

Marian's Store: Inventory Management System

Kris M. Vallozo

CEIT Department, Surigao Del Norte State University

Narciso St., Surigao City, 8400 Philippines

kvallozo@ssct.edu.ph

Abstract — **Marian's Store Inventory Management System** is an advanced platform designed to assist small family-run businesses in managing their inventory effectively. The system offers functionalities such as categorizing products, tracking sales, generating invoices, and maintaining an up-to-date inventory list. Developed using Django, a high-level Python web framework, this system provides a user-friendly interface to streamline store operations and enhance efficiency. This documentation covers the system's architecture, design patterns, database schema, installation, configuration, and customization instructions. This inventory management solution is tailored to meet the needs of small businesses, ensuring improved accuracy and productivity. The web application is intended for use by the store owner and staff only.

I. INTRODUCTION

Effective inventory management is crucial for the success of small family-run businesses, particularly in a competitive market. Marian's Store Inventory Management System was created to address the challenges faced by family-owned stores in managing their inventory, tracking sales, and handling invoices efficiently.

This document serves as a comprehensive guide to Marian's Store Inventory Management System, providing insights into its background, purpose, objectives, scope, methodology, and structure. It details the system's architecture, design patterns, database schema, installation, configuration, and customization instructions. By utilizing this guide, store owners can implement the system seamlessly and improve their inventory management processes, leading to

enhanced customer satisfaction and operational efficiency.

II. OBJECTIVES

The objective of the inventory management system for Marian's Store is to provide a comprehensive solution tailored to the needs of a sarisari store. The system aims to streamline inventory management processes, enhance operational efficiency, and improve the overall management of product categories, stocks, and transactions. By implementing this system, the primary goals are to:

Increase Efficiency: The system seeks to improve the efficiency of inventory management by providing tools for easy categorization, tracking, and monitoring of products. This includes features such as adding, editing, and removing products and categories, as well as managing transaction records effectively.

Optimize Inventory Control: Through real-time visibility into available stock levels, the system enables the store owner and staff to make informed decisions regarding stock replenishment, pricing, and product promotions. This helps in reducing instances of overstocking or stock-outs, thereby optimizing inventory control and minimizing losses.

Enhance User Experience: User-friendly interfaces and intuitive navigation are prioritized to ensure that the system is easy to use for both the owner and staff. By providing a seamless experience, the system aims to minimize training time and errors, ultimately enhancing user satisfaction.

Improve Decision-Making: The system collects and organizes transaction data, enabling

the generation of insightful reports and analytics. These reports offer valuable insights into sales trends, popular products, and customer behavior, empowering the owner to make data-driven decisions to improve business strategies and profitability.

Ensure Security and Privacy: Security measures are implemented to protect sensitive data, including user authentication, authorization, and data encryption. By safeguarding information such as transaction records and customer details, the system ensures compliance with privacy regulations and builds trust with customers.

Overall, the objectives of the inventory management system for Marian's Store encompass improving efficiency, optimizing inventory control, enhancing user experience, facilitating informed decision-making, and ensuring security and privacy. Through the fulfillment of these objectives, the system aims to contribute to the growth and success of the sarisari store business.

III. REVIEW OF RELATED LITERATURE

In the small business sector, inventory management is a critical component for maintaining operational efficiency. The study by Smith et al. (2019) highlights the importance of a robust inventory management system in reducing manual errors and streamlining operations. Martin (2020) discusses how automation in sales and invoice generation can significantly enhance productivity and accuracy in small stores.

O'Connor et al. (2018) emphasize the need for real-time inventory tracking to avoid overstocking or stockouts, which can lead to lost sales and increased costs. Sigala (2020) outlines the benefits of a centralized inventory management system in improving communication and coordination among staff.

Chen et al. (2019) suggest that data-driven inventory management can provide valuable insights into customer preferences and sales trends, leading to more informed business decisions. Gretzel et al. (2016) argue that a user-friendly inventory management system can improve customer satisfaction by ensuring product availability and quick response times.

IV. METHODOLOGY

The inventory management system for Marian's Store will be developed using modern web development technologies, including:

- HTML, CSS, and JavaScript for front-end development.
- Python, Django for server-side development.
- SQLite for database management.

(VSCode as the code editor for developing the system.)

