SPS Sankul Collage – Atkot

A Project Report On

Inventory Management System

Acedemic Year  
2024  
  
Developed By

Kachhadiya Dhruvika  
Khokhani Kashish  
  
Internal Guided By

Respected Chandanimem vyas

Submitted To



Shree Saurashtra University - Atkot

**Shree Saurashtra College of MGT. & Computer Science**

**(College Code : 73001)**

**Opp. Bhoomi Ginning, Atkot – Jasdan Highway, Atkot – 360040**

**Certificate**

**This is to certify that Mr. / Ms.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**has satisfactorily completed his/her academic Project work on**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**as a part of the BCA Semester – 5 Project work prescribed by the Saurashtra University for the academic year 2024-25.**

Project ID :- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Exam Seat No. :- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class :- **BCA Sem - 5 .**

Enrollment No. :- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project guide :- Principal :-**Chandanimem vyas

**P R E F A C E**

Understanding the concept of a "project" can indeed be crucial for effectively managing and executing it. Breaking down the word "project" into its seven letters, each with its own significance, might look something like this:

P – Purpose: Every project has a clear objective or goal it aims to achieve.

R – Resources: Projects require various resources including time, money, and human effort.

O – Organization: A project needs a structured plan and organization to manage its tasks and processes.

J – Journey: The project is a process with distinct stages or phases from start to finish.

E – Evaluation: Regular assessment of progress and performance is essential to ensure the project stays on track.

C – Collaboration: Successful projects often depend on effective teamwork and communication among members.

T – Timeframe: Projects are typically bound by a schedule or deadline for completion.

In today’s dynamic business environment, effective inventory management is crucial for maintaining operational efficiency and meeting customer demands. The primary goal of this project is to design and implement an Inventory Management System (IMS) that enhances the accuracy, speed, and efficiency of inventory tracking and control.

An Inventory Management System is designed to streamline the process of managing inventory levels, orders, sales, and deliveries. By integrating real-time data and providing insightful analytics, the system aims to minimize stockouts and overstock situations, optimize inventory levels, and improve overall supply chain management.

This project will involve analyzing current inventory management practices, identifying key requirements, and developing a solution that addresses these needs. The system will be tailored to fit specific business processes, ensuring it delivers a user-friendly interface and robust functionalities.

Key objectives of this project include:

System Design and Development: Creating a comprehensive system architecture that supports various inventory management functions such as tracking, reporting, and forecasting.

User Interface: Designing an intuitive and accessible user interface that facilitates ease of use for all stakeholders involved.

Integration: Ensuring seamless integration with existing business processes and technology stacks.

Testing and Deployment: Conducting thorough testing to validate system performance and reliability before full deployment.

Training and Support: Providing adequate training for end-users and ongoing support to ensure smooth operation.

**ACKNOWLEDGEMENT**

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this project. Special thanks are to my mentor Chandanimem vyas whose help, stimulating suggestions and encouragement helped me in all time of fabrication process and in writing this report. I also sincerely thanks for the time spent proofreading and correcting my many mistakes.

Many thanks go to the all lecturers who have given their full effort in guiding me in achieving the goal as well as their encouragement to maintain our progress in track. My profound thanks go to all classmates, especially to my friends for spending their time in helping and giving support whenever I need it in fabricating my project.

Finally, We are thankful to all the individuals who names are not included here, but who have rendered their co-operation, little or more, directly or indirectly in the course of development of this system & preparation of this document.

THANKS ALL…

**Project Index**

|  |  |
| --- | --- |
| **Sr. No.** | **Description** |
| **1** | **INTRODUCTION** |
|  | 1.1 EXISTING SYSTEM |
|  | 1.2 LIMITATION OF EXISTING SYSTEM |
| **2** | **PROPOSED SYSTEM** |
|  | 2.1 PROJECT PROFILE |
|  | 2.2 INTRODUCTION |
|  | 2.3 BASIC FUNCTIONALITY OR SCOPE |
|  | 2.4 OBJECTIVE |
| **3** | **INTRODUCTION TO DEVELOPMENT ENVIRONMENT** |
|  | 3.1 WHAT IS PHP? |
|  | 3.2 MYSQL |
| **4** | **SYSTEM PLANNING** |
|  | 4.1 FEASIBILITY STUDY |
|  | 4.2 PROCESS MODEL |
| **5** | **SYSTEM REQUIREMENT SPECIFICATION** |
|  | 5.1 SYSTEM REQUIREMENT |
|  | 5.2 MODULE SPECIFICATION |
| **6** | **DETAIL PLANNING** |
|  | 6.1 DATA DICTIONARY |
|  | 6.2 ENTITY – RELATIONSHIP DIAGRAM |
| **7** | **SYSTEM DESIGN** |
|  | 7.1 INPUT DESIGN |
|  | 7.2 OUTPUT DESIGN |
|  | 7.3 SCREENSHOTS OF THE SYSTEM |
| **8** | **SOFTWARE TESTING** |
| **9** | **LIMITATIONS AND FUTURE SCOPE OF ENHANCEMENT** |
| **10** | **REFERENCE** |

**1. Introduction**

1.1 Existing System

1.2 Limitation of Existing System

The Inventory Management System (IMS) is a robust and adaptable solution that empowers businesses in various sectors, including supermarkets and medical stores, to efficiently manage their inventory, sales, and purchases. Developed using a combination of PHP, MySQL, HTML, CSS, Bootstrap, JavaScript, Ajax, and jQuery, this system caters to two distinct user roles: Users and Admins.

For Users, IMS offers a seamless experience, enabling them to register and log in securely, manage purchases, process sales, handle returns, generate bills, and conveniently select products with detailed information added by Admins.

Admins, on the other hand, wield comprehensive control over the system. They can manage units, maintain purchase party information, oversee product details, track stock levels, monitor product expiry dates, access inventory information, generate purchase and sales reports, view party-specific reports, and manage returns, among other functionalities.

With a robust MySQL database underpinning the system and a technology stack that combines PHP, Bootstrap, JavaScript, and more, this Inventory Management System streamlines operations, enhances inventory control, and provides valuable insights into business transactions. This documentation serves as a guide to understanding and utilizing this dynamic and versatile system to its fullest potential.

**1.1 EXISTING SYSTEM**

**1. Authentication**

No centralized authentication. Access is managed through informal methods.

**2. Unit Management**

Units of measurement are tracked manually in spreadsheets or physical records.

**3. Party Information Management**

Purchase party details are maintained in paper files or basic spreadsheets, making updates and retrieval slow.

**4. Product Information Management**

Product details are managed manually or in basic databases, with frequent errors and inefficiencies.

**5. Purchase Record Management**

Records are manually maintained in ledgers or spreadsheets, leading to delays and inaccuracies.

**6. Sales and Billing**

Sales are processed manually with billing done using physical invoices or spreadsheets, resulting in manual stock updates.

**7. Expiry Date Tracking**

Expiry dates are tracked manually with no automated alerts for nearing expiry.

**8. Stock Information**

Stock levels are updated manually, with reports compiled manually, leading to potential inaccuracies and delays.

**9. Reports**

Reports are generated manually or via basic spreadsheet functions, lacking real-time accuracy.

**10. Return Management**

Returns are handled manually, with stock adjustments made in physical records or spreadsheets.

**11. Bill Generation**

Bills are manually generated with no automated system to update stock or financial records.

**Challenges:**

Inefficiencies, risk of errors, scalability issues, and fragmented processes impact overall operational effectiveness.

**1.2 LIMITATION OF EXISTING SYSTEM**

**Manual Processes:**

Error-Prone: Reliance on manual data entry increases the likelihood of errors.

Time-Consuming: Manual updates and processes are slow and labor-intensive.

**Lack of Real-Time Data:**

Delayed Updates: Inventory levels and transaction records are not updated in real-time, leading to outdated information.

**Inaccurate Reporting:**

Reports are often based on outdated or inaccurate data due to the delay in manual updates.

**Limited Scalability:**

Inefficient Handling of Growth: As inventory volume and transaction frequency increase, the manual system becomes increasingly cumbersome and difficult to manage.

**Integration Issues:**

Fragmented Processes: Different aspects of inventory management (e.g., stock tracking, sales, billing) operate in isolation, leading to inefficiencies and potential inconsistencies.

**Reporting Challenges:**

Manual Report Generation: Reports are created manually or through basic tools, making them time-consuming and prone to errors.

**Lack of Customization:**

Reporting capabilities are limited, with little flexibility for customized or real-time reports.

**Limited Tracking and Alerts:**

Expiry Tracking: Manual tracking of expiry dates lacks automated alerts, risking potential spoilage and loss.

**Stock Alerts:**

There are no automated notifications for low stock levels or other critical inventory issues.

**Security and Access Control:**

Inadequate Security: Lack of a centralized authentication system means limited control over user access and potential security risks.

**Return Management:**

Inefficient Processing: Handling returns is manual and can be prone to errors, affecting inventory accuracy and stock levels.

**2. Proposed System**

2.1 Project Profile

2.2 Introduction

2.3 Basic functionality or scope

2.4 Objective

**2.1 PROJECT PROFILE**

|  |  |
| --- | --- |
| **Project Name** | **Inventory Management System** |
| **Technology** | **PHP, MySQL** |
| **Front End** | **HTML , CSS , JS** **,** **Ajax** |
| **Additional Libraries** | **jQuery , Bootstrap** |
| **Back End** | **MySQL** |
| **Develop Environment** | **Visual Studio code** |
| **Web Browser** | **Chrom or any other** |
| **Web Server** | **XAMPP** |
| **Operating System** | **Windows** |
| **Guided by** | **Chandanimem vyas** |
| **Developed By** | **Kachhadiya Dhruvika**  **Khokhani Kashish** |

**2.2 INTRODUCTION**

An Inventory Management System (IMS) is a tool used by businesses to track and manage their inventory levels, orders, sales, and deliveries. The primary goal of an IMS is to ensure that a company has the right amount of stock at the right time to meet customer demand while minimizing excess inventory and associated costs. Here’s a brief overview:

**Key Components:**

**Inventory Tracking:** Monitors stock levels in real-time, helping businesses keep an accurate count of their inventory.

**Order Management:** Handles purchase orders, sales orders, and inventory replenishment, ensuring that stock levels are maintained.

**Stock Valuation:** Calculates the value of inventory based on various methods such as FIFO (First In, First Out), LIFO (Last In, First Out), or average cost.

**Reporting:** Generates reports on stock levels, turnover rates, and other key metrics to support decision-making.

**Integration:** Often integrates with other systems like accounting, sales, and supply chain management to streamline operations.

**Automation:** Automates routine tasks such as reordering stock, updating inventory levels, and generating alerts for low stock.

**Benefits:**

Efficiency: Reduces manual tracking and paperwork, leading to faster and more accurate inventory management.

**Cost Savings:** Helps in reducing overstock and stockouts, minimizing carrying costs and lost sales.

**Improved Accuracy:** Provides real-time data to reduce errors and discrepancies in inventory records.

**Better Forecasting:** Enhances the ability to predict demand and plan inventory purchases more effectively.

**Use Cases:**

**Retail:** Manages stock across multiple locations, tracks sales patterns, and optimizes reordering.

**Manufacturing:** Monitors raw materials and finished goods, coordinating supply with production schedules.

**E-commerce:** Synchronizes online orders with inventory levels, ensuring availability and timely fulfillment.

In essence, an Inventory Management System helps businesses maintain optimal inventory levels, improve operational efficiency, and make informed decisions based on accurate data.

**2.3 BASIC FUNCTIONALITY OR SCOPE**

**Scope**

* The scope of the proposed Inventory Management System (IMS) is extensive, designed to cater to the dynamic needs of various businesses, including supermarkets and medical stores. Its versatility makes it a valuable tool in managing inventory, sales, and purchases efficiently. Below, we outline the scope of the system, focusing on the aspects related to the scope without delving into functionality details:
* **Adaptability:** The IMS is adaptable to a wide range of business environments, including supermarkets and medical stores. It can be customized to suit the specific needs of any industry that requires inventory management.
* **Dynamic Nature:** The system is inherently dynamic, accommodating changes in product offerings, units of measurement, purchase parties, and product details seamlessly.
* **User Roles:** The IMS caters to two main user roles: "user" and "admin." Users can engage in purchase, sales, and billing operations, while administrators have access to comprehensive management and reporting capabilities.
* **Inventory Control:** The system enables efficient control over inventory by allowing users to add, edit, and delete items. It also handles stock decreases during sales and manages returns.
* **Reporting:** Detailed reporting functionalities are available to administrators, offering insights into purchase history, sales trends, stock levels, and more. Reports can be generated for specific date ranges, facilitating informed decision-making.
* **Data Security:** User data and sensitive business information are secured through user authentication and access control measures, ensuring data privacy and integrity.
* **Efficiency:** The IMS streamlines operations, reducing manual effort in inventory management, sales, and billing. This efficiency leads to cost savings and improved productivity.
* **Flexibility:** The system accommodates various units of measurement (e.g., KG, LITRE, GRAMS, PIECES) and supports multiple purchase party details, making it versatile for different businesses.
* **Expiry Date Tracking:** Administrators can track product expiry dates, helping them identify and manage products nearing expiration.
* **User-Friendly:** The system's user interface is designed to be user-friendly, facilitating ease of use for both users and administrators.

**2.4 OBJECTIVE**

**1. Authentication Module:**

* **Objective**: Ensure secure access to the system.
* **Functionalities**:
  + Admin and users can register and log in.
  + Authentication and authorization mechanisms to control access.
  + Admin can log out of the system when the session is complete.

**2. Unit Management Module:**

* **Objective**: Manage units of measurement for products.
* **Functionalities**:
  + Admin can add, edit, and delete units (e.g., KG, LITRE, GRAMS, PIECES).

**3. Party Information Module:**

* **Objective**: Manage information about purchase parties.
* **Functionalities**:
  + Admin can add, edit, and delete party information (e.g., firstname, lastname, contact, address, city).

**4. Product Information Module:**

* **Objective**: Manage product details.
* **Functionalities**:
  + Admin can add, edit, and delete product information.
  + Include details like product name, amount, quantity, purchase party name, purchase type (cash, debit), expiry date, etc.

**5. Purchase Record Module:**

* **Objective**: Keep track of purchases and update stock.
* **Functionalities**:
  + Admin can add, edit, and delete purchase records.
  + Stock is automatically increased upon purchase.

**6. Sales and Billing Module:**

* **Objective**: Manage sales and generate bills.
* **Functionalities**:
  + Admin can sell items and generate bills.
  + Stock is automatically decreased upon sale.

**7. Expiry Date Tracking Module:**

* **Objective**: Monitor products nearing expiration.
* **Functionalities**:
  + Admin can add expiry dates for products.
  + View products with the nearest expiry dates.

**8. Stock Information Module:**

* **Objective**: Provide real-time information on stock.
* **Functionalities**:
  + Admin can view the current stock of items.

**9. Reports Module:**

* **Objective**: Generate various types of reports.
* **Functionalities**:
  + Admin can generate purchase reports between two dates.
  + Admin can generate sales reports between two dates.
  + Admin can view the purchase report.
  + Admin can view the report of any specific party.
  + Admin can view day-to-day selling reports.
  + Admin can print stock reports.
  + Admin can view nearest expiry product lists.

**10. Return Management Module:**

* **Objective**: Handle product returns.
* **Functionalities**:
  + Admin can process returns for items.

**11. User Management Module (Admin Only):**

* **Objective**: Admin can manage user accounts.
* **Functionalities**:
  + Admin can add, edit, and delete user accounts.
  + User can log out of the system when the session is complete

**12. Edit Stock Module (Admin Only):**

* **Objective**: Admin can manually edit stock items.
* **Functionalities**:
  + Admin can edit stock items.

**3. INTRODUCTION TO DEVELOPMENT ENVIRONMENT**

3.1 WHAT IS PHP?

3.2 MYSQL

**3.1 WHAT IS PHP?**

PHP (Hypertext Preprocessor) is a server-side scripting language used for web development. It is a mature, open-source language that is widely used for creating dynamic and interactive web applications.

**History of PHP:**

PHP was first released in 1995 by Rasmus Lerdorf, a Danish-Canadian programmer. Initially, it was called "Personal Home Page Tools" and was used for creating simple web applications. Over the years, PHP has evolved to become a powerful and feature-rich language, with a large community of developers contributing to its growth..

**Features of PHP:**

* **Server-side scripting**: PHP code is executed on the server, allowing for dynamic content generation and interaction with databases.
* **Loose typing**: PHP is a dynamically-typed language, which means that you don't need to declare variable types before using them.
* **Object-oriented**: PHP supports object-oriented programming (OOP) concepts, such as classes, objects, inheritance, and polymorphism.
* **Large community**: PHP has a massive community of developers, which means there are many resources available for learning and troubleshooting.
* **Cross-platform**: PHP can run on various operating systems, including Windows, macOS, and Linux.

**3.2 MYSQL**

MySQL is a popular, open-source relational database management system (RDBMS) used for storing and managing data. It is a key component of the LAMP (Linux, Apache, MySQL, PHP) stack, which is widely used for web development.

**History of MySQL:**

MySQL was first released in 1995 by MySQL AB, a Swedish company founded by David Axmark, Allan Larsson, and Michael Widenius. In 2008, Sun Microsystems acquired MySQL AB, and in 2010, Oracle Corporation acquired Sun Microsystems, becoming the owner of MySQL.

**Features of MySQL:**

* **Relational database:** MySQL is a relational database, which means it stores data in tables with well-defined relationships between them.
* **SQL support:** MySQL supports Structured Query Language (SQL), which is used to manage and manipulate data.
* **ACID compliance**: MySQL is ACID (Atomicity, Consistency, Isolation, Durability) compliant, ensuring that database transactions are processed reliably.
* **Scalability:** MySQL is designed to handle large amounts of data and scale horizontally, making it suitable for high-traffic web applications.
* **Security:** MySQL has robust security features, including encryption, access control, and authentication.

**4. SYSTEM PLANNING**

4.1 Feasibility study

4.2 Process Model

**4.1 Feasibility study**

In the initial phases of developing the Inventory Management System (IMS), a comprehensive feasibility study was conducted to evaluate the practicality and viability of the project. This study examined various aspects to ensure that the IMS would meet the requirements and expectations of users and administrators.

The feasibility study encompassed the following key components:

**4.1.1 Technical Feasibility:**

* **Hardware Compatibility:** We assessed the compatibility of the chosen hardware, including processor, RAM, storage, and display, to ensure that it could effectively run the IMS software stack.
* **Software Compatibility:** We verified that the selected software components, such as the operating system, web server (Apache), database server (MySQL), and development environment (Notepad, Visual Studio Code), were compatible with one another.
* **Web Browser Compatibility:** We tested the system on popular web browsers (Chrome, Firefox, Safari) to ensure a consistent user experience across different platforms.

**4.1.2 Economic Feasibility:**

* **Cost Analysis:** We conducted a cost analysis to estimate the expenses associated with the development, deployment, and maintenance of the IMS. This included costs for hardware, software licenses (if any), and ongoing operational expenses.
* **Return on Investment (ROI):** We calculated the potential ROI by assessing the benefits of improved inventory management, reduced errors, and increased efficiency against the projected costs.

**4.1.3 Operational Feasibility:**

* **User Requirements:** We gathered user requirements through interviews and surveys to understand the specific needs and workflows of both users and administrators.
* **Workflow Analysis:** We analyzed the existing inventory management workflows within different business environments, such as supermarkets and medical stores, to ensure that the IMS could adapt seamlessly to these scenarios.
* **User Training:** We assessed the feasibility of providing training and support to users and administrators to ensure efficient system adoption.

The results of the feasibility study conclusively demonstrated that the Inventory Management System was both technically and economically viable. It was found to be compatible with a wide range of hardware and software configurations, promised a positive ROI through enhanced inventory management, and was operationally feasible due to user-friendly workflows. Legal and regulatory compliance measures were also taken into account, ensuring the system's development in accordance with industry standards.

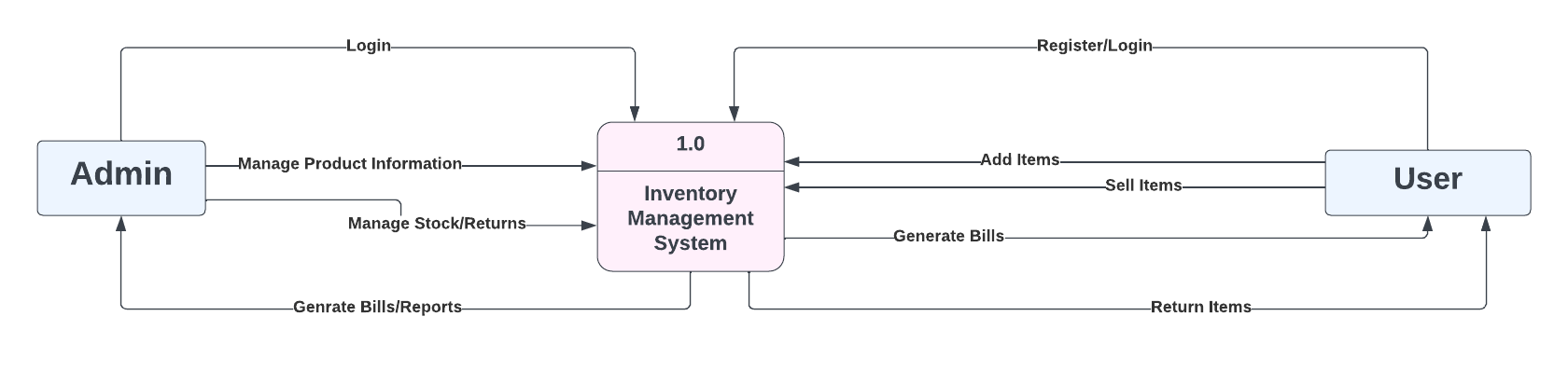
With the green light from the feasibility study, the project proceeded to the design and development phases, reinforcing the system's adaptability and versatility for use in diverse business environments.

