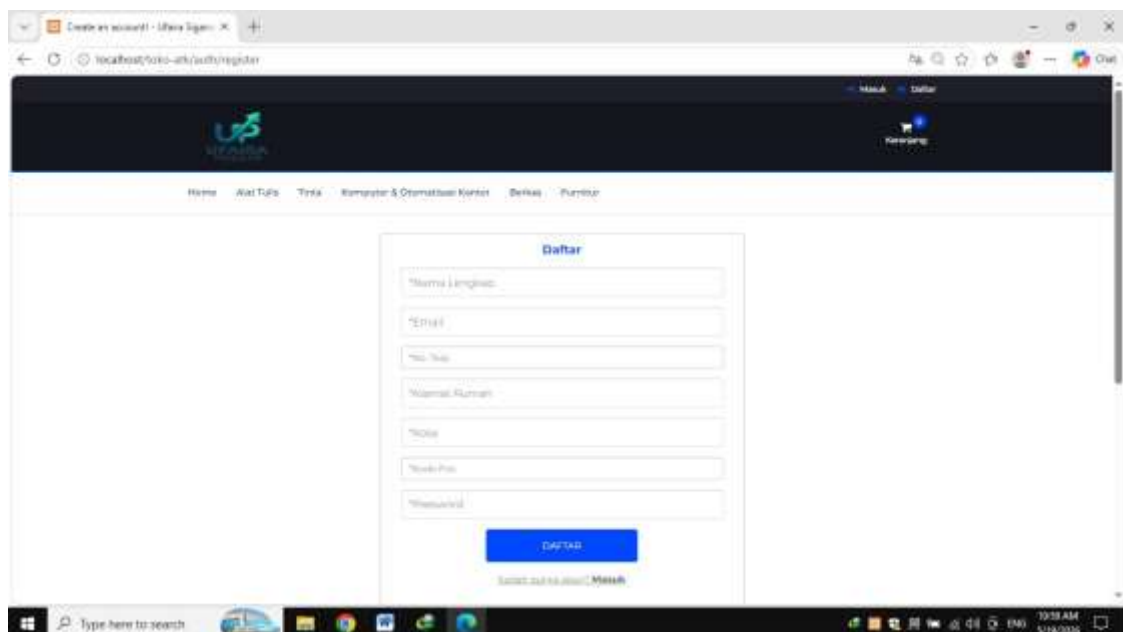


Figure 1. Customer registration page

Figure 2 presents the shared login page. The system validates registered credentials and then directs the user to the customer-facing catalog or the administrator dashboard according to the assigned role.