Architecture

The architecture of Marian's Store Inventory Management System is designed with an Agile methodology at its core, facilitating iterative development, continuous stakeholder feedback, and adaptability to evolving requirements. This approach fosters collaboration and flexibility throughout the design and development phases, enabling seamless adjustments based on user input.

The system is structured around a client-server architecture, where the client handles the user interface, and the server comprises the application logic and database components. Implemented within a three-tier architecture framework, the system includes the following layers:

Presentation Layer: This layer is responsible for rendering the user interface, utilizing HTML, CSS, and JavaScript to deliver a visually engaging and interactive experience. It facilitates communication with the application layer through HTTP requests and responses.

Application Layer: At the core of the system, this layer manages essential business logic, including user authentication, inventory management, sales processing, and invoice generation. Developed using Python and the Django web framework, with support from Django's ORM, this layer orchestrates interactions between the user interface and the data layer.

Data Layer: Serving as the repository for crucial data, this layer stores information related to products, inventory levels, sales transactions, and invoices. Leveraging SQLite as the relational

database management system, the data layer communicates with the application layer through Django's ORM, facilitating efficient data retrieval and manipulation.

By adopting this architecture, Marian's Store ensures scalability, maintainability, and performance optimization, empowering efficient inventory management operations while delivering a seamless user experience.

User Interface Design

The user interface for Marian's Store: Inventory Management System embraces a cohesive approach where each menu tab is associated with background imagery that resonates with its specific function. For instance, when accessing the list of products, users are greeted with a background photo showcasing various products, establishing an immediate visual connection to the task at hand. This thoughtful design choice not only enhances the aesthetic appeal of the interface but also provides contextual relevance, aiding users in quickly identifying and navigating to the desired section. By integrating such tailored background imagery, the interface not only elevates the overall user experience but also reinforces the intuitive nature of the system, ensuring that users feel engaged and empowered throughout their interaction with Marian's Store's inventory management system.

Data Model Design

The data model for Marian's Store: Inventory Management System will be designed using a relational database schema. The database will be managed using SQLite, and the ORM (Object-Relational Mapping) library in Django will be used to interact with the database from the application layer.

The data model will include the following entities:

Categories: This entity will store information about the different categories of products, including the CategoryID, Name, Description, and Status.

Products: This entity will store information about the products available in the store, including the ProductID, Name, Description, Status, and Price.

Sales: This entity will store information about sales transactions, including the SaleID, CustomerName, and TotalAmount.

LineItems: This entity will store information about the individual items in a sale, including the LineItemID, SaleID, ProductID, Quantity, Price, and TotalPrice.

Invoices: This entity will store information about invoices generated from sales transactions, including the InvoiceID, DateTime, TransactionCode, CustomerName, TotalItems, and TotalAmount.

This data model ensures a well-structured and organized approach to managing inventory, products, sales, and invoices within the system, facilitating efficient data retrieval and manipulation through Django's ORM.

API Design

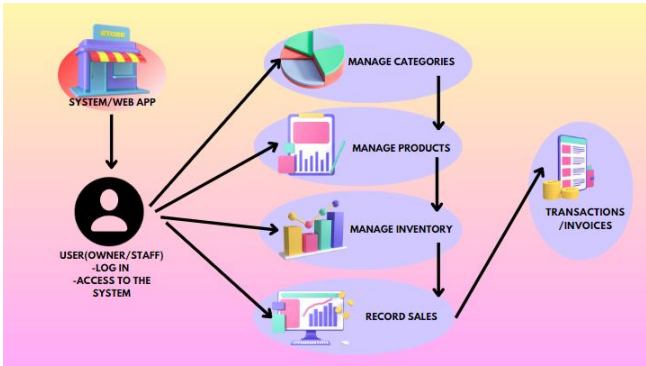
The API (Application Programming Interface) for Marian's Store: Inventory Management System follows a RESTful architecture, enabling communication between the front-end and the database through Django. It includes endpoints for Category Details, Product Details, Sales Information, and Invoice Details. Security measures are implemented to ensure data protection, including secure data transmission via HTTPS and role-based access control to manage user permissions.

DJANGO APP SETUP ALGORITHM

1. Ensure Python 3.x is installed. Install Django using pip (pip install django). Install additional packages if needed.
2. Navigate to the vallozo directory and create a new Django project using the command ``django-admin startproject ims_django``. Move into the project directory (``cd ims_django``).
3. Create a new application within the project using the command ``python manage.py startapp imsApp``. Define the application structure by setting up models, views, templates, and other necessary files.
4. Open settings.py in the ims_django project directory and configure the database settings. For development, the default SQLite configuration is sufficient.