**4.2 PROCESS MODEL**

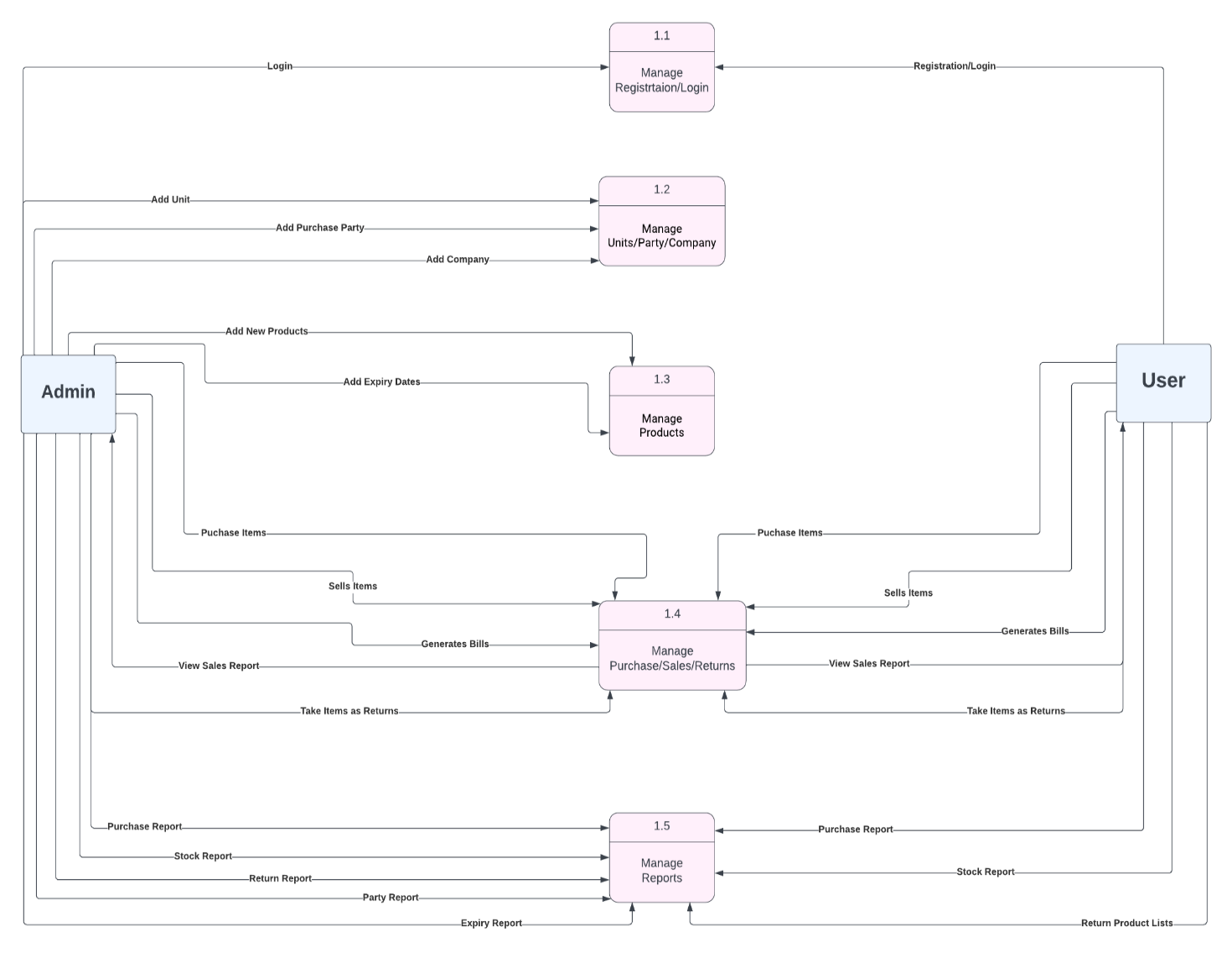
DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

* Logical information flow of the system
* Determination of physical system construction requirements
* Simplicity of notation
* Establishment of manual and automated systems requirements

**Zero Level DFD**

****

**First Level DFD**

****

**5. SYSTEM**

5.1 System Requirement

5.2 Module Specification

**5.1 SYSTEM REQUIREMENT**

|  |  |
| --- | --- |
| **Software Component** | **Version** |
| Operating System | Windows |
| Web Server | Apache (XAMPP recommended) |
| Database Server | MySQL |
| Development Environment | Visual Studio Code |
| Web Browser | Chrome |
| Front-end Design | HTML,CSS,JS |
| Additional Libraries | jQuery, Bootstrap, Ajax |

**5.2 MODULE SPECIFICATION**

* **Admin Functionalities:**

1. **Authentication:**
   * Admin can log in to the system with a username and password.
   * Admin can log out of the system when the session is complete.
2. **Unit Management:**
   * Admin can add/edit/delete units (e.g., KG, LITRE, GRAMS, PIECES) used to measure products.
3. **Party Information Management:**
   * Admin can add/edit/delete purchase party information, including details like first name, last name, contact, address, and city.
4. **Product Information Management:**
   * Admin can add/edit/delete product information, including product name, amount, quantity, purchase party name, purchase type (cash, cheque), expiry date, etc.
5. **Purchase Record Management:**
   * Admin can add/edit/delete records for purchasing items, which will increase the stock of items. This includes dynamically selecting product details added by the admin.
6. **Sales and Billing:**
   * Admin can sell items, generate bills for sales, and automatically decrease the stock when items are sold.
7. **Expiry Date Tracking:**
   * Admin can add expiry dates for products, and the system can display the nearest expiry date products when needed.
8. **Stock Information:**
   * Admin can view the current stock of items and print stock reports.
9. **Reports:**
   * Admin can generate various reports, including purchase reports between two dates, sales reports between two dates, return reports,expiry reports and reports for any specific party.
10. **Return Management:**
    * Admin can handle returns from customers, which will affect the stock accordingly.
11. **Bill Generation:**
    * Admin can generate bills for sales transactions.
12. **Editing Stock Items:**
    * Admin can edit stock item details.

* **User Functionalities:**

1. **Registration and Authentication:**
   * Users can register and log in to the system with a username and password.
   * Users can log out of the system when the session is complete.
2. **Purchase Items:**
   * Users can add/edit/delete purchase items, increasing the stock when items are purchased. Users can select product details dynamically, as added by the admin.
3. **Sales and Billing:**
   * Users can sell items, generate bills for sales, and automatically decrease the stock when items are sold.
4. **Return Management:**
   * Users can handle returns from customers, which will affect the stock accordingly.
5. **Bill Generation:**
   * Users can generate bills for sales transactions.
6. **Reports:**
   * User can generate various reports, including purchase reports between two dates, sales reports between two dates.