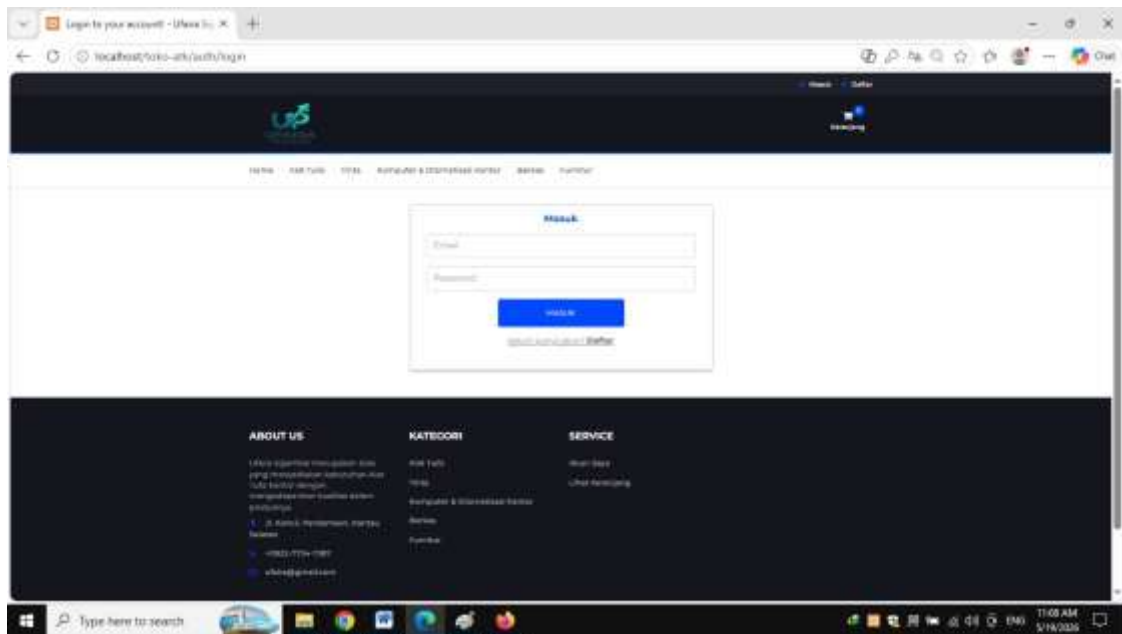


Figure 2. Customer and administrator login page

Figure 3 shows the add-product page used by an administrator. Product identity, category, stock, price, description, and image data can be entered in one form, connecting catalog presentation with stored inventory information.

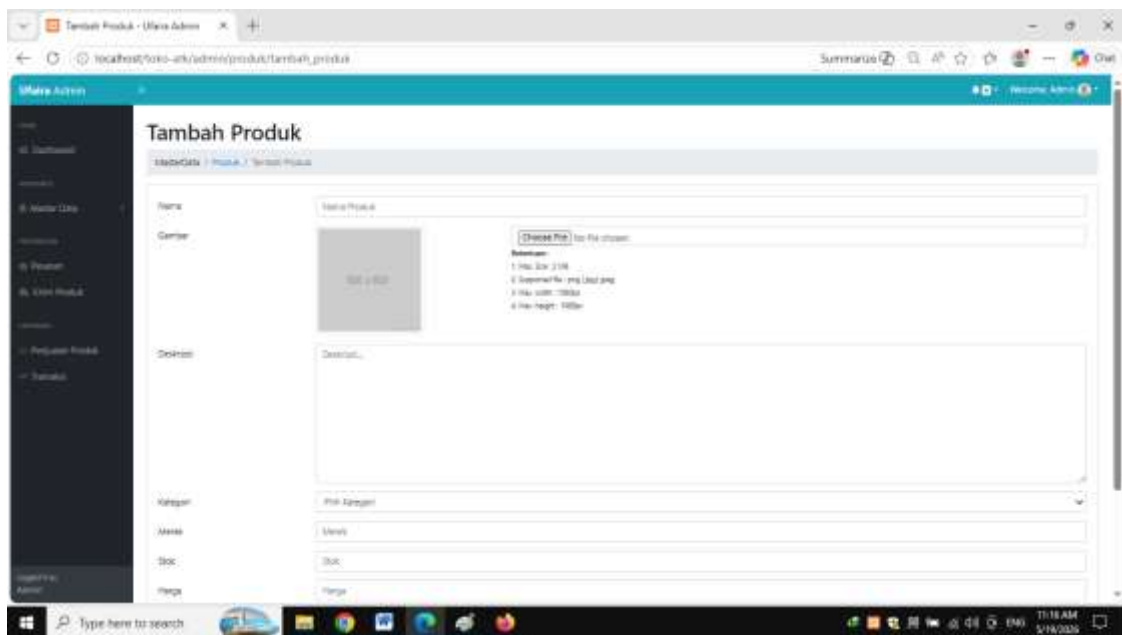


Figure 3. Add-product administration page

Figure 4 presents category management. The administrator can review existing categories and add a new category through a focused dialog, allowing products to be grouped consistently for browsing and reporting.