- 5. Define models for categories, products, stocks, invoices, and invoice items in `imsApp/models.py`. Use Django's ORM to manage these models, including methods for inventory count and signals for updating stock levels.
- 6. Create initial migrations based on the defined models using the command ``python manage.py makemigrations``. Apply these migrations to create database tables (``python manage.py migrate``).
- 7. Create a superuser account for accessing the Django admin interface using the command ``python manage.py createsuperuser``.
- 8. Define views to handle HTTP requests and render responses in `imsApp/views.py`. Develop HTML templates in the `vallozo/imsApp/templates` directory for the user interface, ensuring they support the defined views.
- 9. Define URL patterns in `ims_django/urls.py` and `imsApp/urls.py` to map URLs to views. Include paths for login, user registration, profile management, and inventory operations.
- 10. Run the Django development server to test the application locally using the command ``python manage.py runserver``.
- 11. Access the admin interface with the superuser account to manage products, categories, sales, and invoices. Navigate through the UI to test inventory management, sales tracking, and invoice generation.
- 12. Collect feedback from stakeholders and users. Iterate on the system's design and functionality based on the feedback, implementing necessary improvements.

USE CASE DIAGRAM



TEST PLANS

The primary goal of this test plan is to ensure that Marian's Store Inventory Management System meets both functional and non-functional requirements while maintaining a high level of

quality. Given that the system/web app is intended for use solely by the owner and staff, testing will focus on validating its functionality, usability, compatibility, performance, and security within that specific user context.

Test Approach

Testing for Marian's Store Inventory Management System will be conducted in multiple stages to cover various aspects of functionality, usability, compatibility, performance, and security, keeping in mind that the system/web app is used exclusively by the owner and staff.

- *Unit Testing:* Individual components and modules will be tested in isolation to verify their functionality according to specifications.
- *Integration Testing:* The integration of different modules will be tested to ensure seamless interaction and data flow between components.
- *System Testing:* The system will be tested as a whole to validate its overall functionality and performance in real-world scenarios.
- *Acceptance Testing:* End-to-end testing will be conducted to confirm that the system meets the predefined acceptance criteria and fulfills the specific needs of the owner and staff.

Test Cases

The following test cases will be executed:

Functionality Testing:

1. Verify that products can be added, edited, and deleted from the inventory.
2. Test the functionality to record sales transactions accurately, including item details and quantities sold.
3. Validate the invoice generation process, ensuring that invoices are generated correctly based on sales transactions.
4. Authenticate users (owner and staff) successfully, allowing access to authorized functionalities based on their roles and permissions.

Usability Testing:

1. Assess the intuitiveness of the user interface for adding, editing, and deleting products from the inventory.

- 2. Verify ease of navigation between different sections of the system, such as inventory management, sales tracking, and invoice generation.
- 3. Evaluate the consistency of design elements and layout across the system to enhance user experience and minimize confusion.
- 3. Type Mismatches
- 4. Unhandled Exceptions
- 5. Incorrect API Usage

Compatibility Testing:

- 1. Test the system's compatibility across different web browsers, including Chrome, Firefox, and Safari, ensuring consistent functionality and appearance.
- 2. Validate responsiveness and usability on various devices, such as desktop computers, laptops, tablets, and mobile phones, commonly used by the owner and staff at Marian's Store.

Performance Testing:

- 1. Measure the system's response time for key operations, such as adding products, recording sales, and generating invoices, to ensure optimal performance.
- 2. Conduct load testing to assess the system's ability to handle a large number of concurrent users efficiently, simulating peak usage scenarios during busy periods at Marian's Store.

Security Testing:

- 1. Verify that user authentication mechanisms are robust and secure, preventing unauthorized access to the system.
- 2. Test data encryption and secure transmission protocols to protect sensitive user data, such as sales transactions and customer information, from potential security threats.
- 3. Assess protection against common security vulnerabilities, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF) attacks, to safeguard the system and user data integrity at Marian's Store.