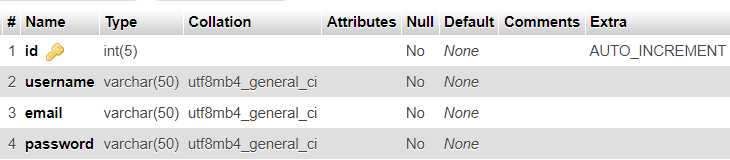
**6. DETAIL PLANNING**

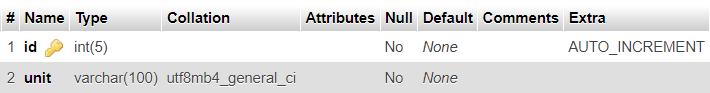
6.1 Data Dictionary

6.2 Entity – Relationship diagram

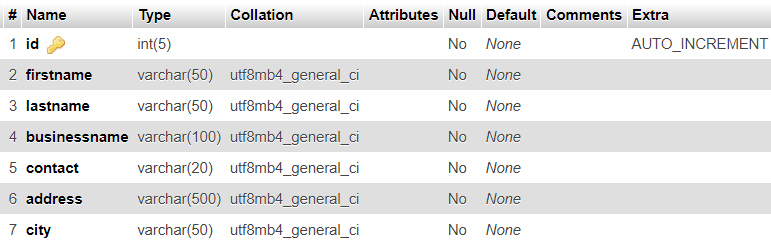
* 1. **Data Dictionary**

**Reglogin Table:**

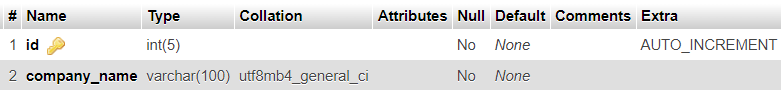
****

**Units Table:**

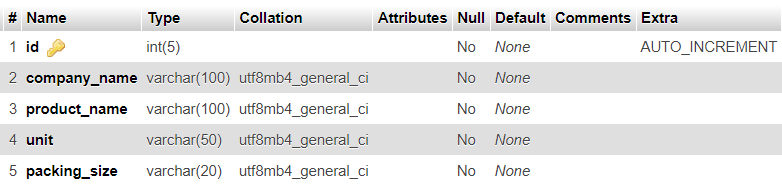
**Party\_Info Table:**

****

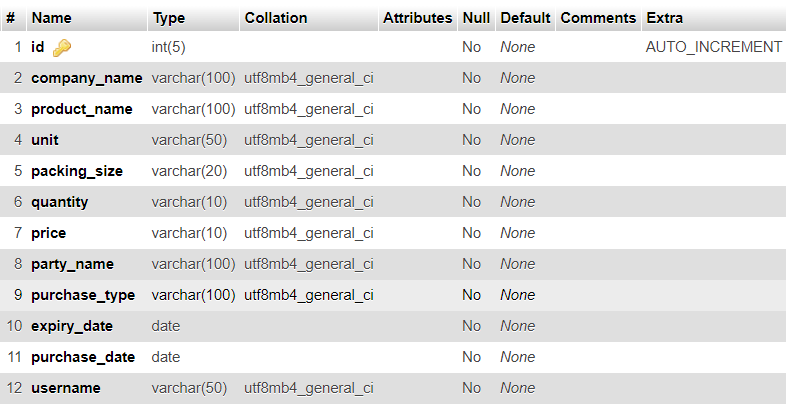
**Company Table:**

****

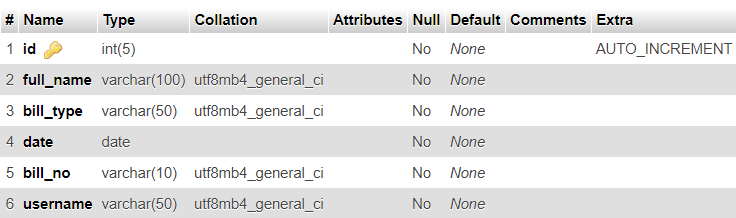
**Products Table:**

****

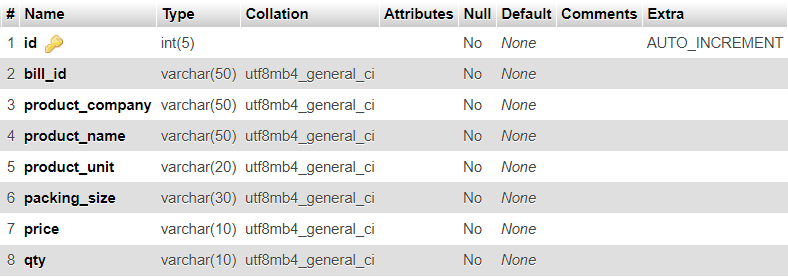