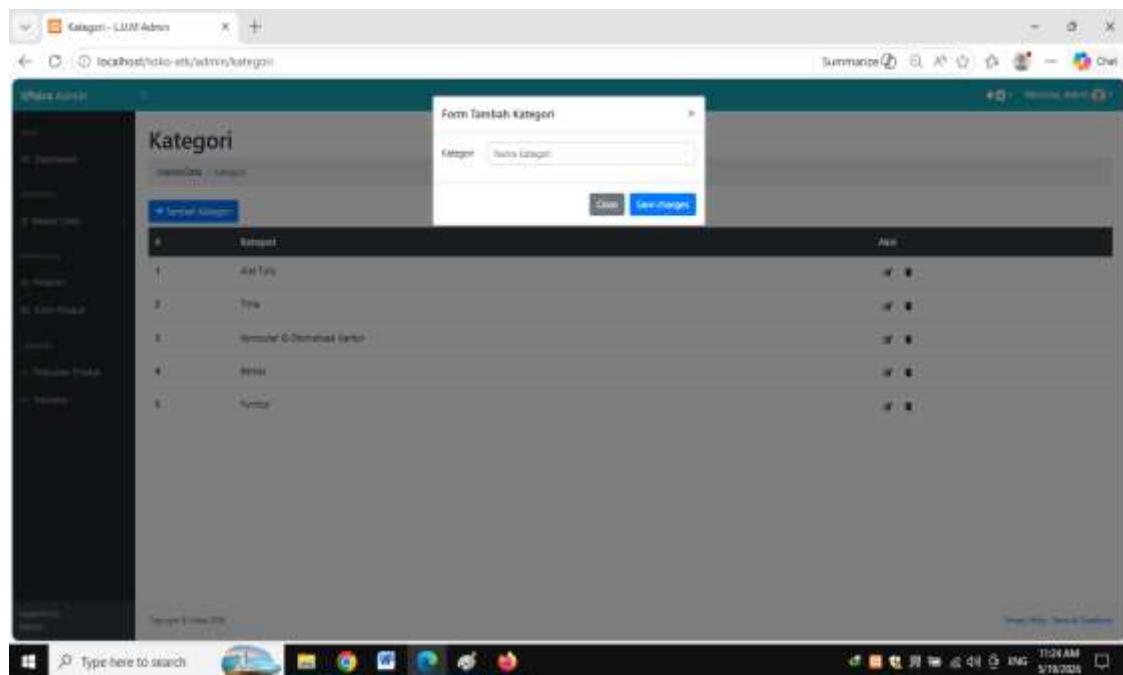


Figure 4. Product-category management page

Figure 5 shows the customer-facing home page. New and best-selling product sections present current offerings, while the navigation bar connects users to catalog categories, account functions, and store information.