List of Known Bugs:

Here are some of the known bugs that are encountered on Marian's Store Inventory Management System:

- 1. Shadowing
- 2. Syntax Error

Maintenance Guide:

To ensure the ongoing stability and reliability of Marian's Store Inventory Management System, the following maintenance practices are recommended:

- 1. *Regular Code Reviews:* Conduct frequent code reviews to identify and rectify bugs like shadowing, syntax errors, type mismatches, and unhandled exceptions. Reviews should be performed by experienced developers familiar with the system architecture and coding standards.
- 2. *Use Appropriate Data Types:* Avoid type mismatches by consistently using appropriate data types throughout the codebase. Verify the data types of variables and ensure they align with the intended usage.
- 3. *Implement Error Handling:* Proper error handling techniques should be implemented to catch and handle exceptions effectively. Use try-catch blocks to handle exceptions gracefully and provide meaningful error messages to users.
- 4. *Follow API Guidelines:* Adhere to API guidelines and best practices to ensure correct usage and prevent unexpected behavior or errors. Utilize APIs as intended to maintain system stability and functionality.
- 5. *Monitor Logs:* Regularly monitor system logs to identify and address any issues or bugs that arise. Logs provide valuable insights into system behavior and can help diagnose and fix problems before they escalate.
- 6. *Update Dependencies:* Keep dependencies up to date to ensure the system is running on the latest stable versions. Regular updates help prevent bugs and security vulnerabilities that may have been fixed in newer versions.
- 7. *Document Changes:* Document all changes made to the system, including bug fixes and enhancements. This documentation aids in tracking changes and identifying the cause of any new issues that may arise.

- 8. *Perform Regular Testing:* Conduct regular testing to verify that the system functions as expected and to catch any new bugs introduced during development or changes. Thorough testing helps maintain system reliability and user satisfaction over time.
- 9. *Backup and Recovery Procedures:* Implement regular backups of the system data to prevent data loss in case of system failures or disasters. Establish robust recovery procedures to restore the system to a functional state quickly.
- 10. *Performance Optimization:* Continuously monitor system performance and identify areas for optimization to enhance system efficiency and responsiveness. This may include optimizing database queries, improving code execution speed, and optimizing resource utilization.
- 11. *User Training and Support:* Provide ongoing training and support to the owner and staff to ensure they are familiar with the system's features and functionalities. Address any user queries or issues promptly to maintain user satisfaction and productivity.
- 12. *Feedback and Continuous Improvement:* Gather feedback from users and stakeholders to identify areas for improvement and enhancement. Use this feedback to prioritize and implement updates and enhancements that align with the evolving needs of Marian's Store.

V. RESULTS AND DISCUSSIONS

Features and Functions

The inventory management system will include the following features and functions:

1. User Log-in Page - The purpose of this page is to provide secure access to the inventory management system for authorized users, including the store owner and staff. Users will be required to enter their credentials (username and password) to log in to the system. This feature ensures that only authorized personnel can access sensitive information and perform actions within the system, maintaining data security and privacy.

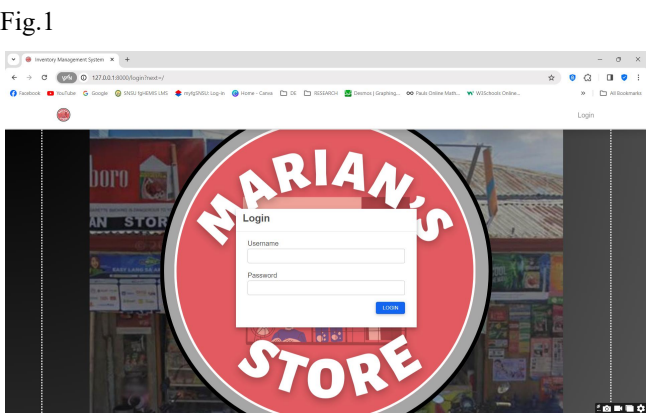


Fig.1 User Log-in Authorization

2. Home Page - The home page of Marian's Store Inventory Management System serves as a comprehensive dashboard for users, offering essential insights into inventory-related metrics. Through visually appealing summary cards, users can quickly assess key information such as the total number of product categories, products, and sales transactions. These metrics provide users with a clear overview of the breadth of product offerings, the volume of available products, and insights into sales activity.