**Purchase\_master Table:**

****

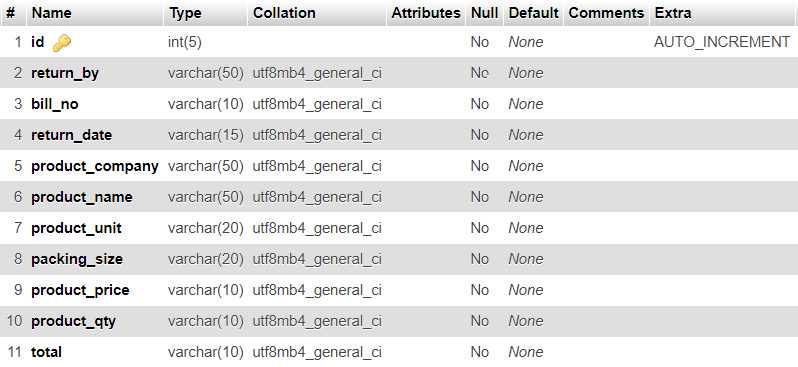
**Billing\_header Table:**

****

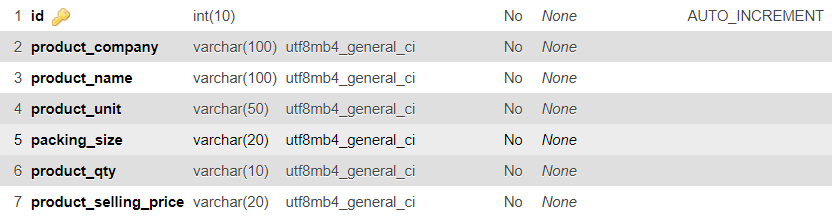
**Billing\_details Table:**

****

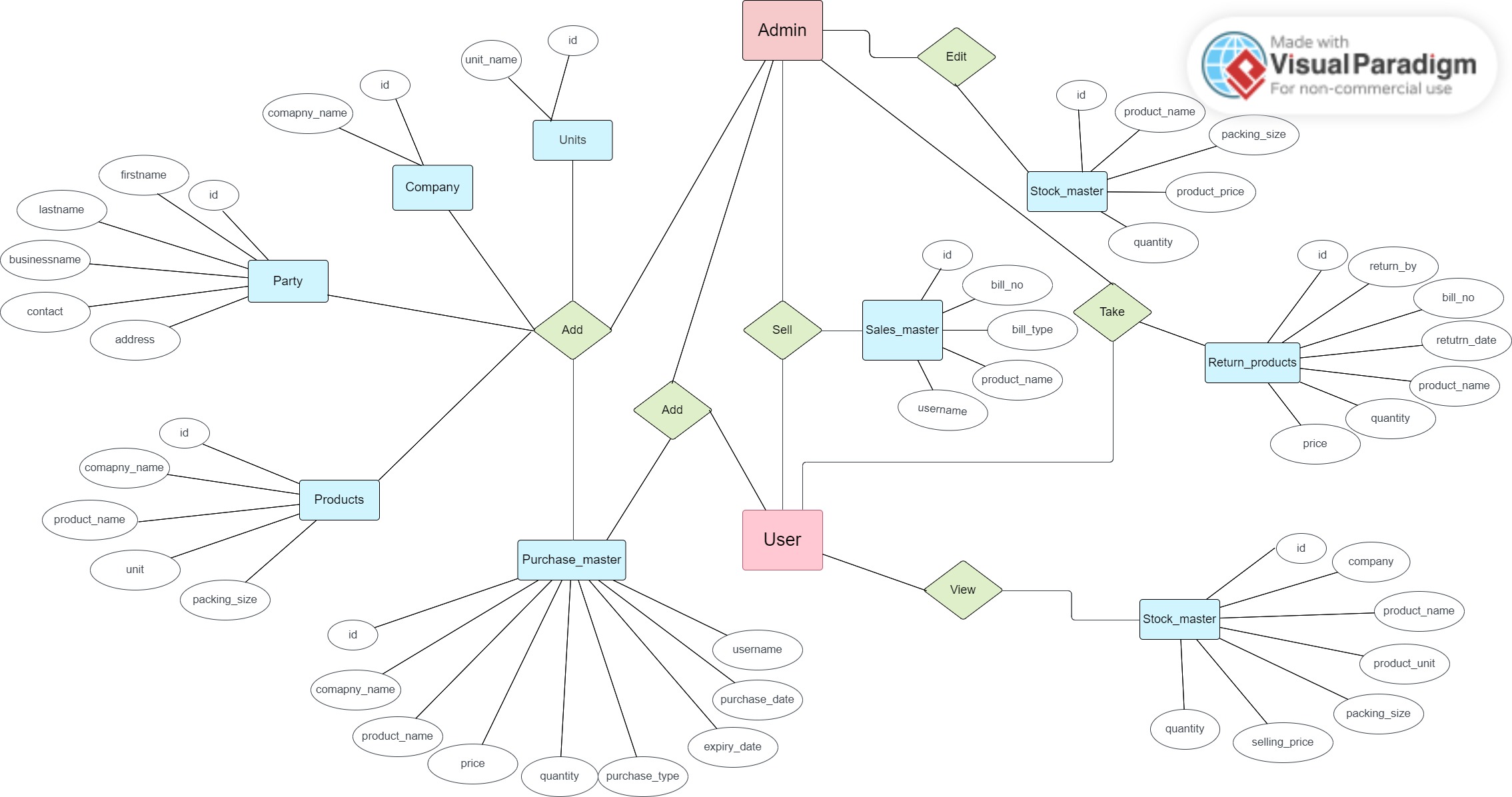
**Return\_products Table:**

****

**Stock\_master Table:**

****

* 1. **Entity – Relationship diagram**

****

**7. SYSTEM DESIGN**

* 1. **Input Design**
  2. **Output Design**
  3. **Screenshots Of the system**

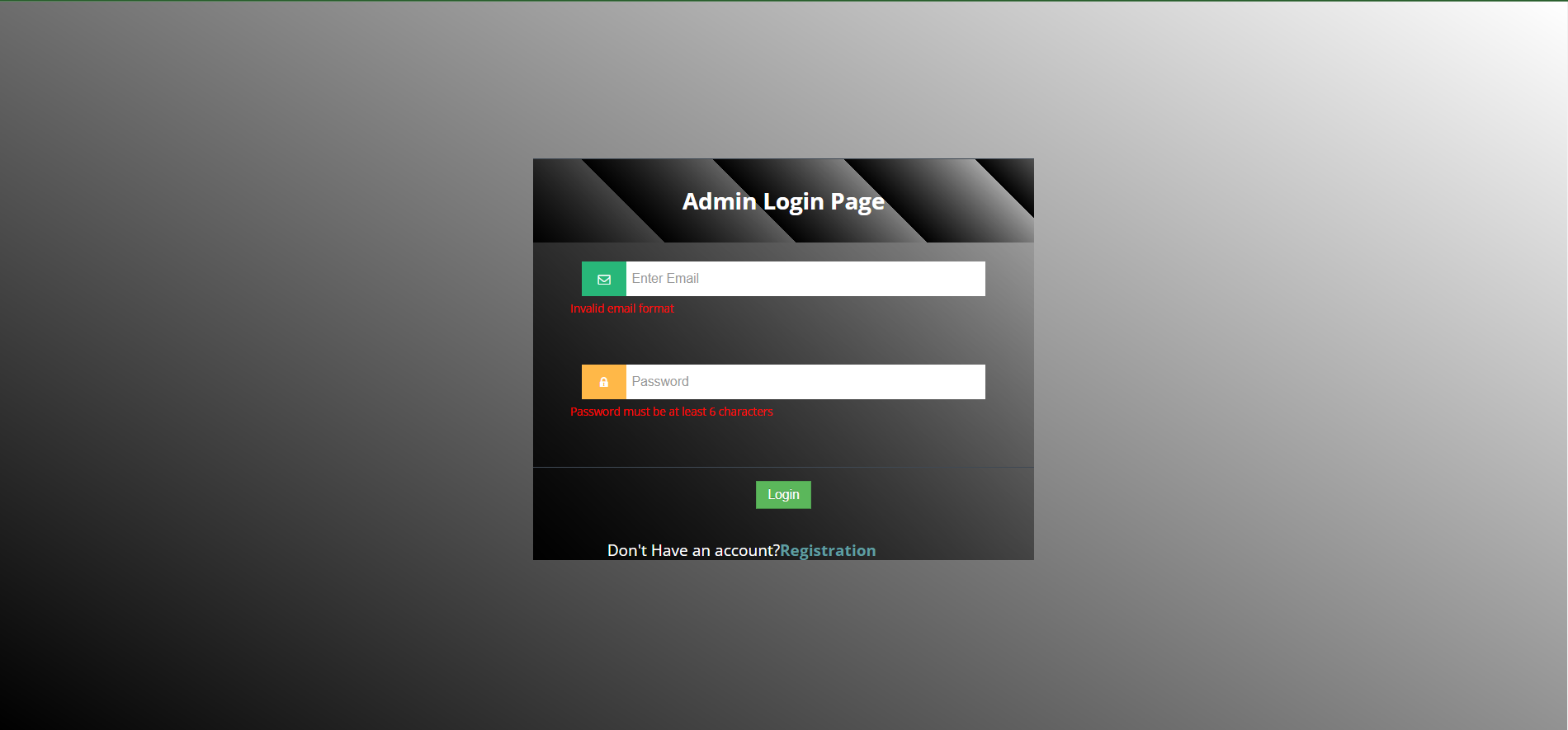
**Home Page**

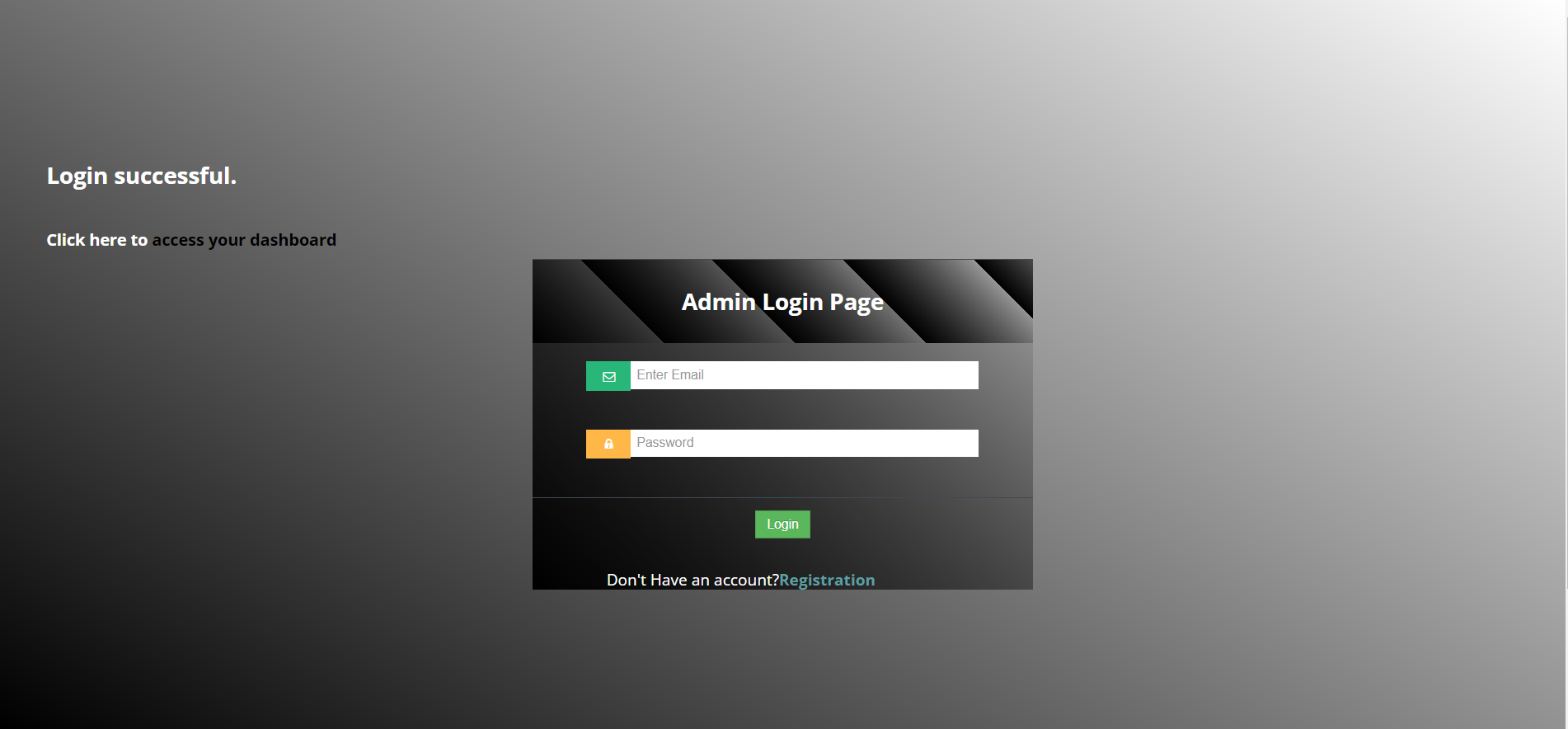