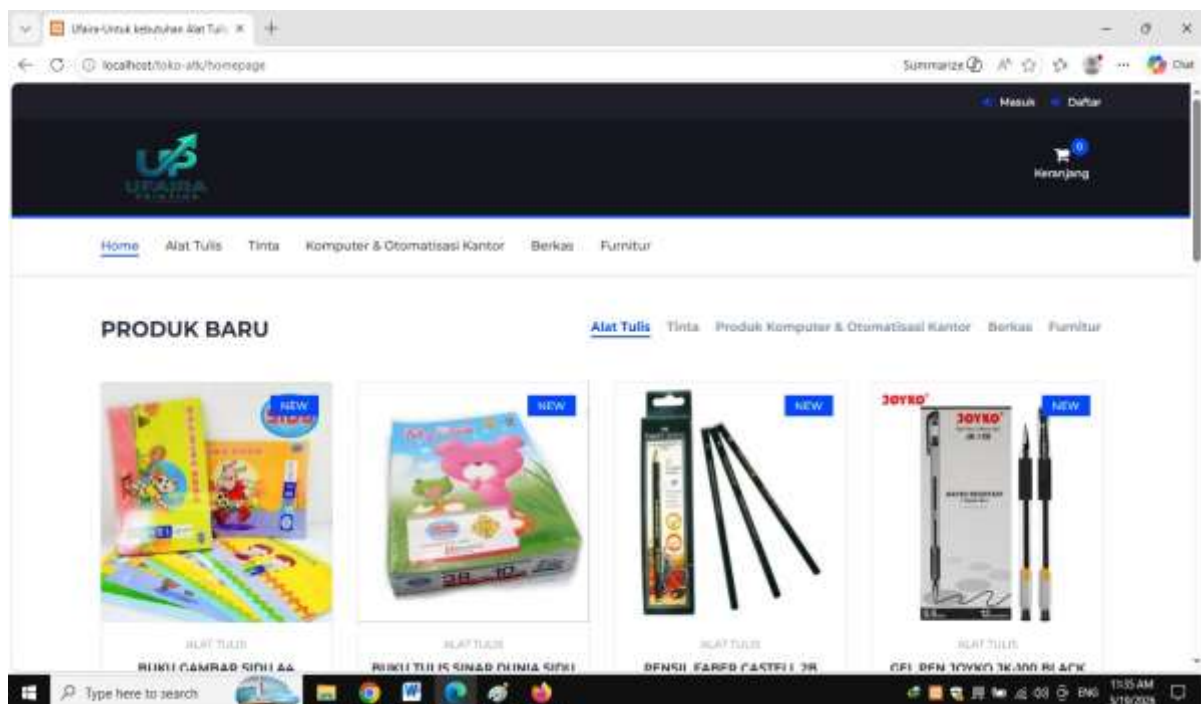


Figure 5. Customer-facing product home page

Figure 6 displays the ink category page. Product cards organize available ink items with names, prices, and purchase actions, demonstrating how database records are transformed into a category-specific catalog output.

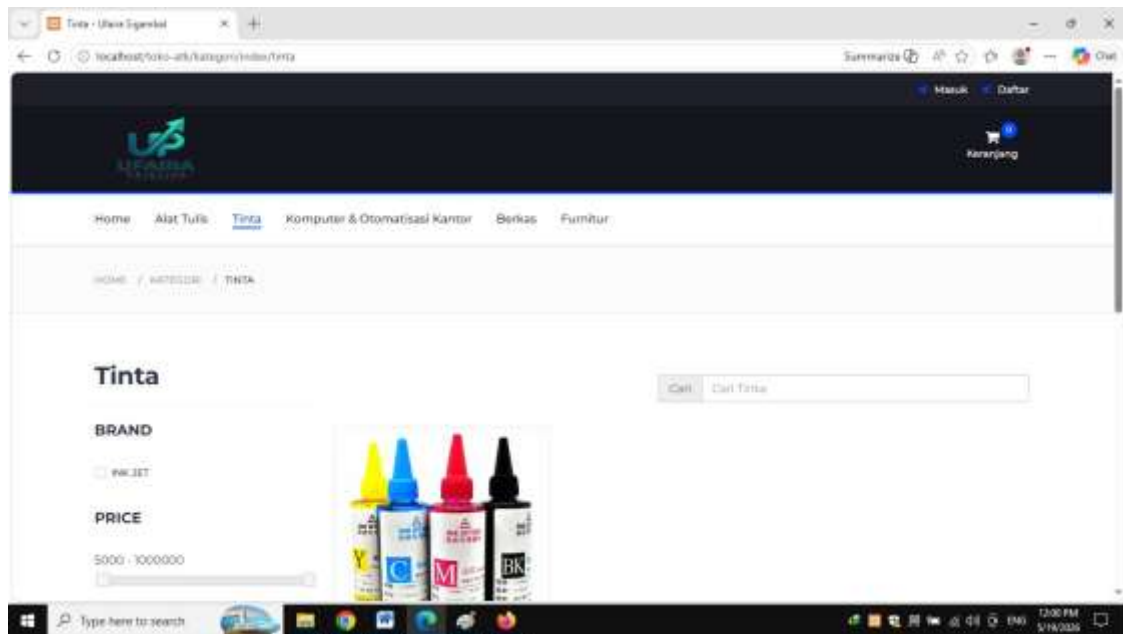


Figure 6. Ink product category page

3.2. Functional verification results

Table 4 consolidates the non-duplicative black-box cases reported in the thesis. All eight retained cases matched the expected behavior, giving an 8/8 pass result within the documented functional scope.

Table 4. Black-box functional test results

No.	Test Scenario	Expected Result	Observed Result	Status
1	Submit correct registration data	User data are stored and login becomes available	Data stored as expected	Passed
2	Submit incorrect or incomplete registration data	Data are rejected and required-field message appears	Validation message appeared	Passed
3	Complete customer registration	Success message and login page are displayed	Success flow displayed	Passed
4	Submit valid email and password	The appropriate user home page opens	Home page opened	Passed
5	Submit invalid email or password	Login remains open and an error message appears	Error message appeared	Passed
6	Submit empty credentials	Login page remains available	Login page remained	Passed
7	Select Logout	Session ends and login page opens	Returned to login	Passed
8	Open product-data menu	Product list is displayed	Product list displayed	Passed

3.3. Discussion

The implemented pages connect customer identity, protected administration, product organization, and catalog output. This integration addresses the original problem more directly than a standalone promotional page because product records entered by the administrator become information visible to customers.