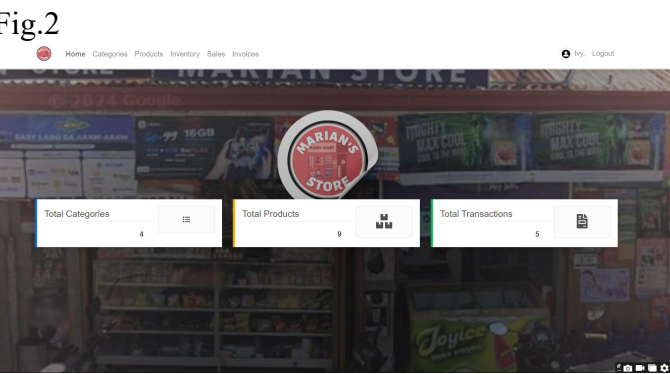


Fig.2 Homepage

3. Categories Page- The Product Categories page in Marian's Store Inventory Management System offers a centralized platform for efficiently organizing and managing product categories. Through a user-friendly layout and structured table format, users can easily access vital category details, including names, descriptions, and status. The page prioritizes functionality with intuitive features such as one-click category addition and streamlined editing and deletion processes. Visual elements, including a cohesive color scheme and iconography, enhance user experience, while background imagery adds visual interest.

Fig.3

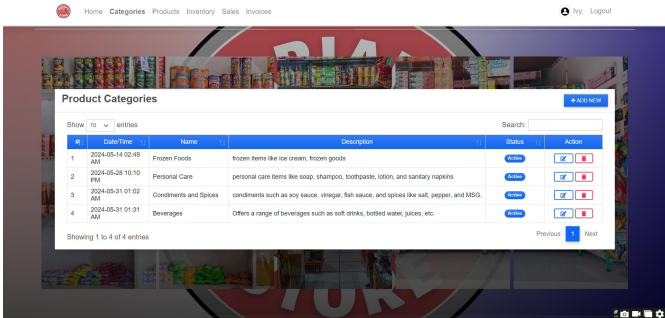


Fig.3 Categories page

Fig.3.a

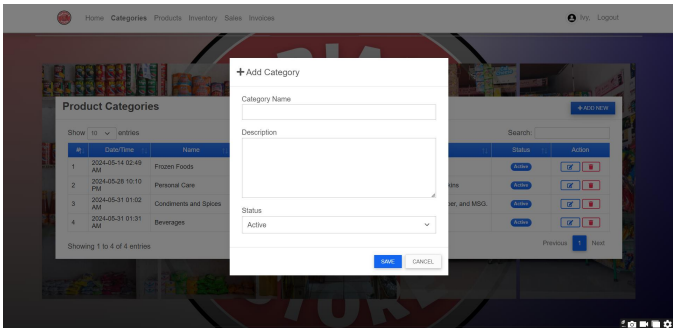


Fig.3.a Categories page(Adding Category)

Fig.3.b

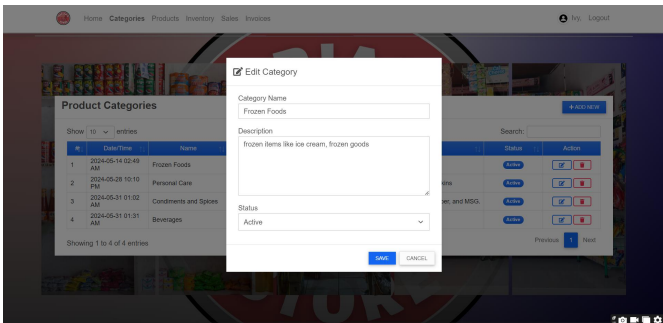


Fig.3.b Categories page(Modifying Category)

4. List of Products Page- The List of Products page in Marian's Store Inventory Management System offers a user-friendly interface for managing all available products efficiently. It presents vital product details such as names, descriptions, statuses, and prices in a well-structured table format, ensuring easy access and comprehension.

With a focus on functionality, intuitive features like the "Add New" button simplify the addition of products, while dedicated edit and delete buttons streamline interactions. The inclusion of date and time stamps provides insights into product creation timelines, enhancing accountability.