**Admin Panel**

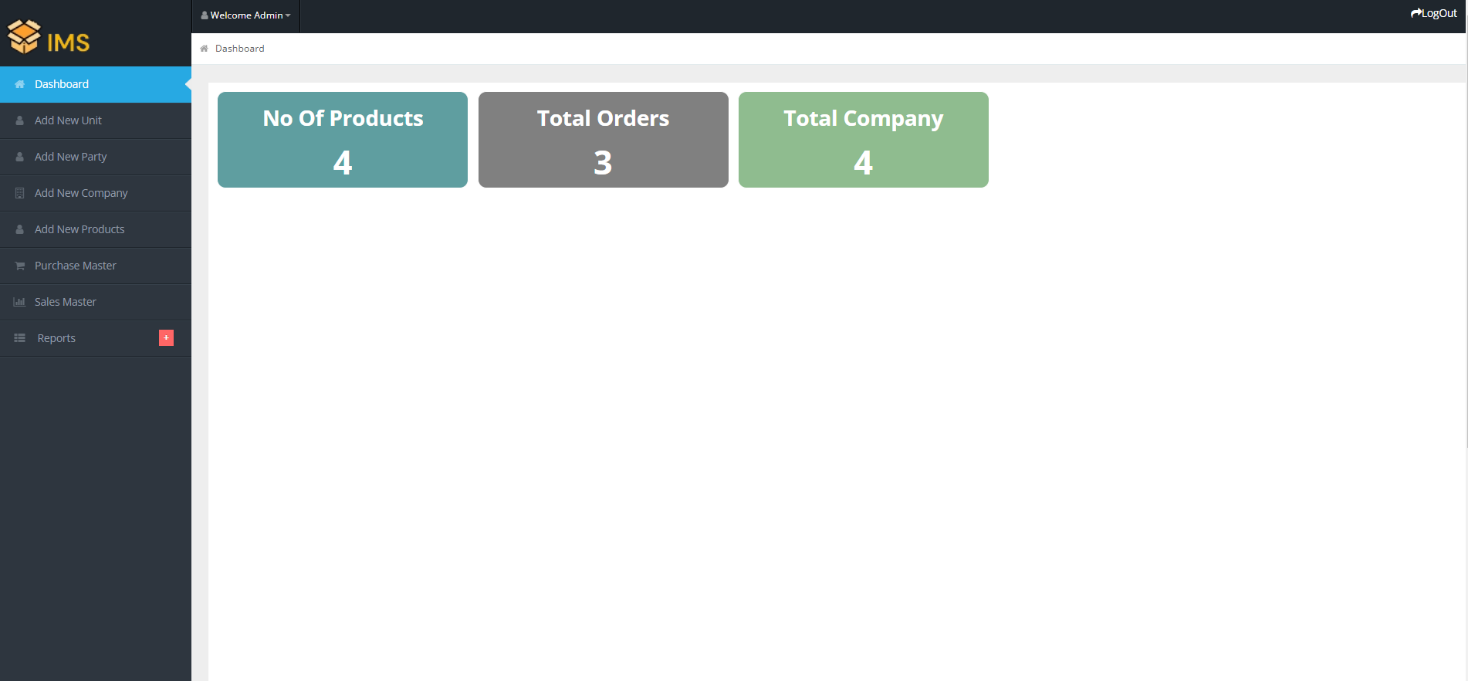
**Login:**

* validation

****

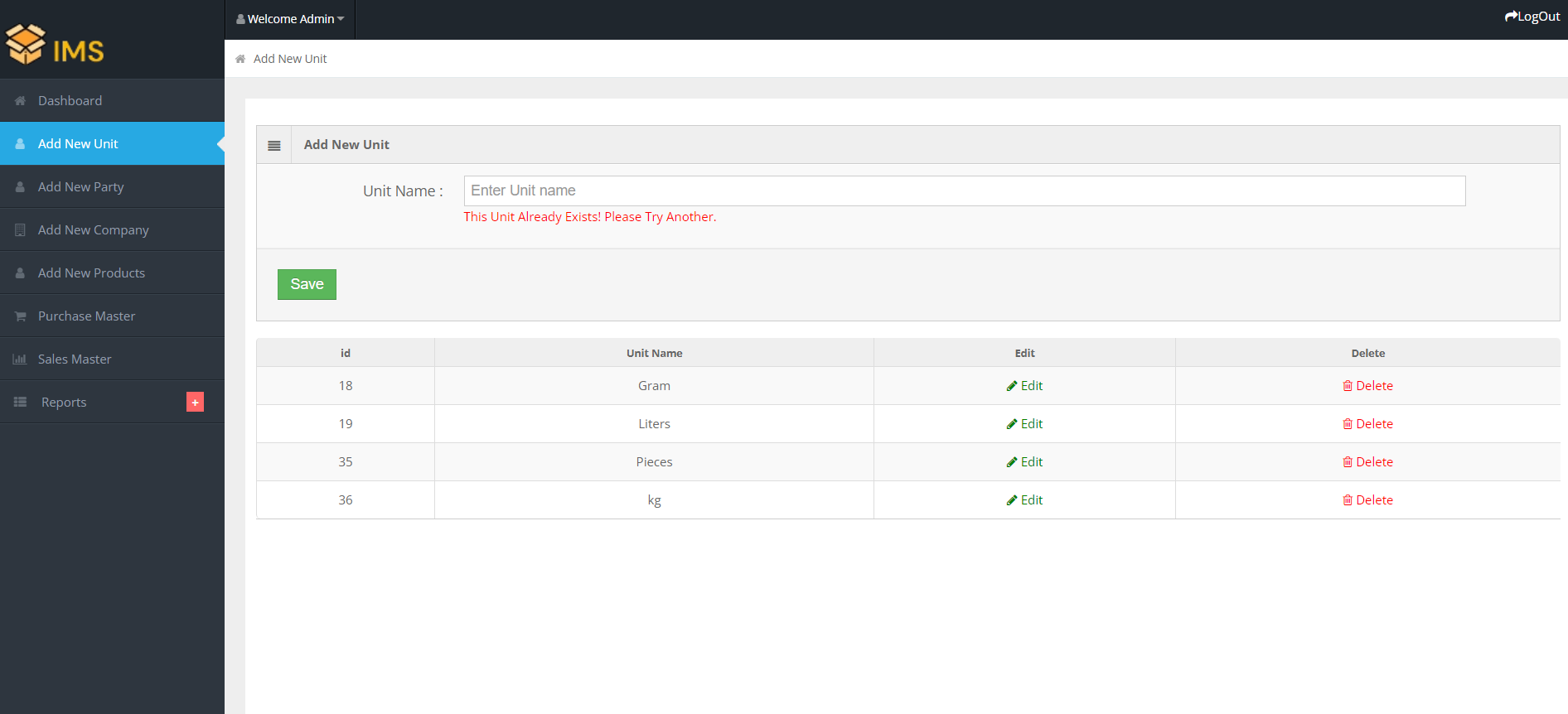


**Dashboard**

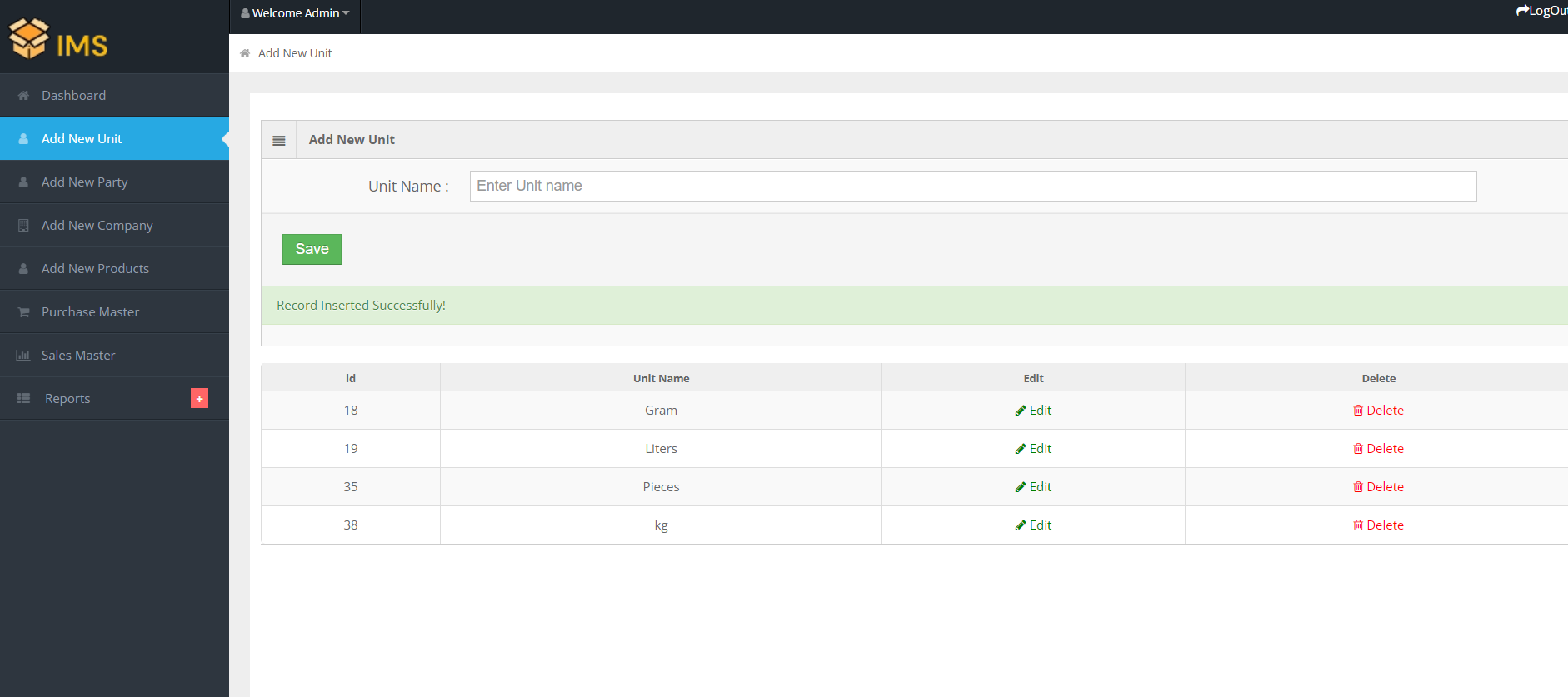


**Add New Unit:**

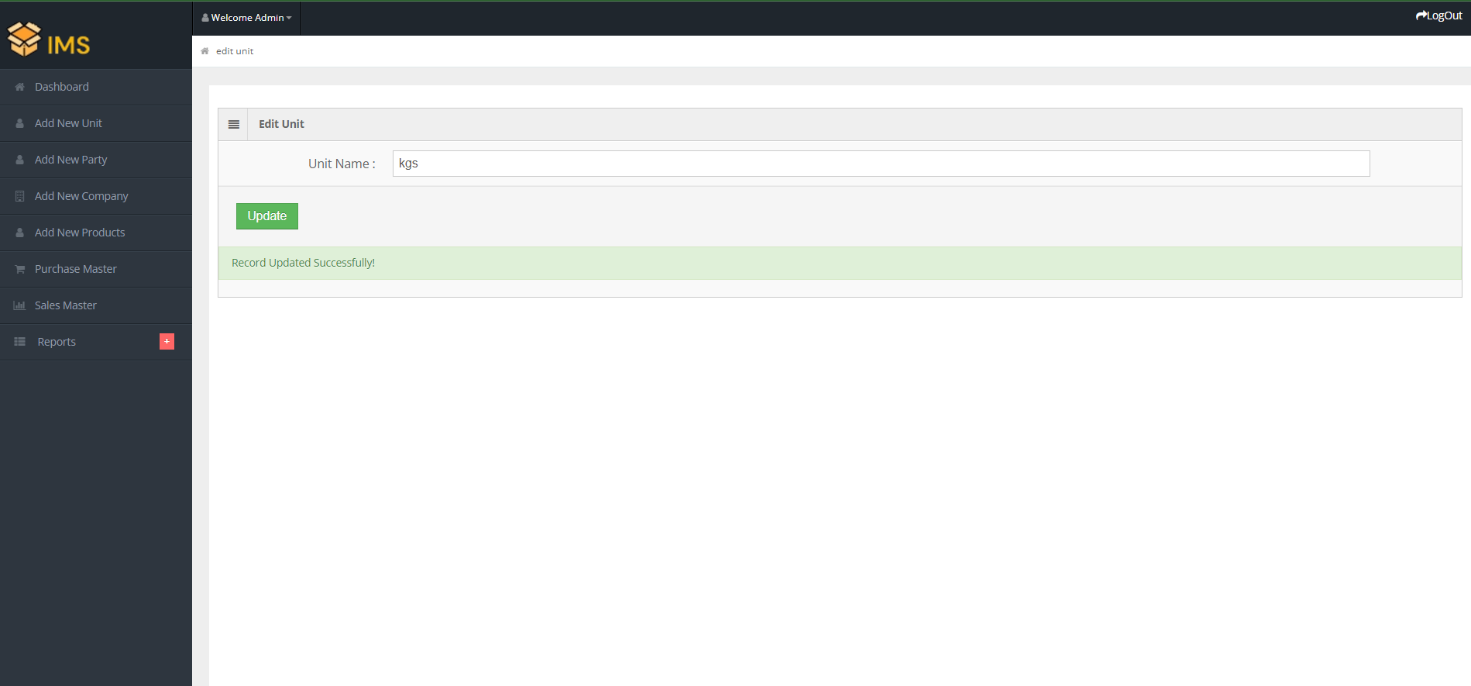
* validation



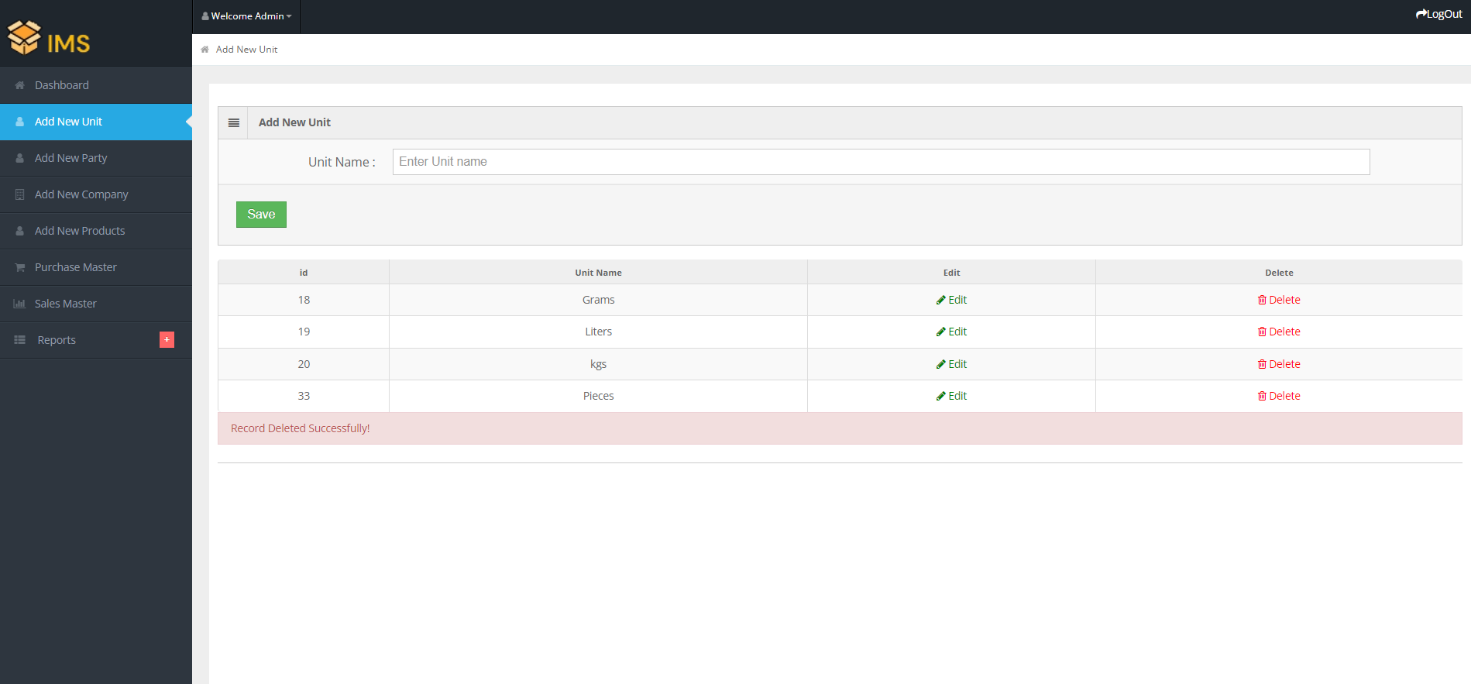
* Insert/View

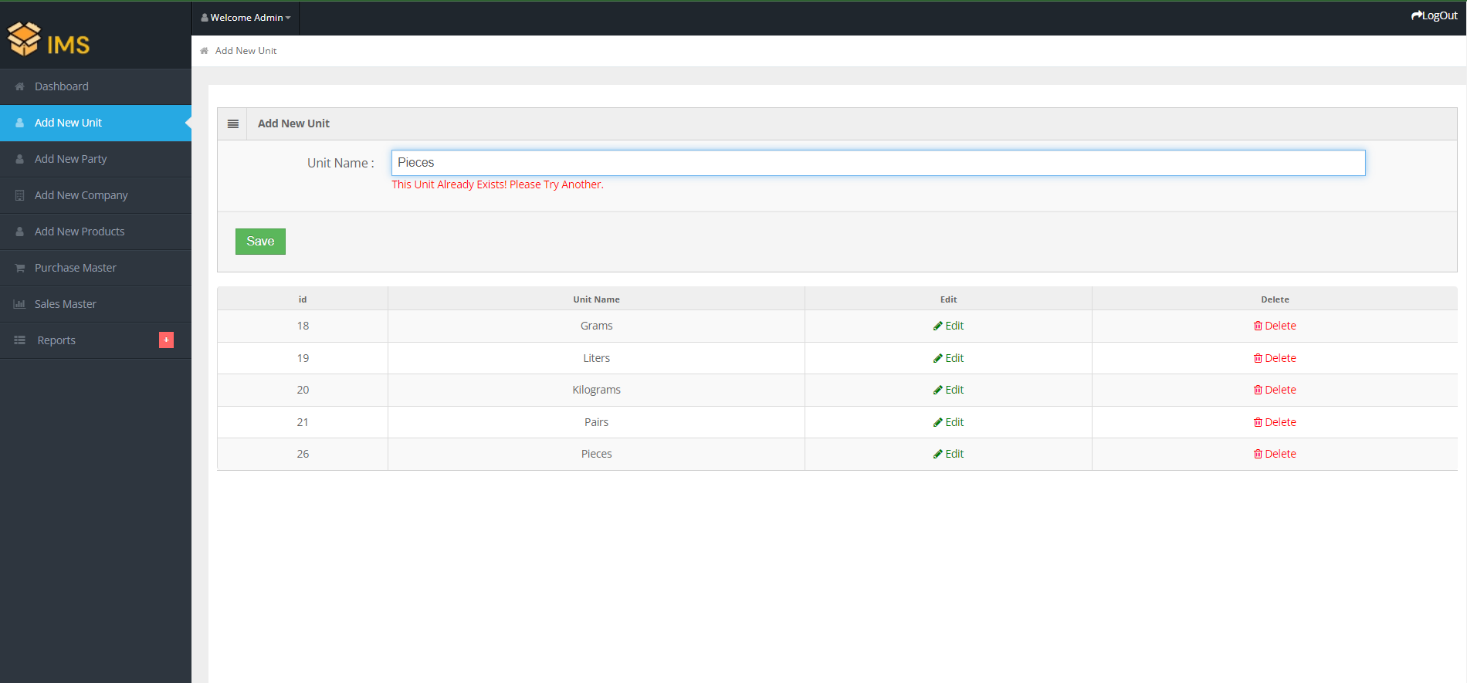
****

* Update



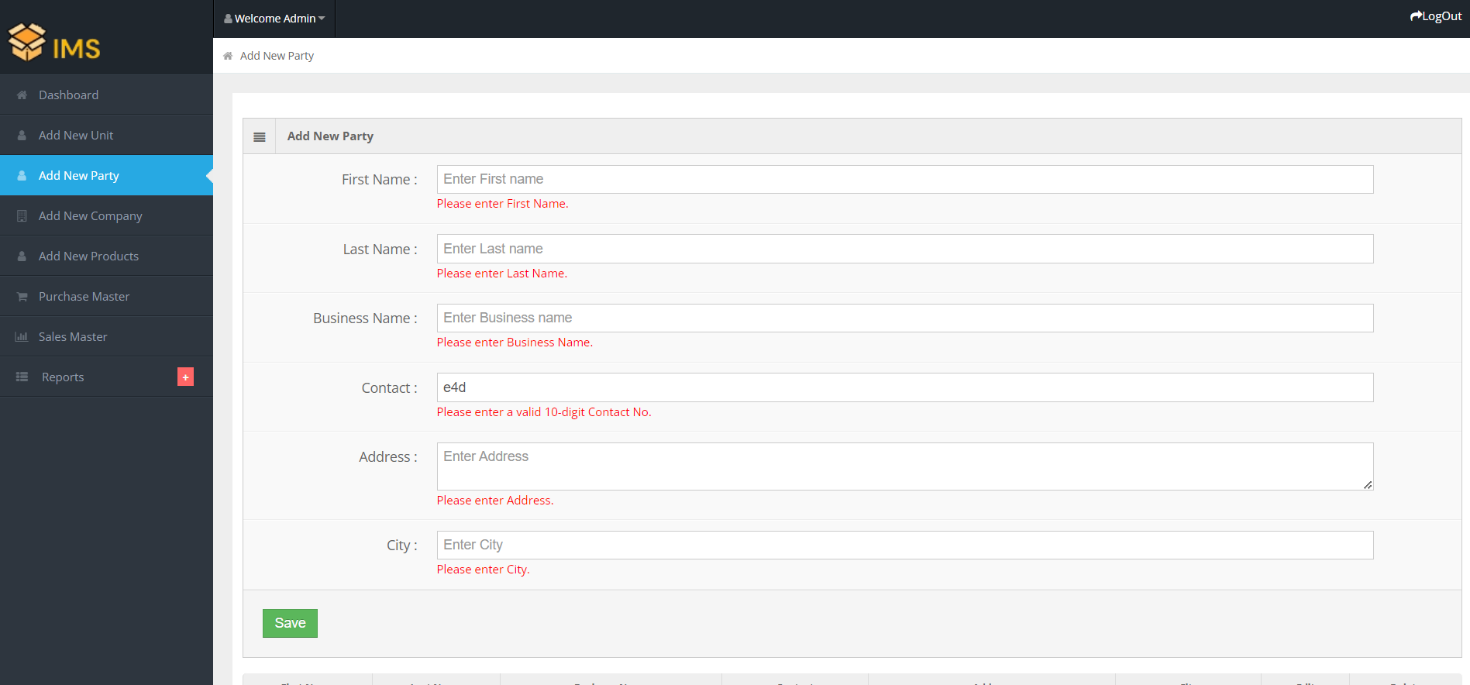
* Delete



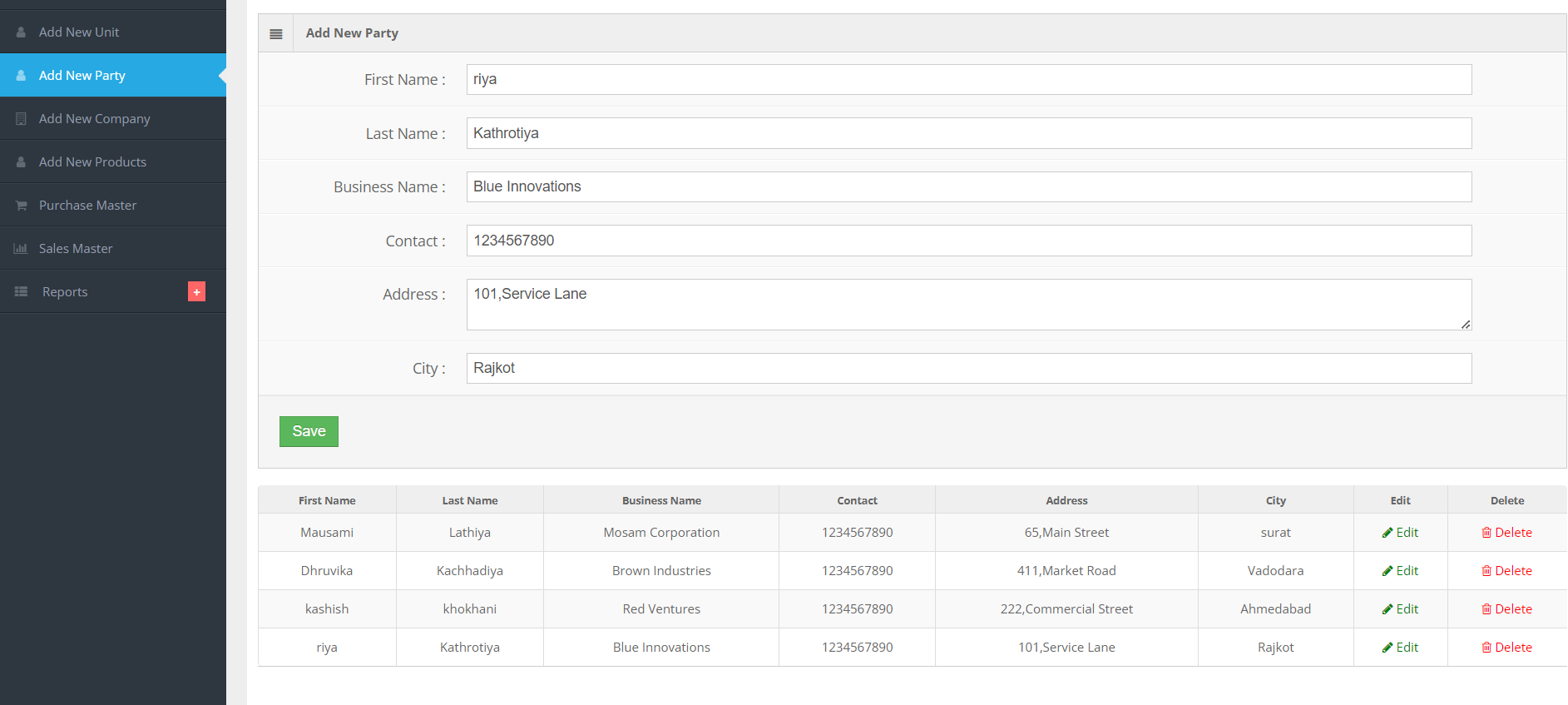
****

**Add New Party**:

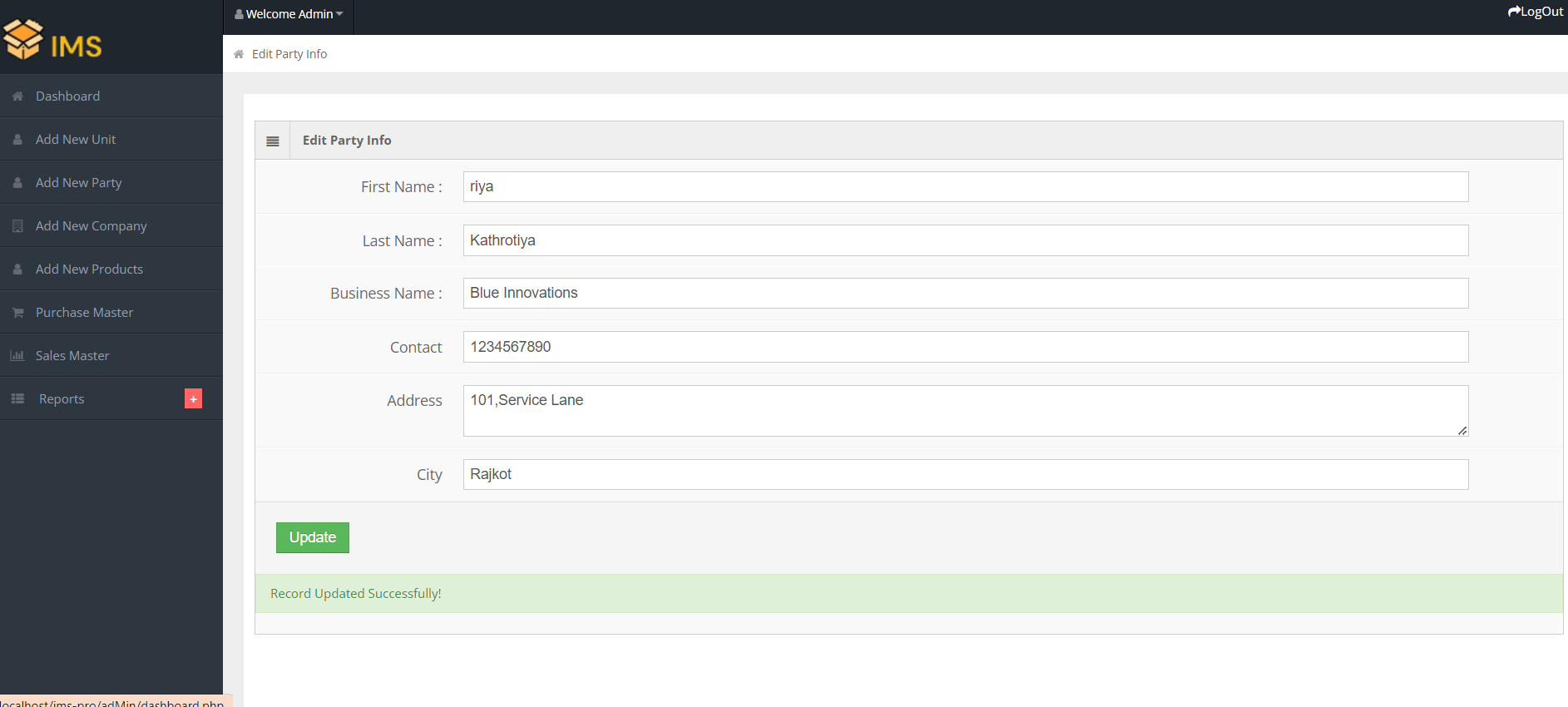
* validations



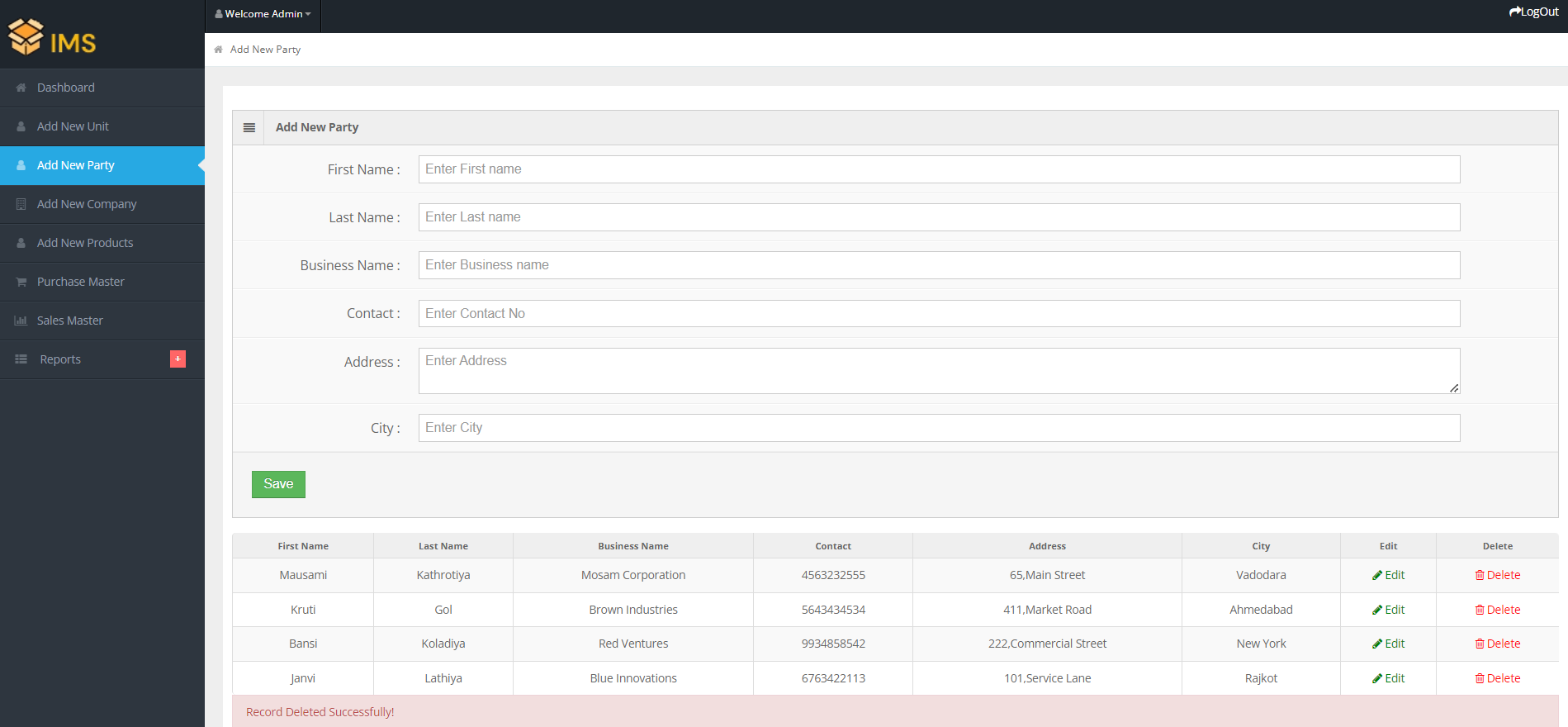