Registration and role-based login provide a basis for separating customer and administrator tasks. However, authentication should not be considered secure solely because valid and invalid credentials produce different responses. Password hashing, session expiration, authorization checks, input validation, and protected transport require dedicated verification [9].

The eight retained black-box cases passed, showing that core registration, authentication, logout, and product-list behaviors were connected. The scope remains narrow: stock deduction, concurrent purchases, transaction totals, payment confirmation, shipping calculation, canceled orders, image validation, and report reconciliation were not documented. Broader test design is needed before operational deployment [10].

Database constraints should protect unique emails, product identifiers, category relationships, non-negative stock, and transaction consistency. MySQL provides constraints and transactions that can support these controls

[11]. PHP code should also separate presentation, business rules, and data access to improve maintainability [12].

Future evaluation should compare inventory discrepancies, time to update a product, report preparation time, catalog accuracy, and user task completion before and after deployment. Responsive presentation and accessibility should be checked on common mobile devices because many customers will access the catalog through a phone [13]. Dreamweaver-based implementation experience provides a reference for later interface maintenance [14]. General web-system development research highlights the need for maintainable architecture [15]. Database application work also demonstrates the importance of validated relational records [16]. These steps would turn the current functional prototype into a more reliable retail service.

4. CONCLUSION

A PHP- and MySQL-based stationery sales information system was developed for Ufaira Printing Sigambal using the waterfall model. The system implements customer registration and login, administrator product and category management, expedition data, and customer-facing product catalogs. Eight non-duplicative black-box cases documented in the thesis produced the expected results. The application provides a structured alternative to manual product information and supports faster catalog administration. Before production use, the store should complete transaction and inventory workflows, strengthen security and data integrity, broaden testing, and evaluate usability and operational impact.

REFERENCES

- [1] I. J. Informatika and M. Teknologi, "Sistem informasi penjualan berbasis web pada Toko Ripani," vol. 25, pp. 117-124, 2023, doi: 10.23969/infomatek.v25i2.10350.
- [2] D. A. E. Sirait and D. T. Seabtian, "Sistem informasi e-marketplace cinderamata Sampit berbasis web," vol. 10, no. 1, 2019.
- [3] B. Subana and G. S. Harliyan, "Sistem informasi penjualan alat tulis kantor berbasis web pada Toko UD Barokah," vol. 6, no. 2, pp. 286-293, 2022.
- [4] M. Rezki and L. Nur, "Perancangan sistem informasi penjualan ATK berbasis Android," vol. 1, no. 4, pp. 302-316, 2024.
- [5] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill, 2020.
- [6] B. A. Herlambang, V. Ana, and V. Setyawati, "Perancangan data flow diagram sistem pakar berbasis web," pp. 78-85.
- [7] M. Ridwan, T. H. Sinaga, and M. Elsera, "Penerapan framework CodeIgniter dalam perancangan sistem," vol. 3, no. 1, 2022.
- [8] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*, 3rd ed. Hoboken, NJ, USA: Wiley, 2011.
- [9] OWASP Foundation, "OWASP Application Security Verification Standard 4.0.3," 2021.
- [10] P. Ammann and J. Offutt, *Introduction to Software Testing*, 2nd ed. Cambridge, UK: Cambridge University Press, 2016.
- [11] Oracle Corporation, "MySQL Reference Manual," MySQL Documentation.
- [12] The PHP Group, "PHP Manual," PHP Documentation.
- [13] C. Perdana and M. A. Wijaya, "Implementasi Framework Bootstrap 5 pada perancangan front-end website," vol. 2, no. 1, pp. 30-43, 2024.
- [14] D. Retnosari, "Implementasi sistem jasa pembangunan dan desain rumah menggunakan Adobe Dreamweaver," vol. 2, no. 1, pp. 30-34, 2016.
- [15] S. Bahri and P. P. Budi, "Pengembangan sistem informasi berbasis web," vol. 8, no. 3, 2020.
- [16] S. Rahmah, "Perancangan sistem aplikasi data siswa menggunakan VB.Net dan MySQL," vol. 3, no. 1, 2018.