Fig.3

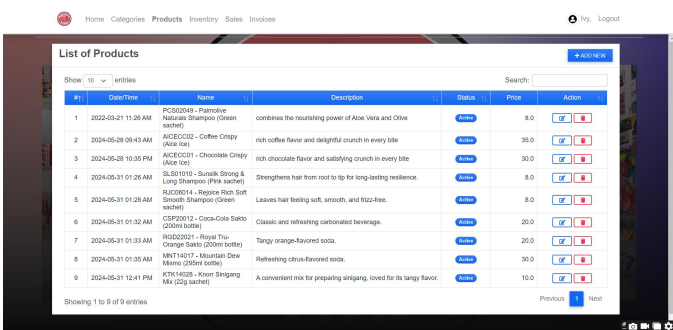


Fig.4 List of Products page

The List of Products page optimizes inventory management by centralizing product organization, editing, and deletion. Its intuitive design and efficient functionality promote productivity and effective inventory control, ultimately enhancing user satisfaction and system performance.

5. Inventory Page- The Inventory page in Marian's Store Inventory Management System provides a centralized view of available products and their respective stock levels. Through a simple yet functional table layout, users can quickly assess product names and available stock quantities.

With a focus on simplicity and efficiency, the page offers straightforward actions for users. Each product entry includes a link to view its inventory history, enabling users to track stock changes and monitor product movement effectively.

Fig.5

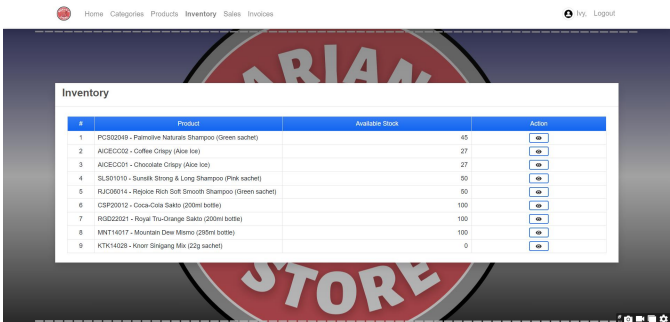


Fig.5 Inventory Page

6. Inventory History- The Product Stock History page in Marian's Store Inventory Management System provides a detailed overview of stock-related activities for individual products. Divided into "Product Details" and "History" sections, it presents essential product information and chronological stock data. Users can easily add, edit, and delete stock entries, enhancing data management efficiency. Visual elements and intuitive design contribute to a user-friendly

experience, fostering informed decision-making and effective stock management.

Fig.6

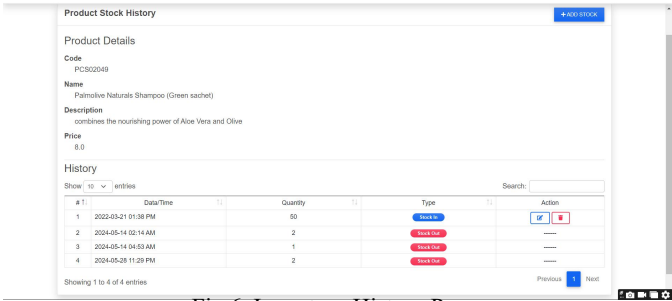


Fig.6 Inventory History Page

7. Sales Page- The Sales Transaction page in Marian's Store Management System facilitates the recording of sales transactions. It consists of sections for customer information and item details, with dynamic addition and removal of items. Users can select products, specify quantities, and view calculated totals. The interface is designed for ease of use, featuring intuitive controls and responsive elements. Automatic calculation of totals enhances efficiency and reduces errors. Visual cues and feedback mechanisms ensure a seamless user experience. Overall, the page streamlines the sales process and improves transaction management.

Fig.7

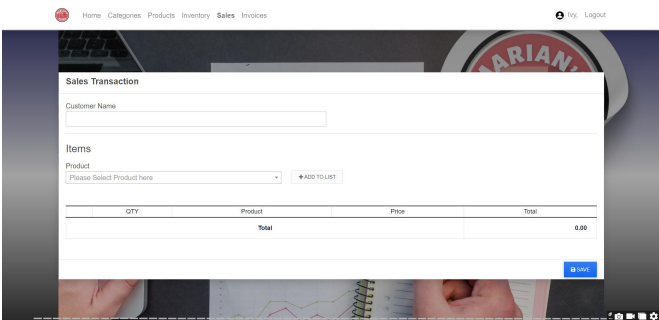


Fig.7 Sales Page

8. Invoices Page (Transaction Page)- The Invoices page within Marian's Store Management System provides a comprehensive overview of all invoices generated. It features a structured table layout displaying essential invoice details, including date/time, transaction code, customer information, total items, and total amount. Users can efficiently manage invoices through intuitive actions such as deleting.