* Insert/View



* Update

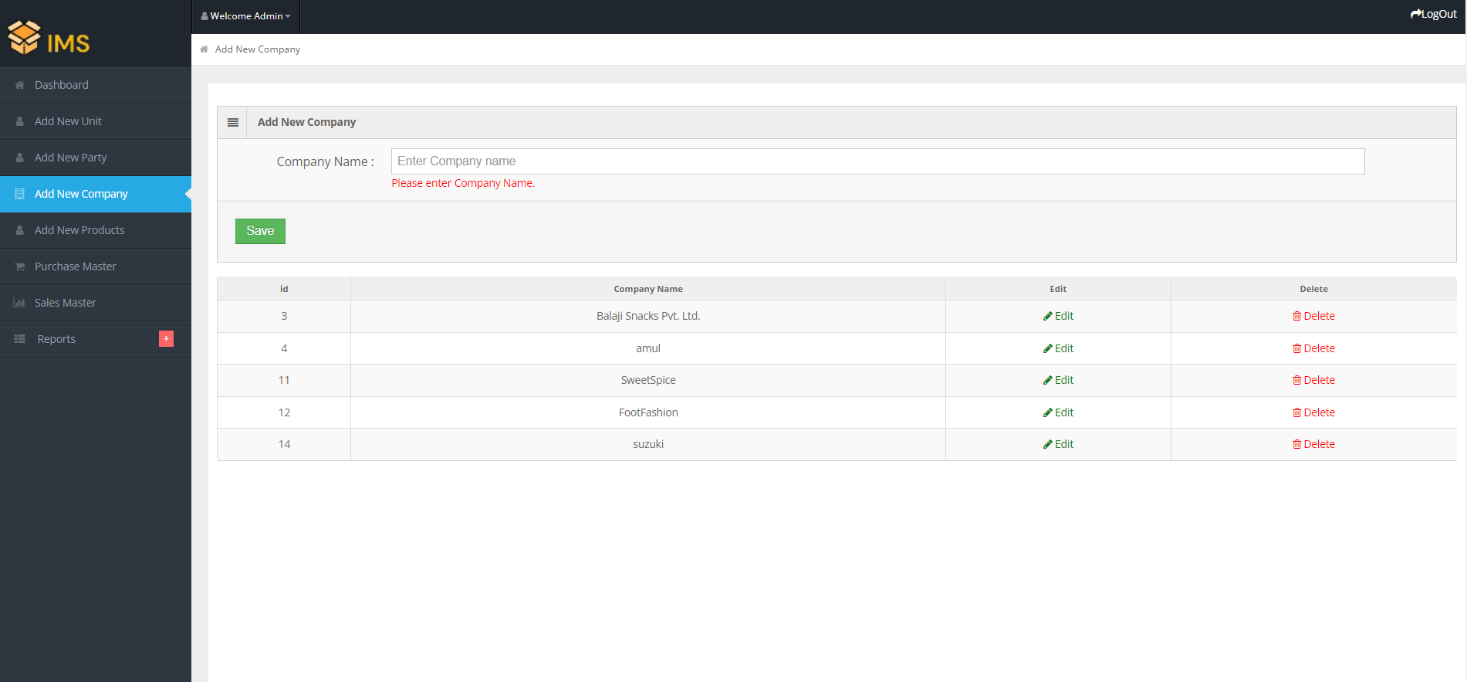


* Delete

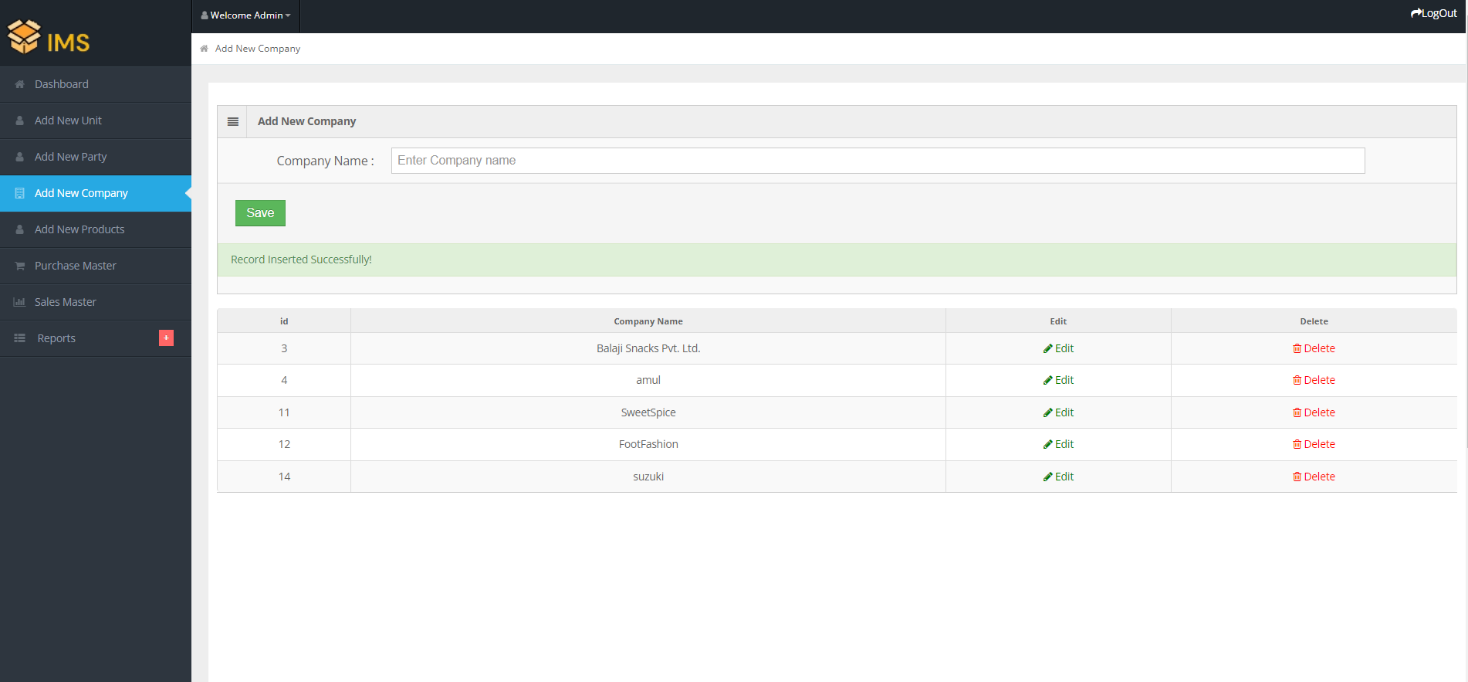


**Add New Company:**

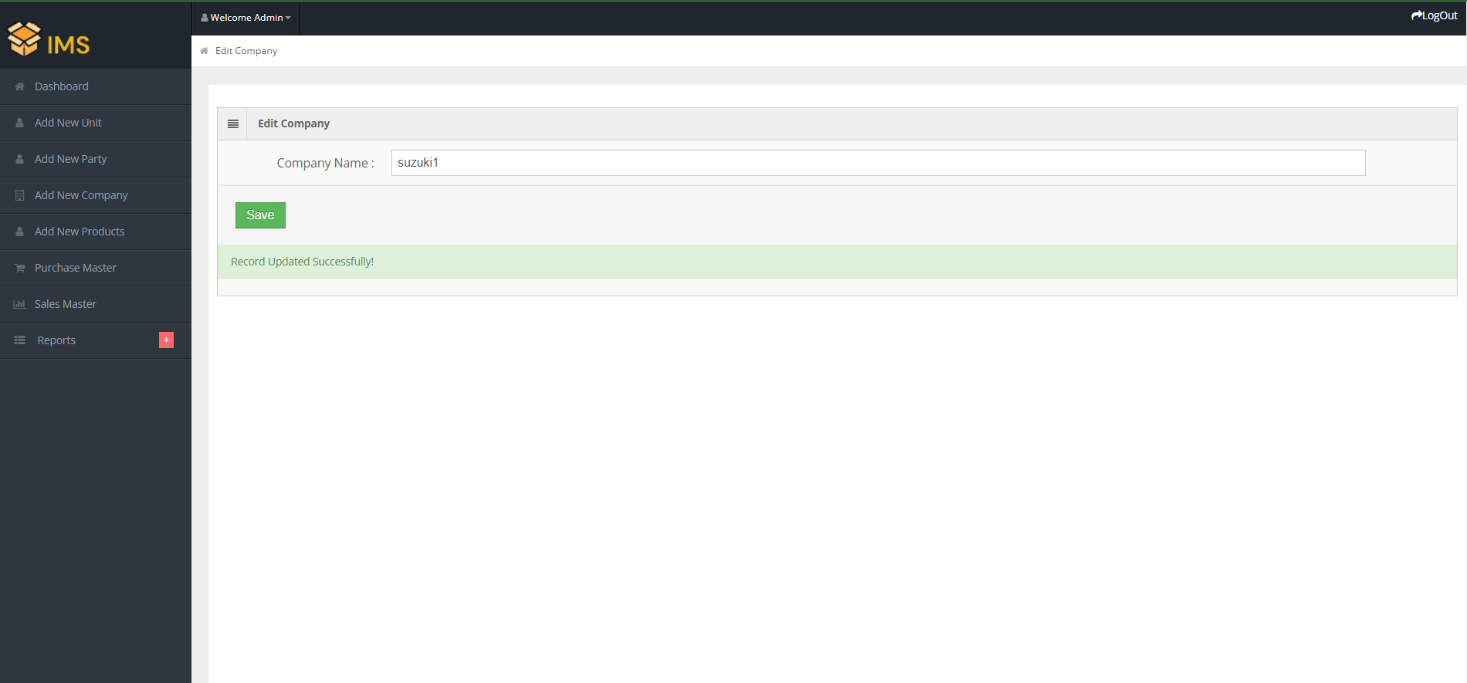
* validations



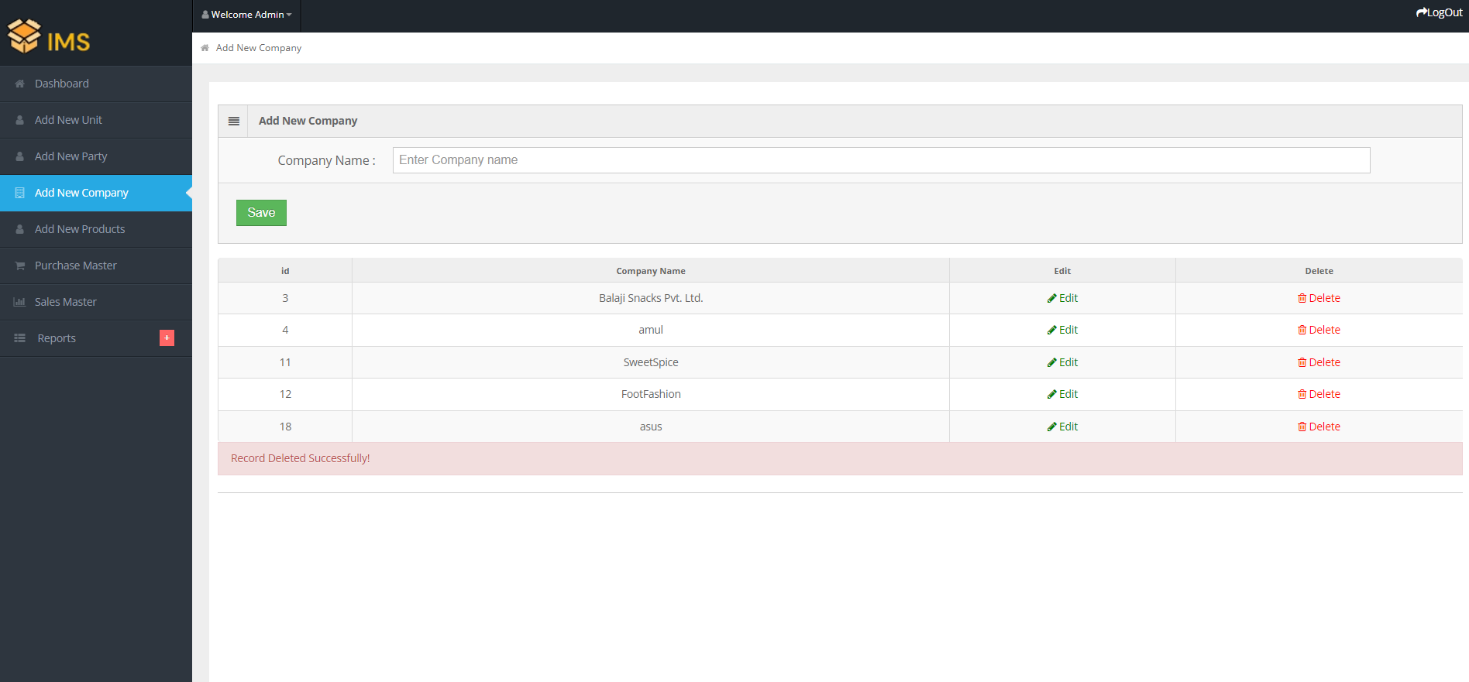
* Insert/View

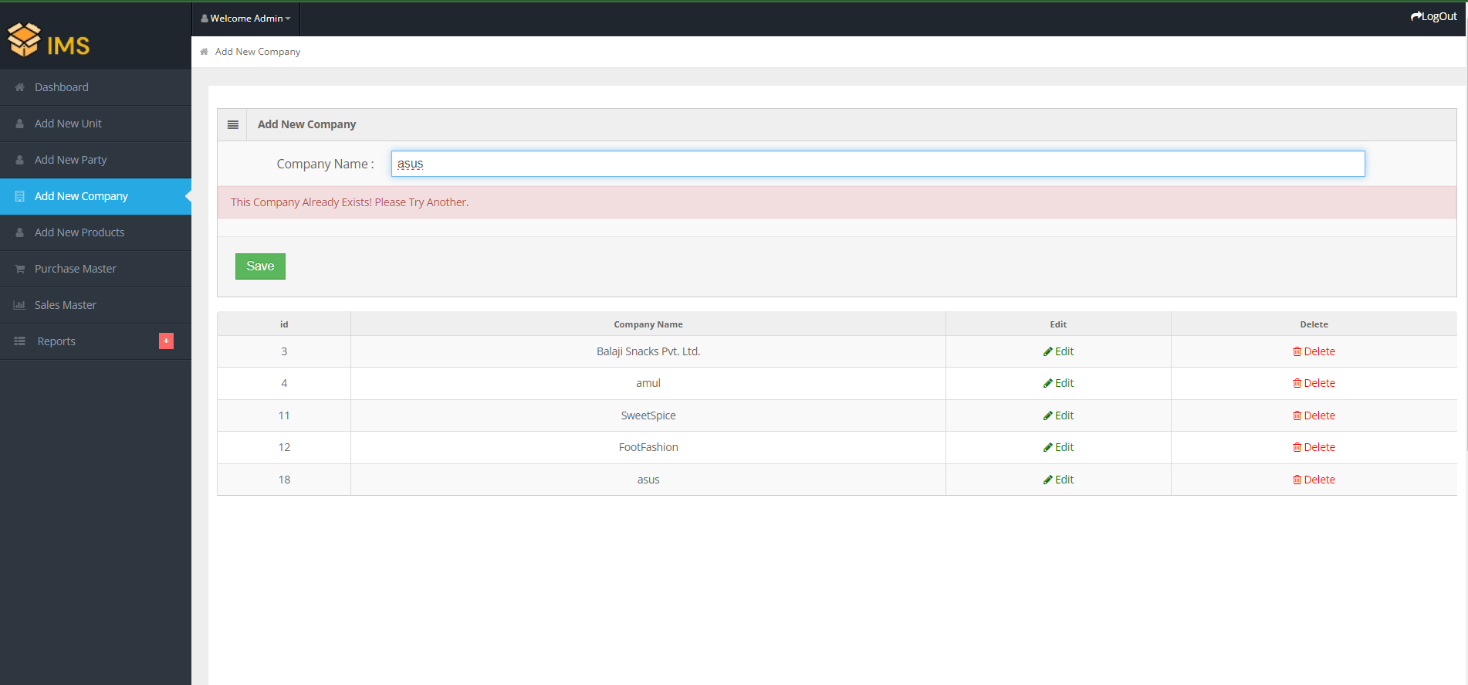


* Update



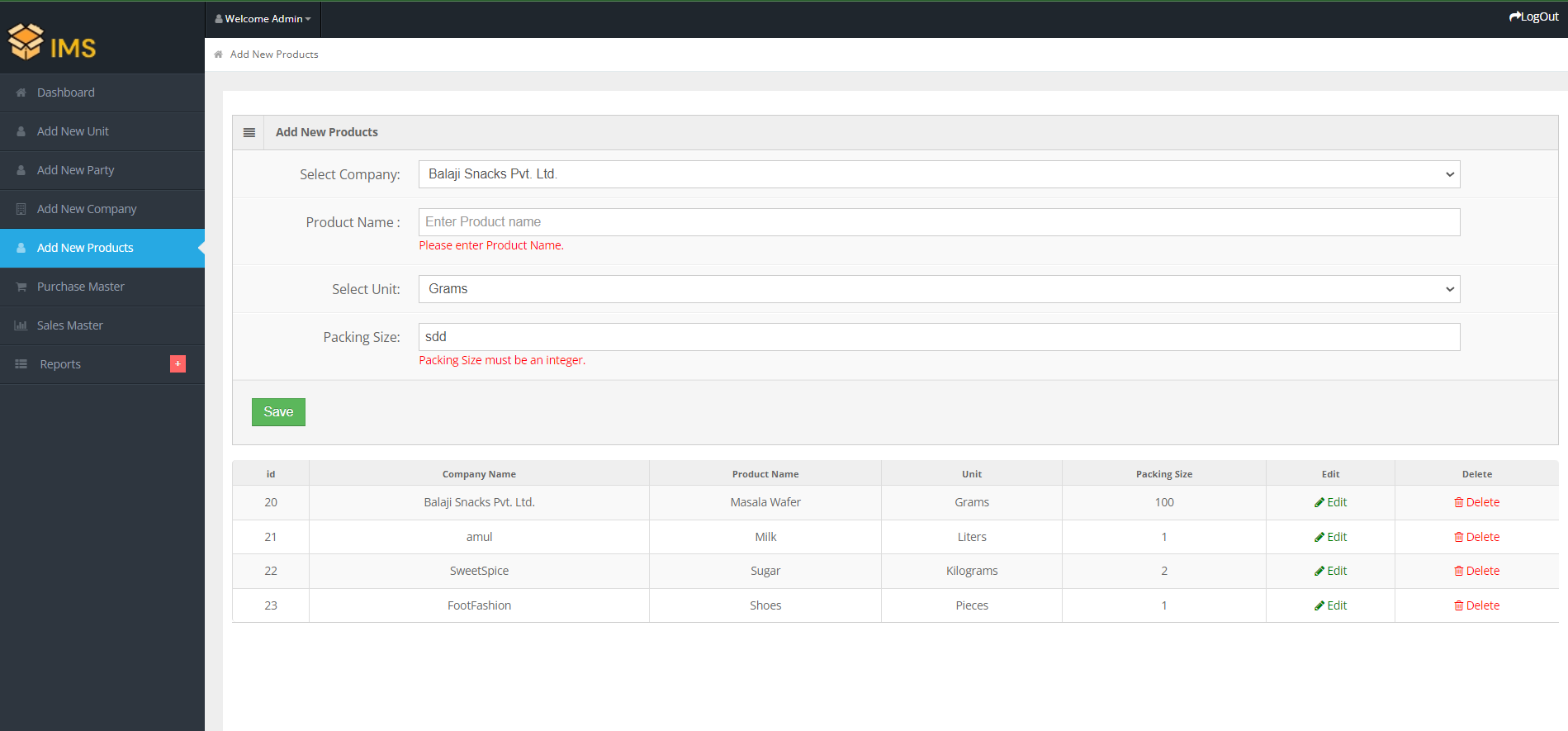
* Delete



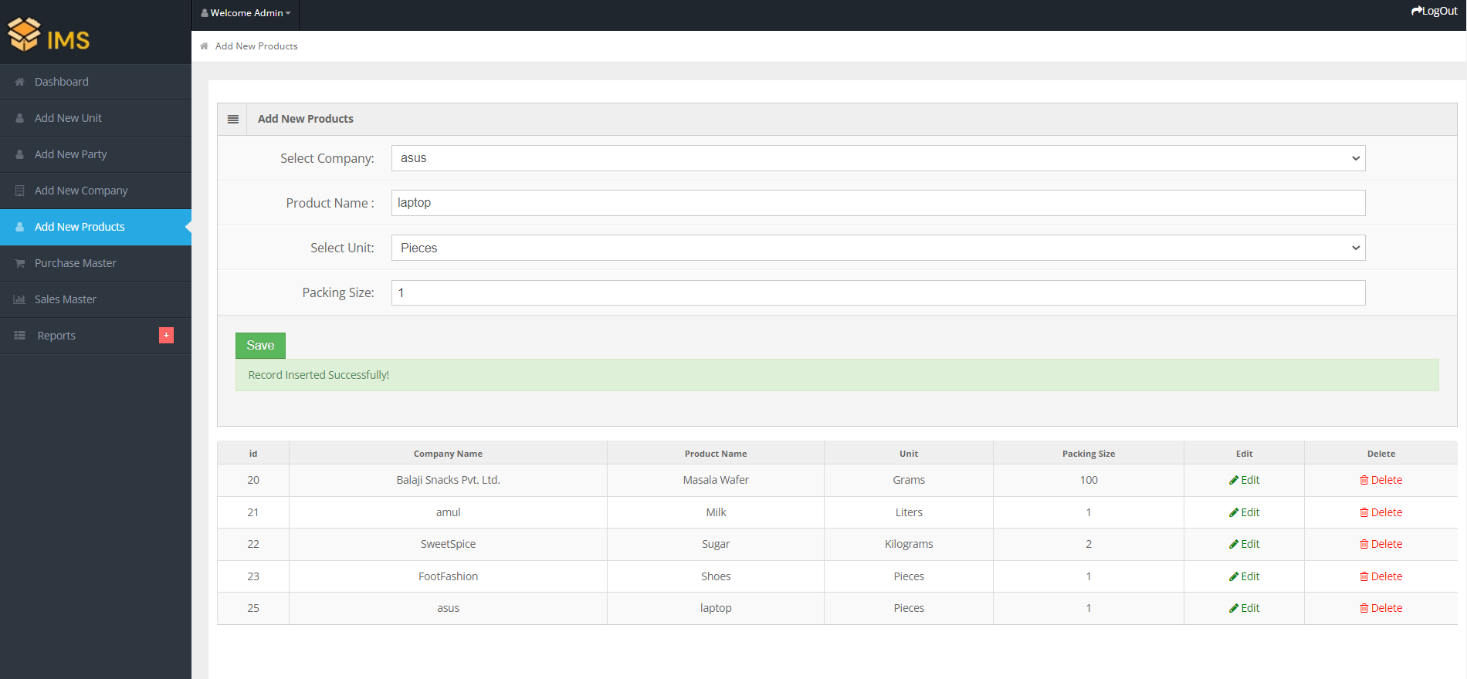


**Add New Products:**

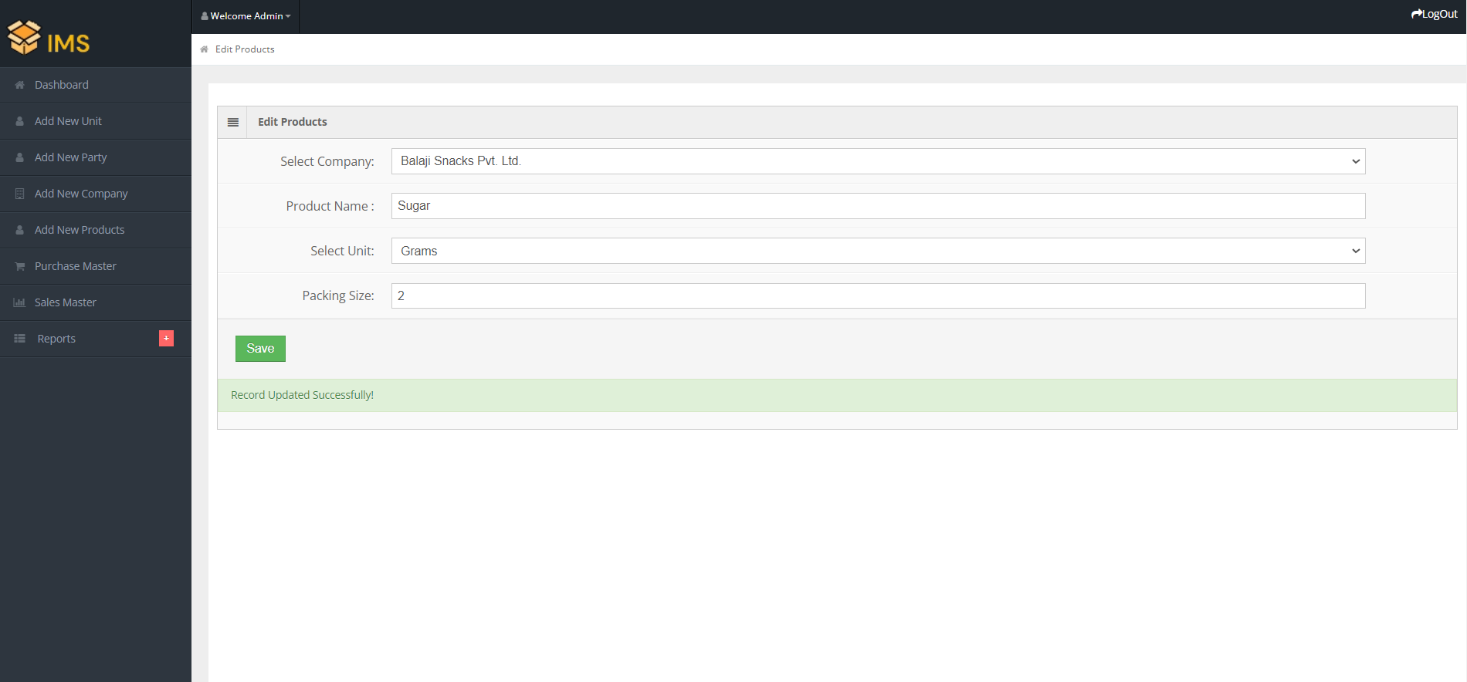
* validations



* Insert/View

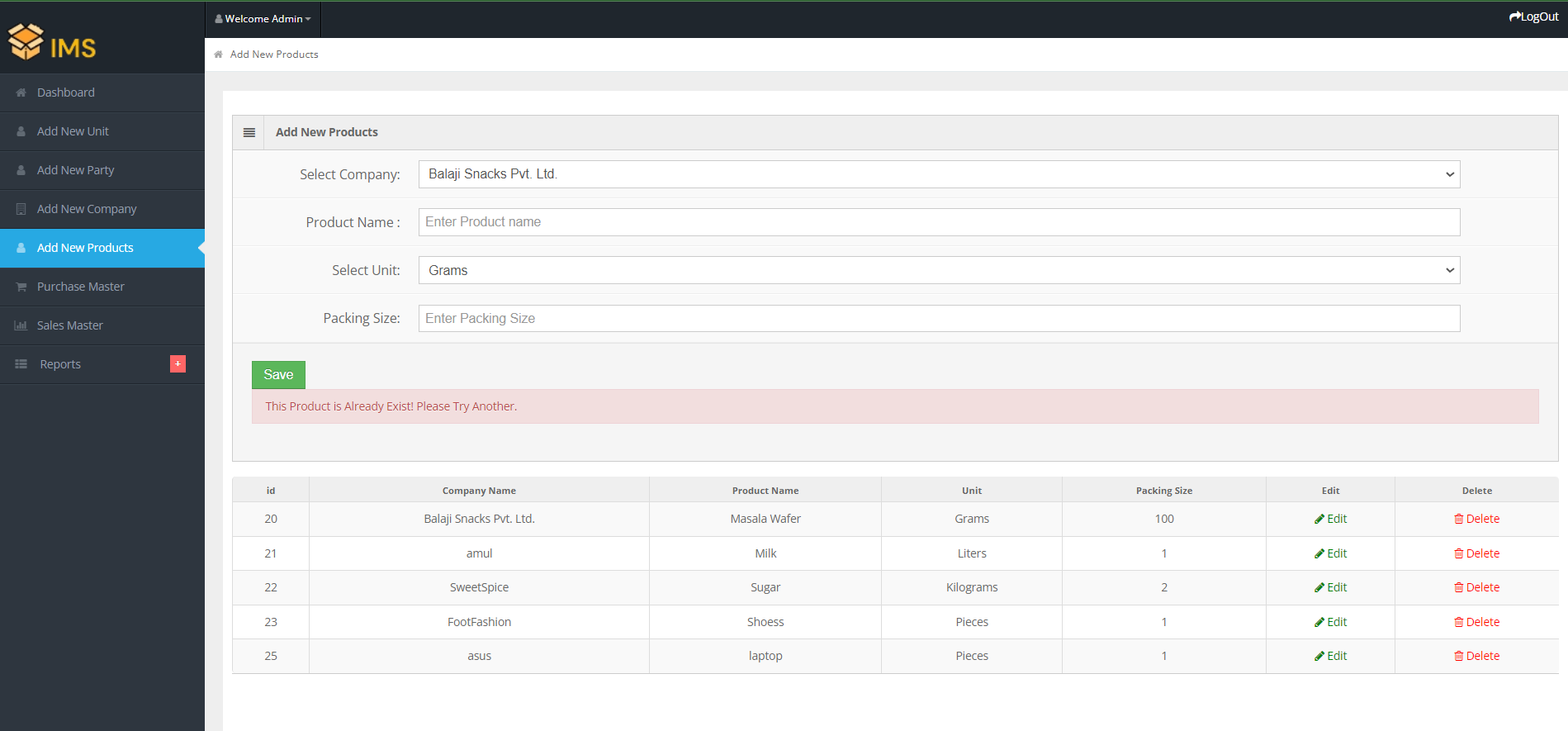


* Update



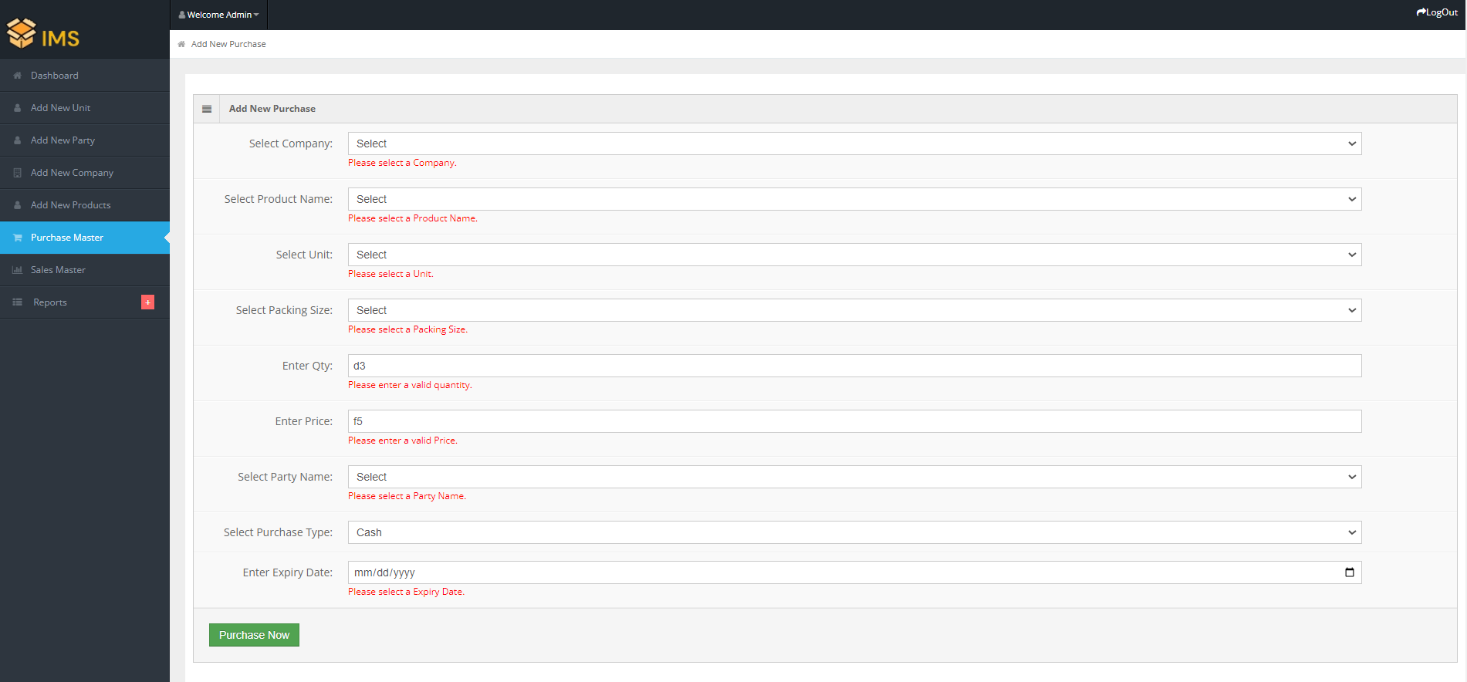
* Delete



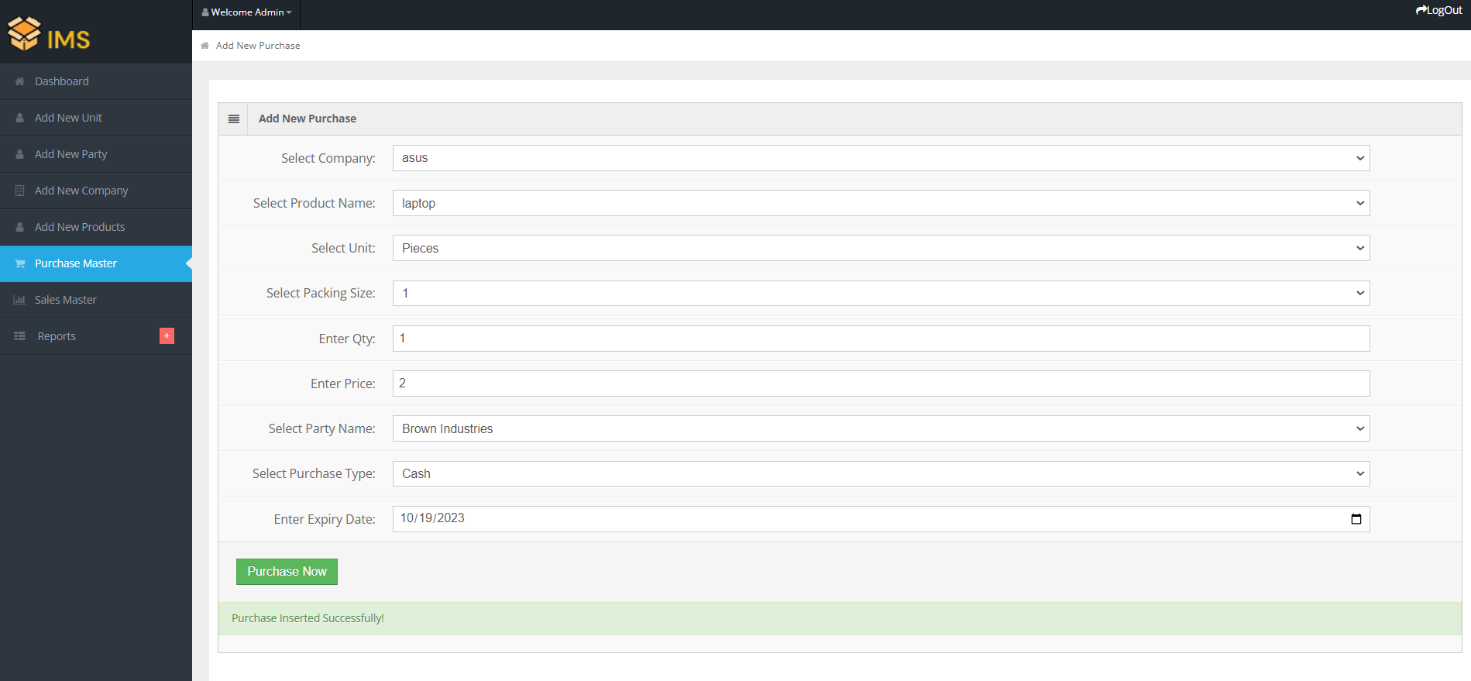


**Purchase Master:**

* validations

****

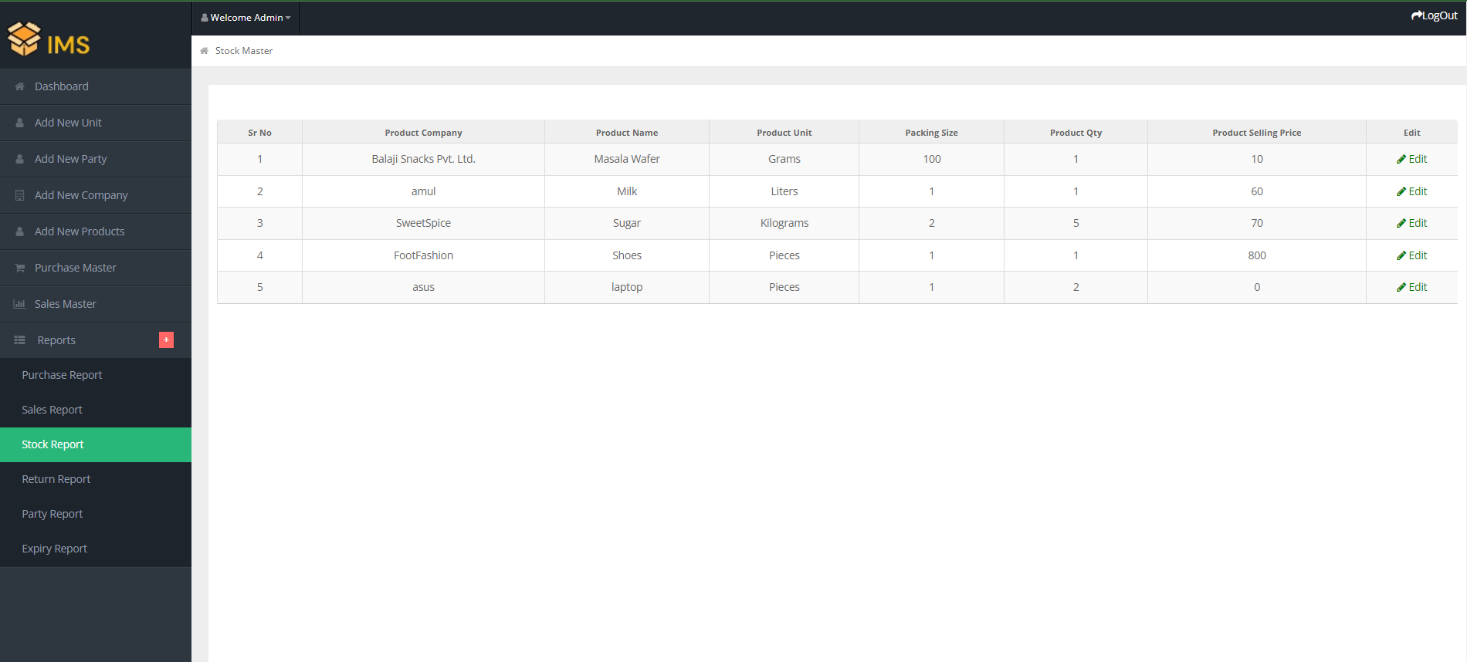
* Insert/View

****

**Purchase Report**

****

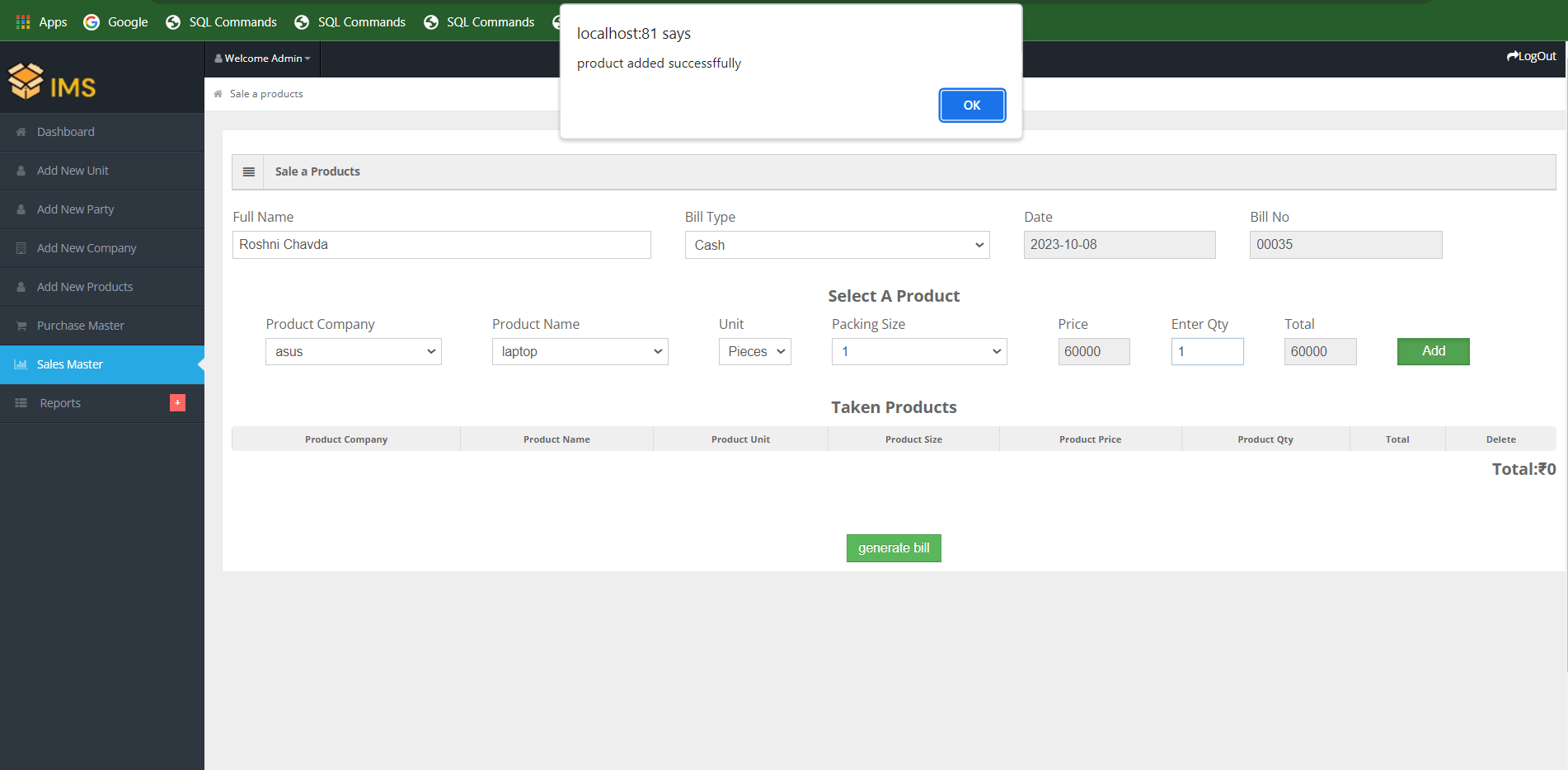
**Stock Report**

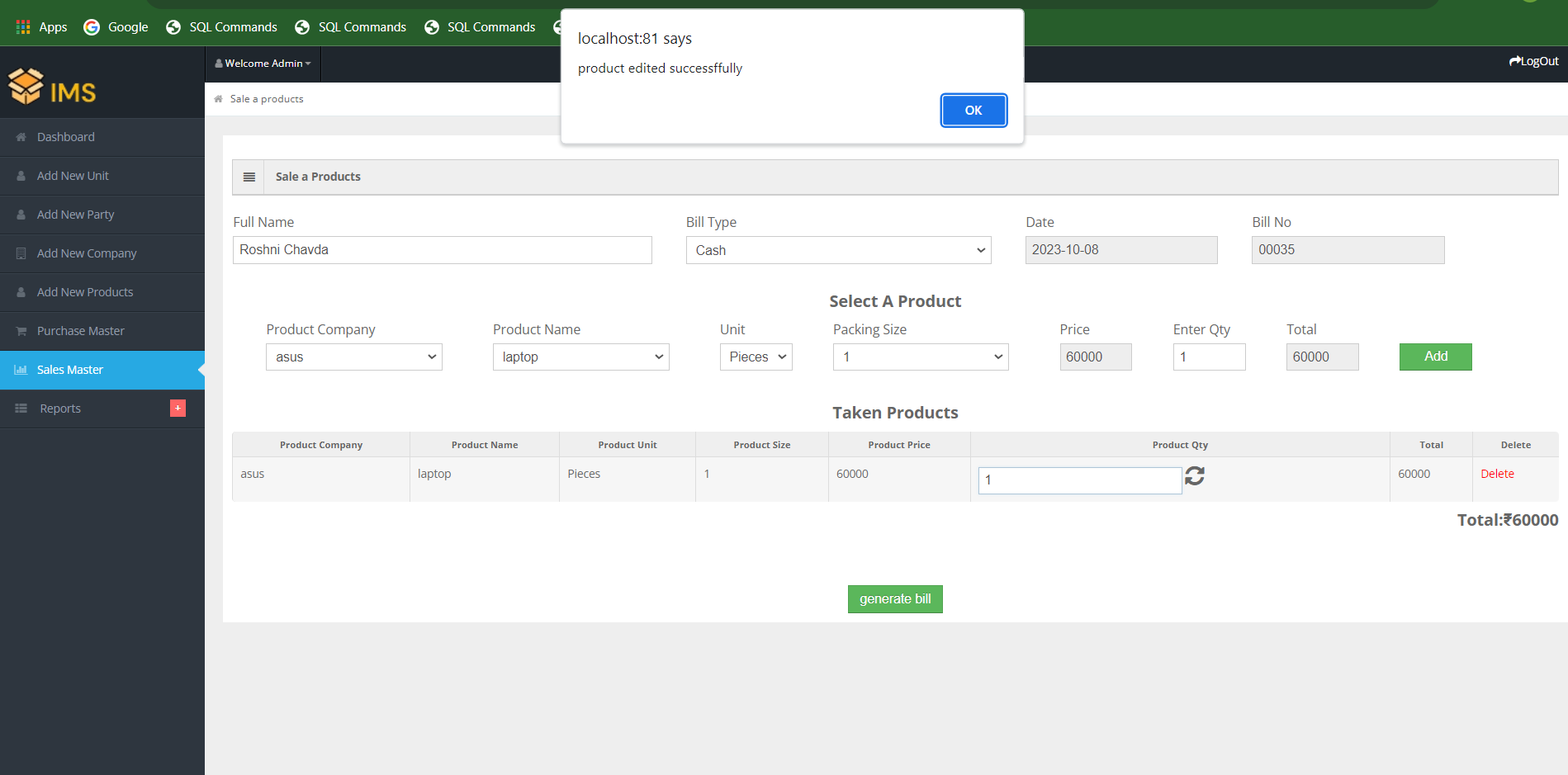
****

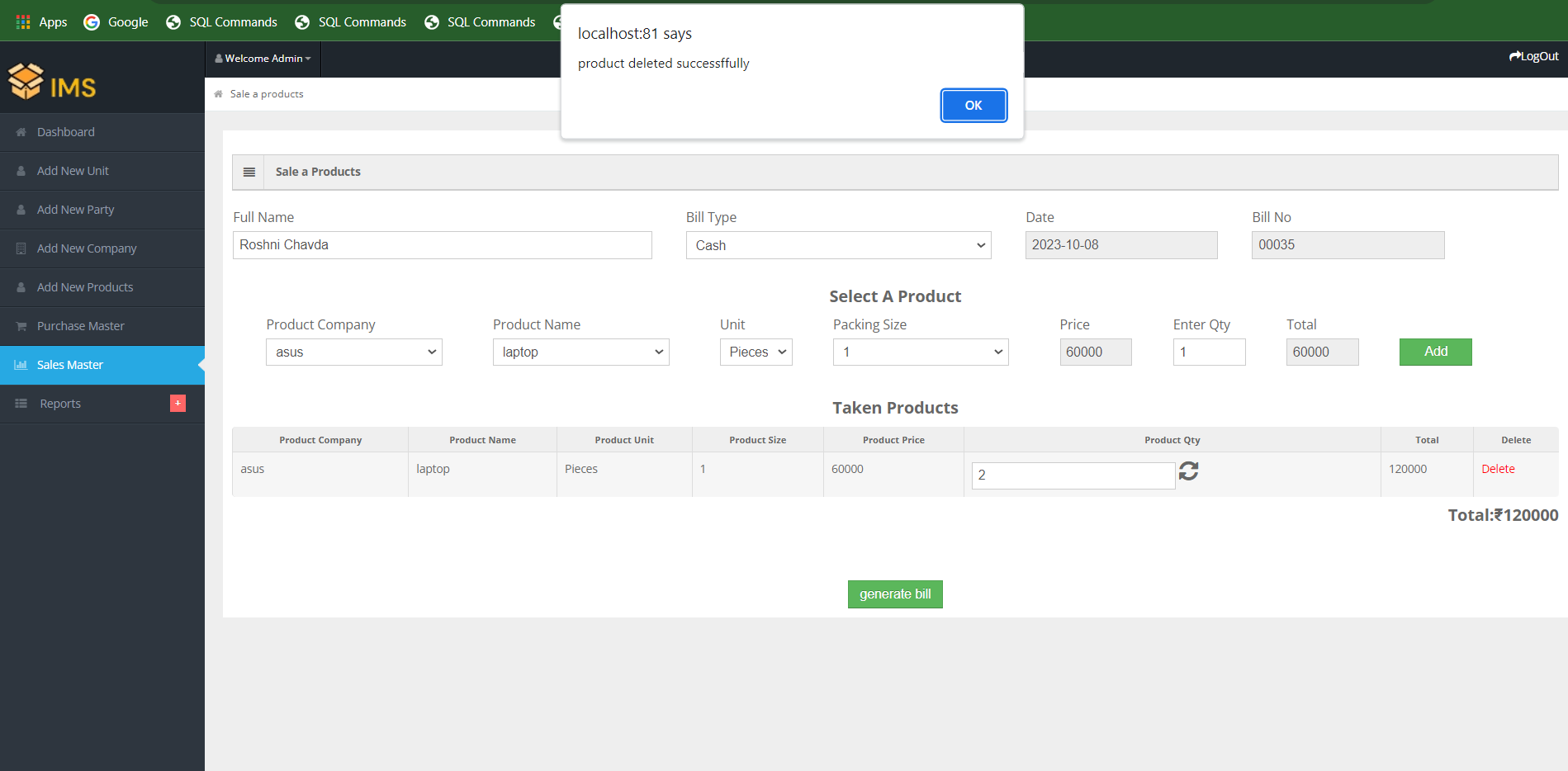
* **Update**

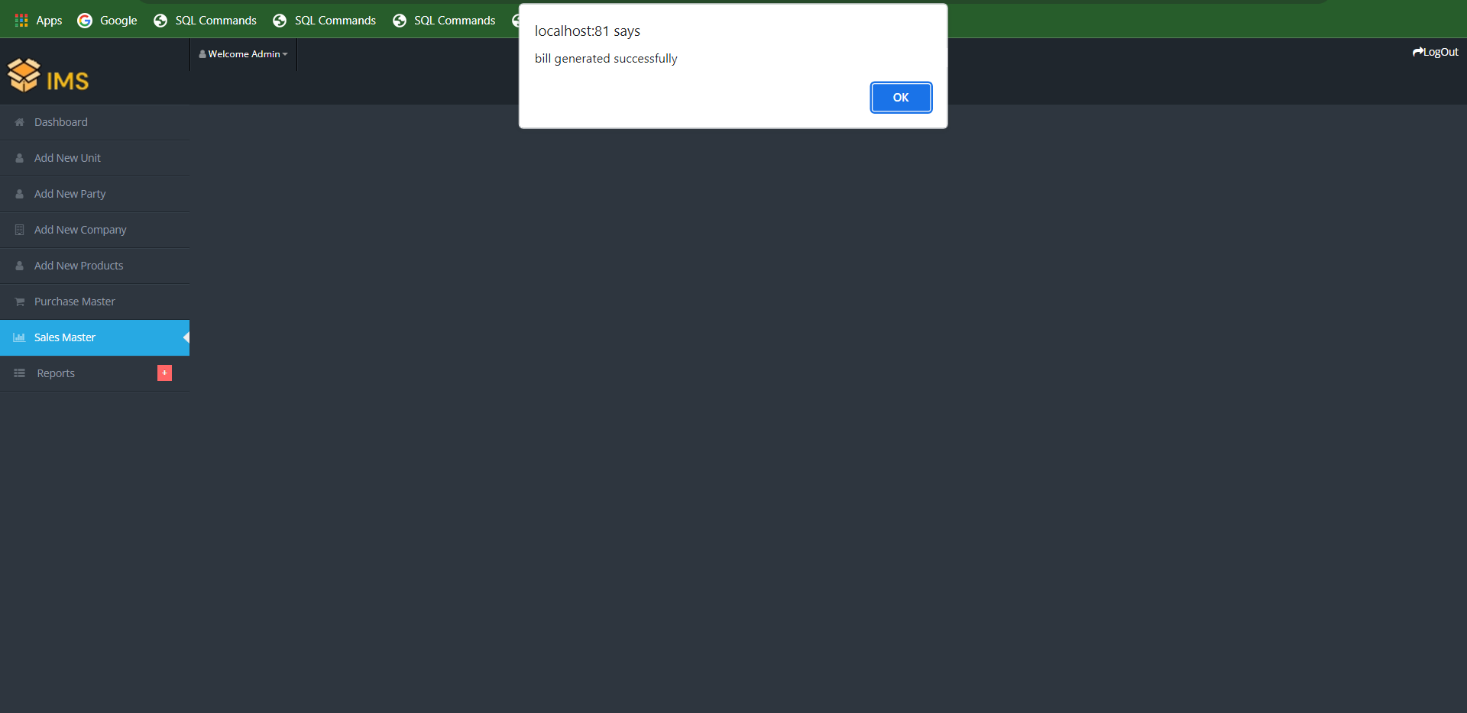
****

**Sales Master**

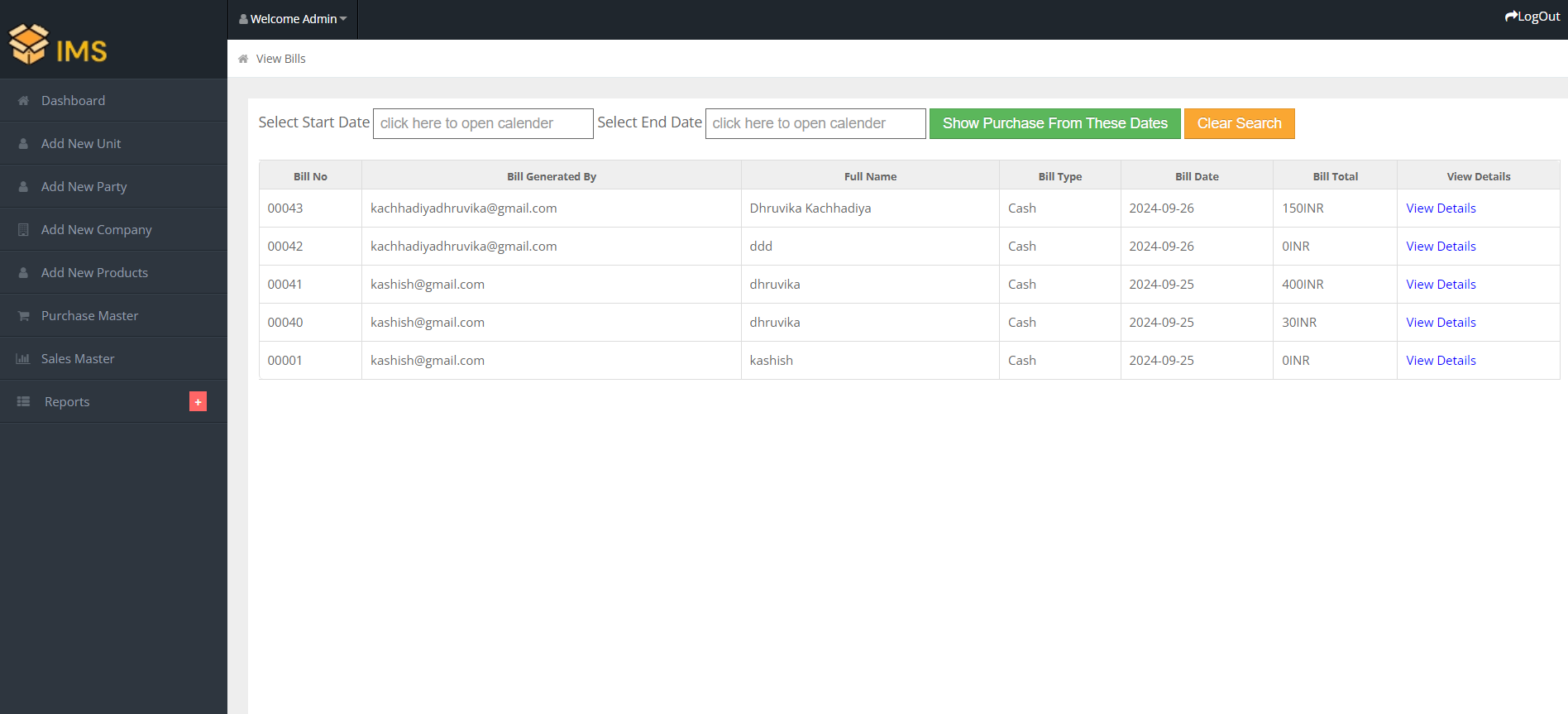




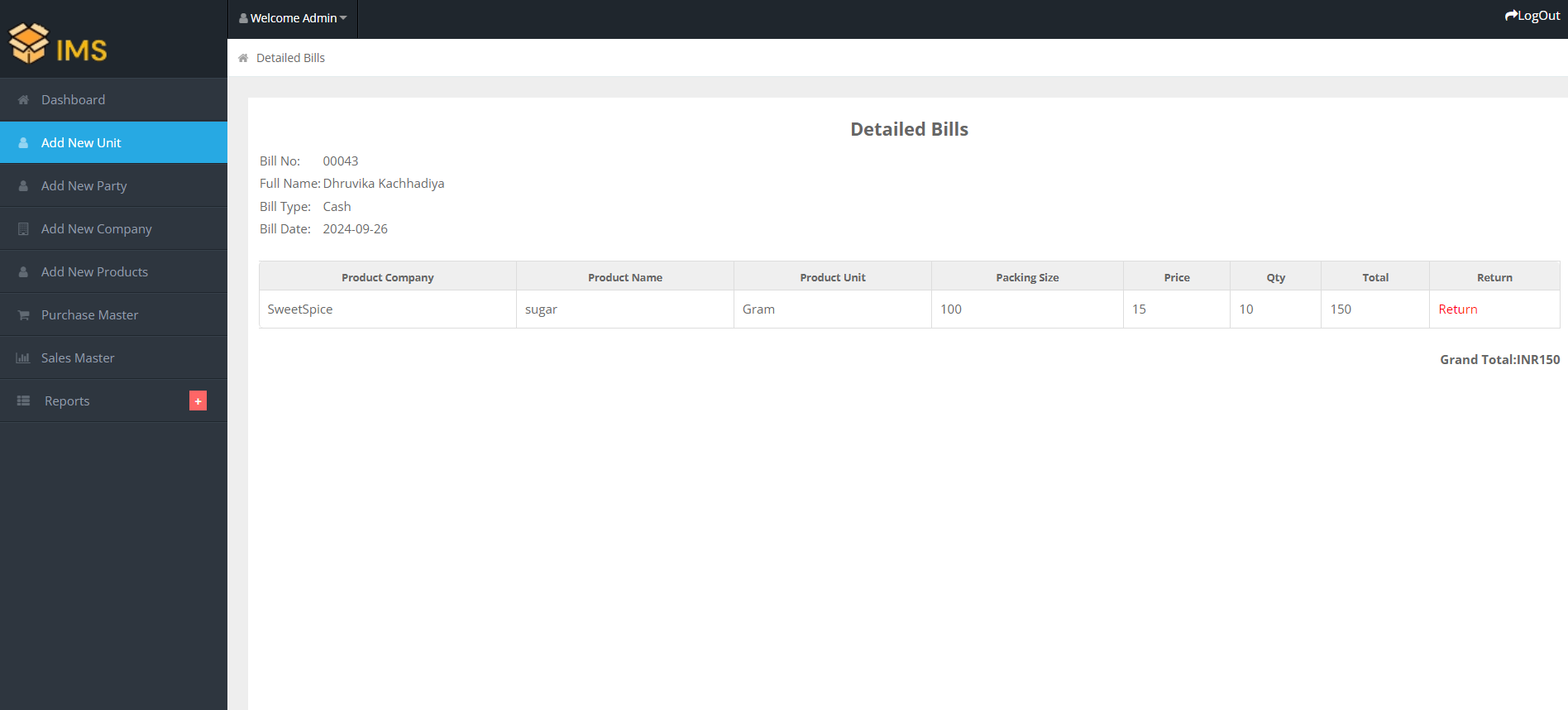