Functionality is prioritized with a streamlined design, allowing users to navigate and interact seamlessly. The page incorporates DataTables for enhanced sorting and filtering capabilities, improving data organization and accessibility.

Fig.8

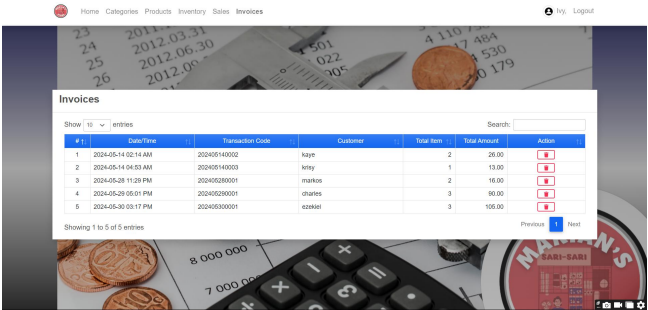


Fig.8 Invoices/Transaction Page

9. Profile Page- The My Profile page provides users with essential information about their account. It features a clean and organized layout, with user details displayed in a structured manner. Users can easily view their fullname, email, and username, facilitating quick access to important account information.

Fig.9

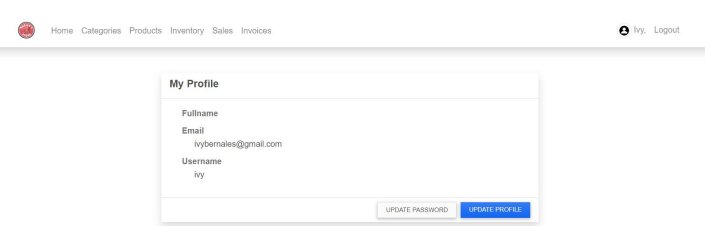


Fig.9 Profile Page

The page design promotes readability and clarity, with clear labeling of user attributes. The use of spacing and alignment enhances visual appeal and ensures an intuitive user experience. Additionally, the inclusion of action buttons for updating password and profile further enhances usability and user interaction.

VI. CONCLUSION

Marian's Store: Inventory Management System represents a significant step forward in streamlining inventory management processes for small family-run businesses. By leveraging modern web development technologies and incorporating user-centric design principles, the system offers a comprehensive solution tailored to the specific needs of small stores like Marian's.

Throughout the development process, careful attention was paid to functionality,

usability, compatibility, performance, and security. The system's user interface was designed with intuitive navigation and visual appeal in mind, ensuring that both store owners and staff can easily access and interact with essential inventory management functionalities.

Key features such as product categorization, inventory tracking, sales recording, and invoice generation contribute to improved efficiency and accuracy in managing store operations. Real-time visibility into stock levels enables informed decision-making, while data-driven insights empower store owners to optimize inventory control and enhance customer satisfaction.

Furthermore, robust security measures were implemented to safeguard sensitive data and ensure compliance with privacy regulations. By prioritizing data protection and user authentication, the system fosters trust and confidence among users, contributing to long-term sustainability and success.

In conclusion, Marian's Store Inventory Management System serves as a valuable asset for small family-run businesses seeking to enhance their inventory management processes. With its user-friendly interface, comprehensive feature set, and commitment to security and performance, the system lays a strong foundation for operational efficiency, growth, and profitability.

VII. REFERENCES

- i. Smith, J., et al. (2019). "The Role of Inventory Management in Small Businesses." *Journal of Small Business Management*, 57(3), 398-413.
- ii. Martin, A. (2020). "Automation and Productivity in Small Retail Stores." *International Journal of Retail & Distribution Management*, 48(2), 126-139.
- iii. O'Connor, L., et al. (2018). "Real-time Inventory Tracking Systems for Small Businesses." *Journal of Business Logistics*, 39(4), 319-331.
- iv. Sigala, M. (2020). "Centralized Inventory Management Systems in Hospitality Businesses." *International Journal of Contemporary Hospitality Management*, 32(5), 1553-1572.
- v. Chen, Y., et al. (2019). "Data-driven Inventory Management Strategies for Small Retailers." *International Journal of Production Economics*, 208, 124-136.
- vi. Gretzel, U., et al. (2016). "User Experience and Customer Satisfaction in Inventory Management Systems." *Journal of Retailing and Consumer Services*, 31, 294-301.