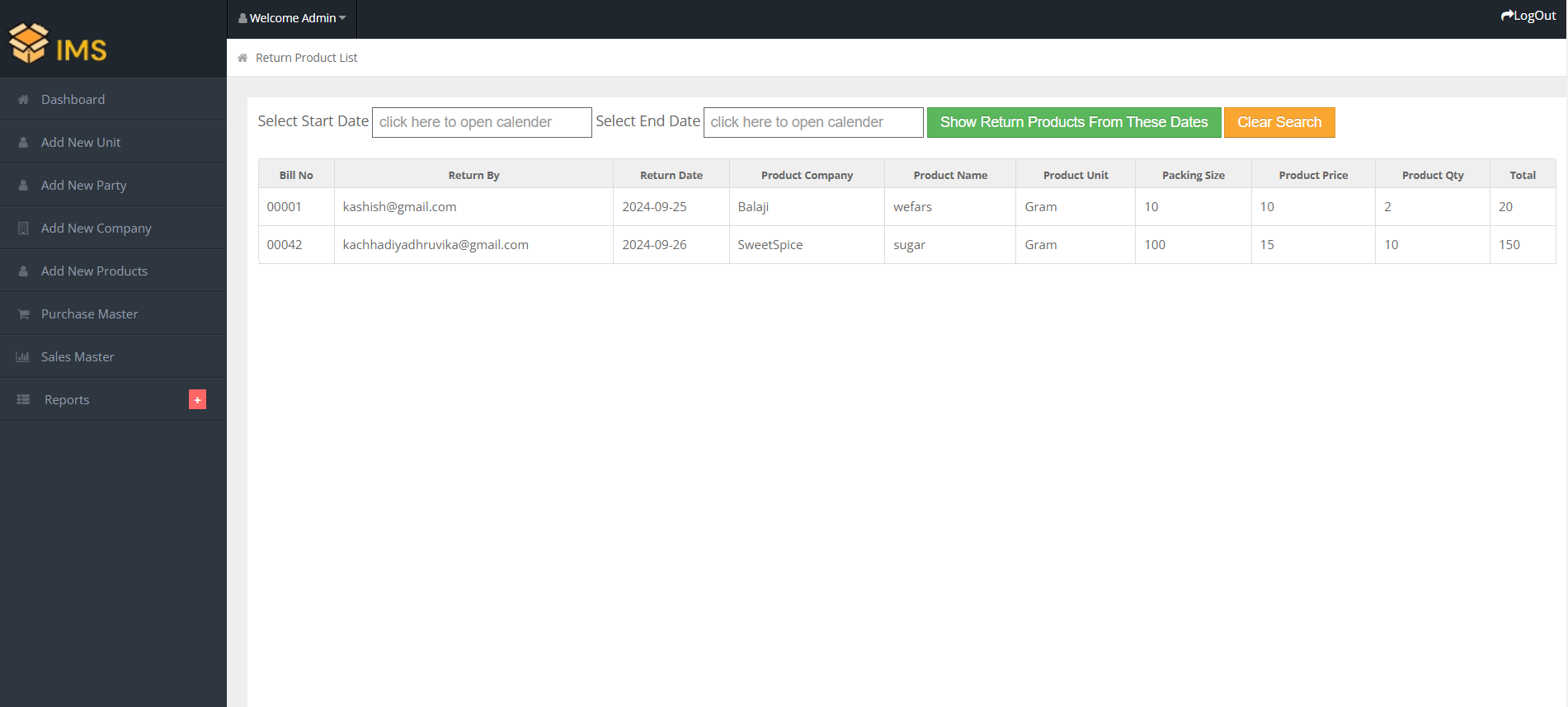
**Sales Report**

****

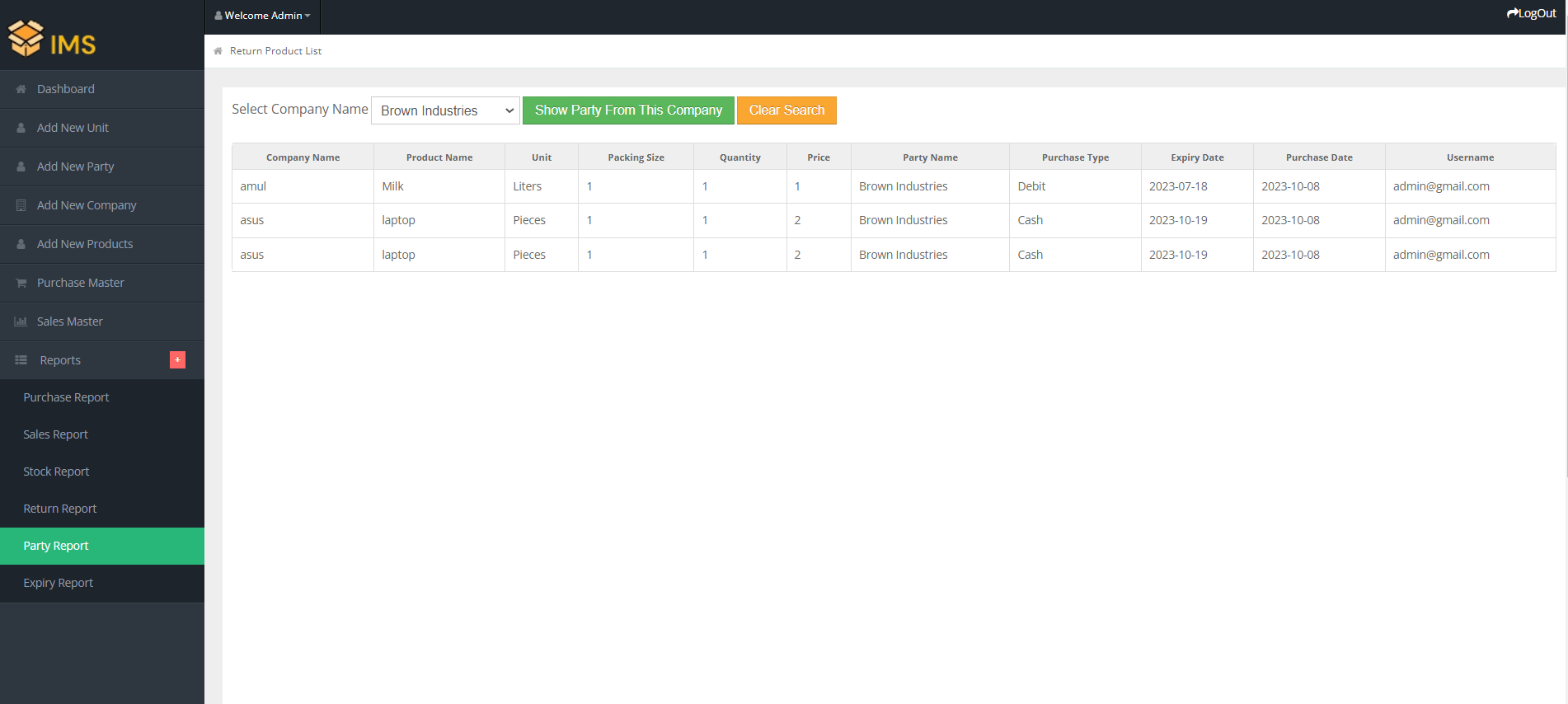
* View details

****

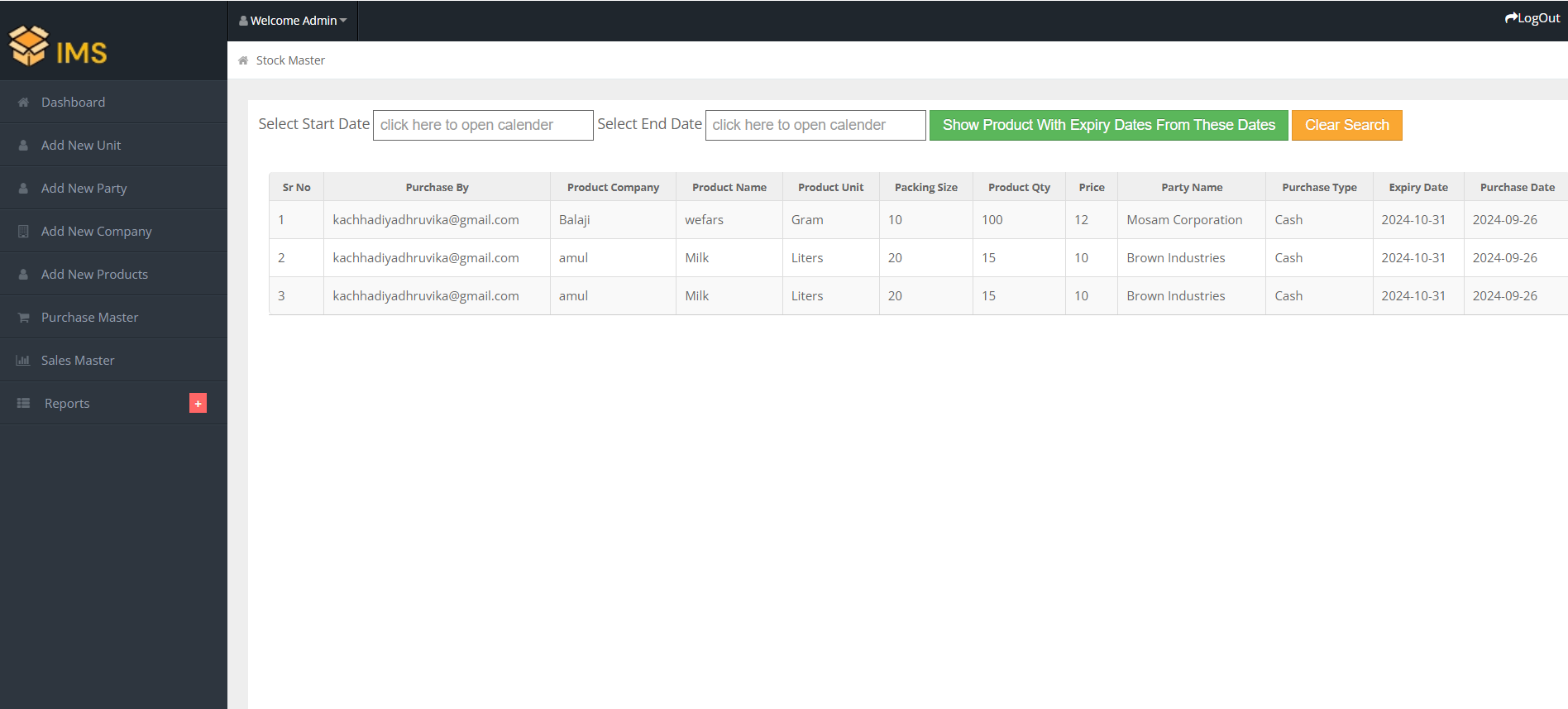
**Return Report**

****

**Party Report**

****

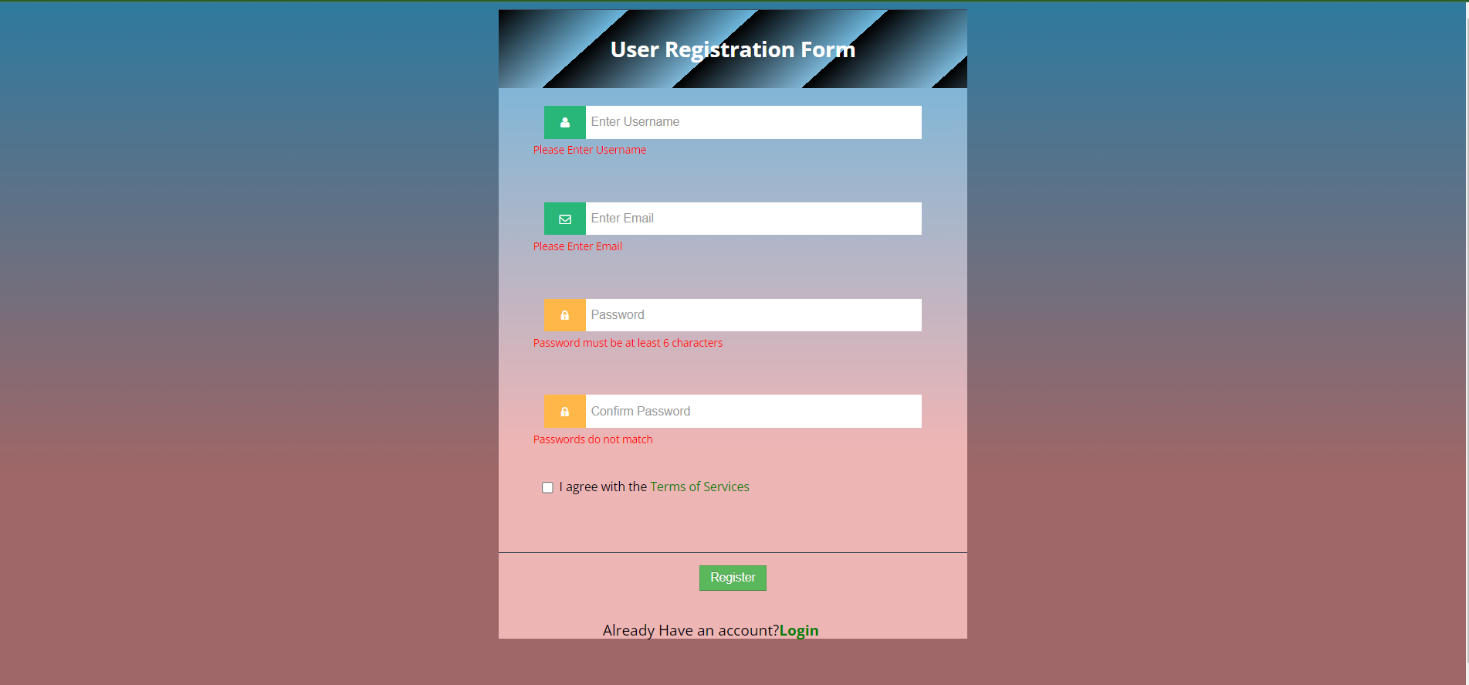
**Expiry Report**

****

**User Panel**

**Registration:**

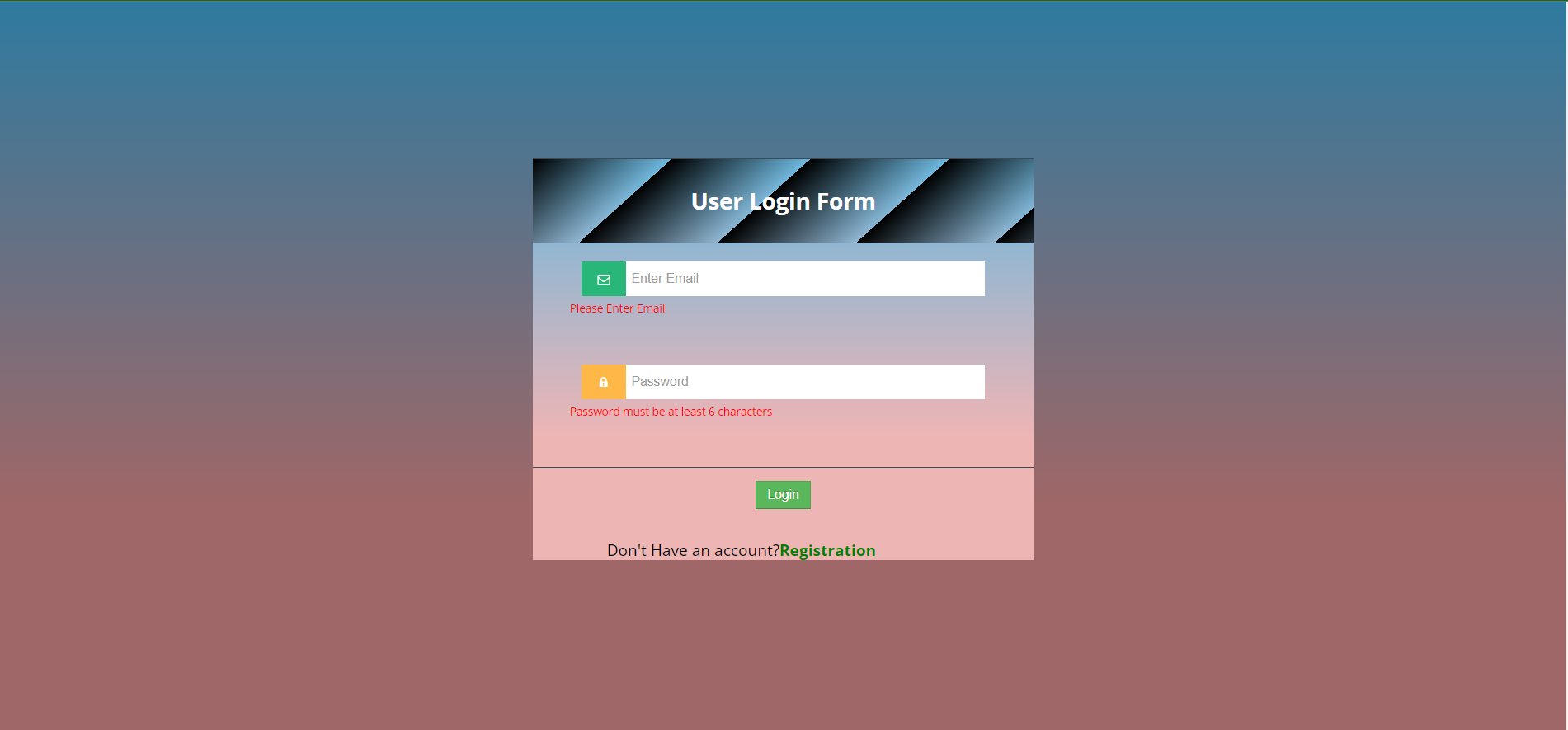
* validations

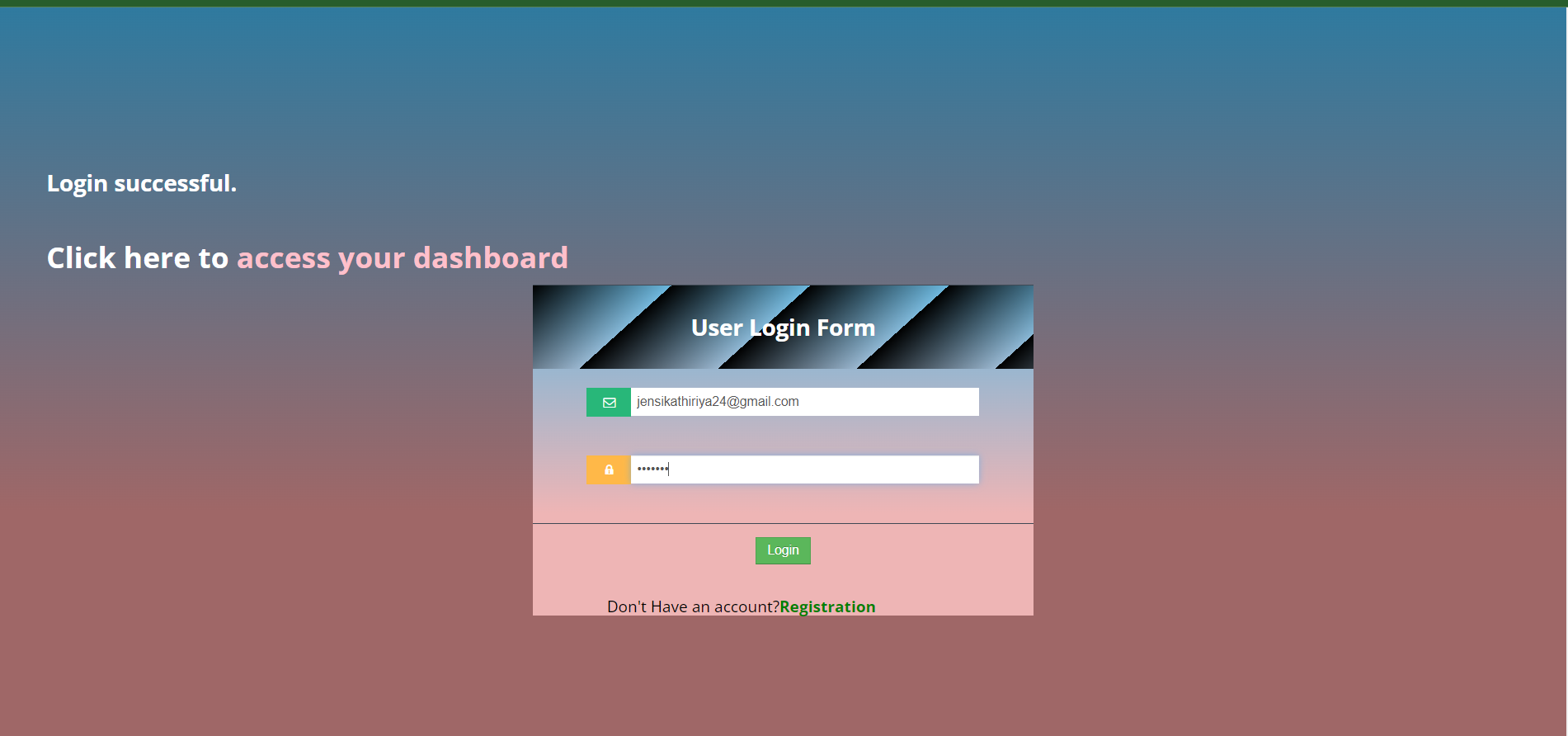
****

****

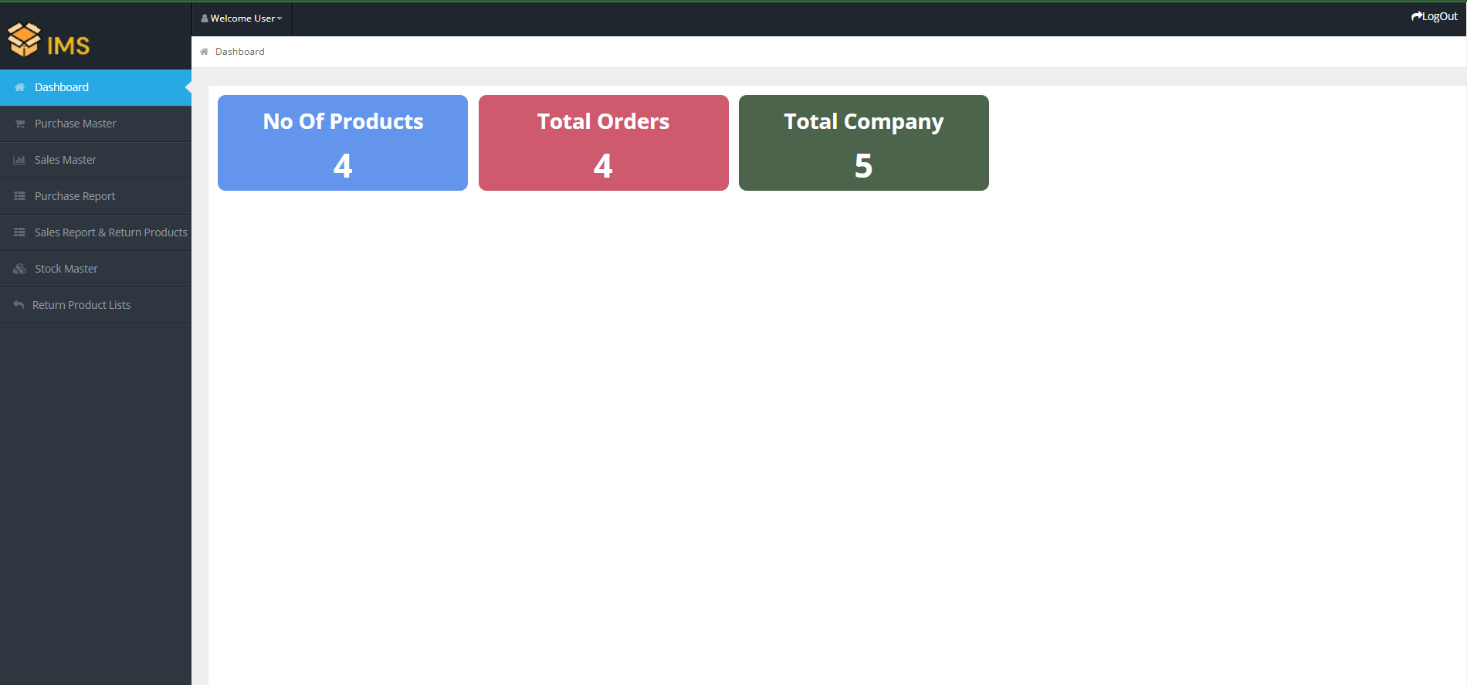
**Login**

* validations

****

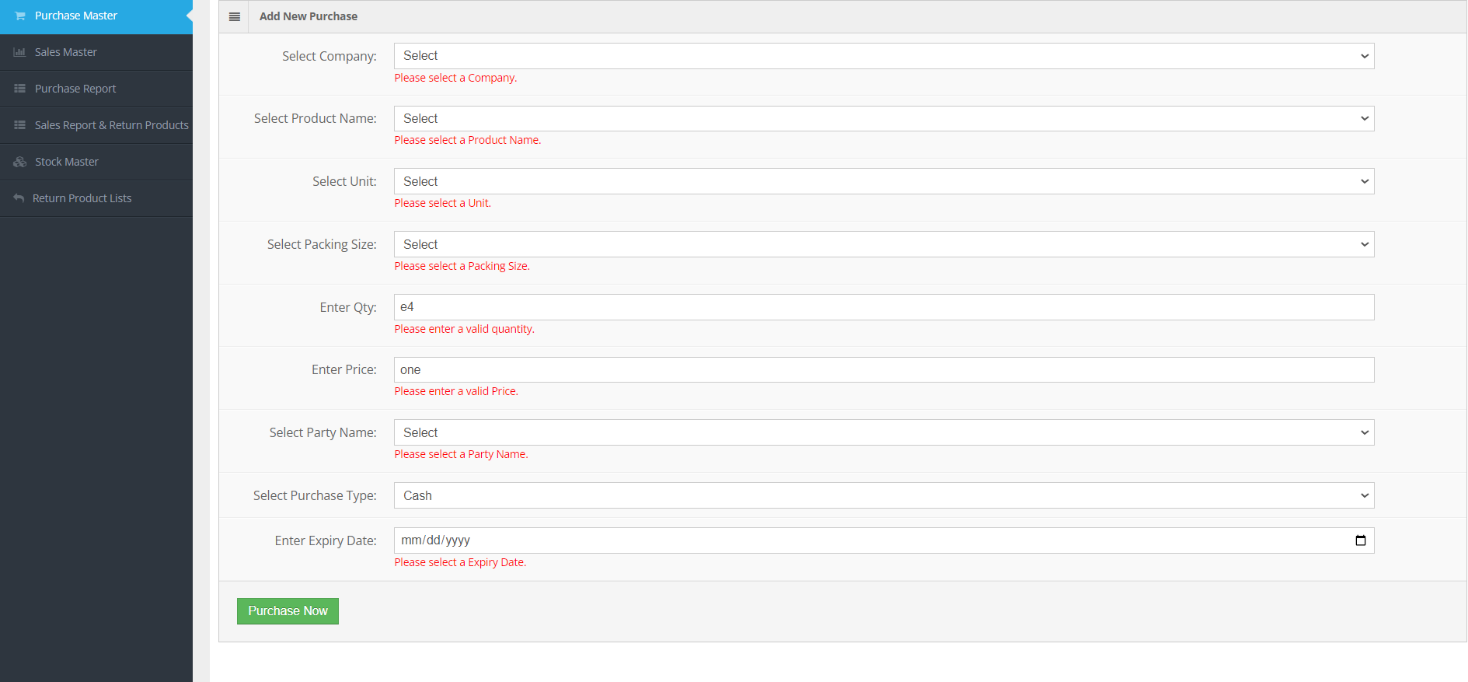
****

**Dahboard**

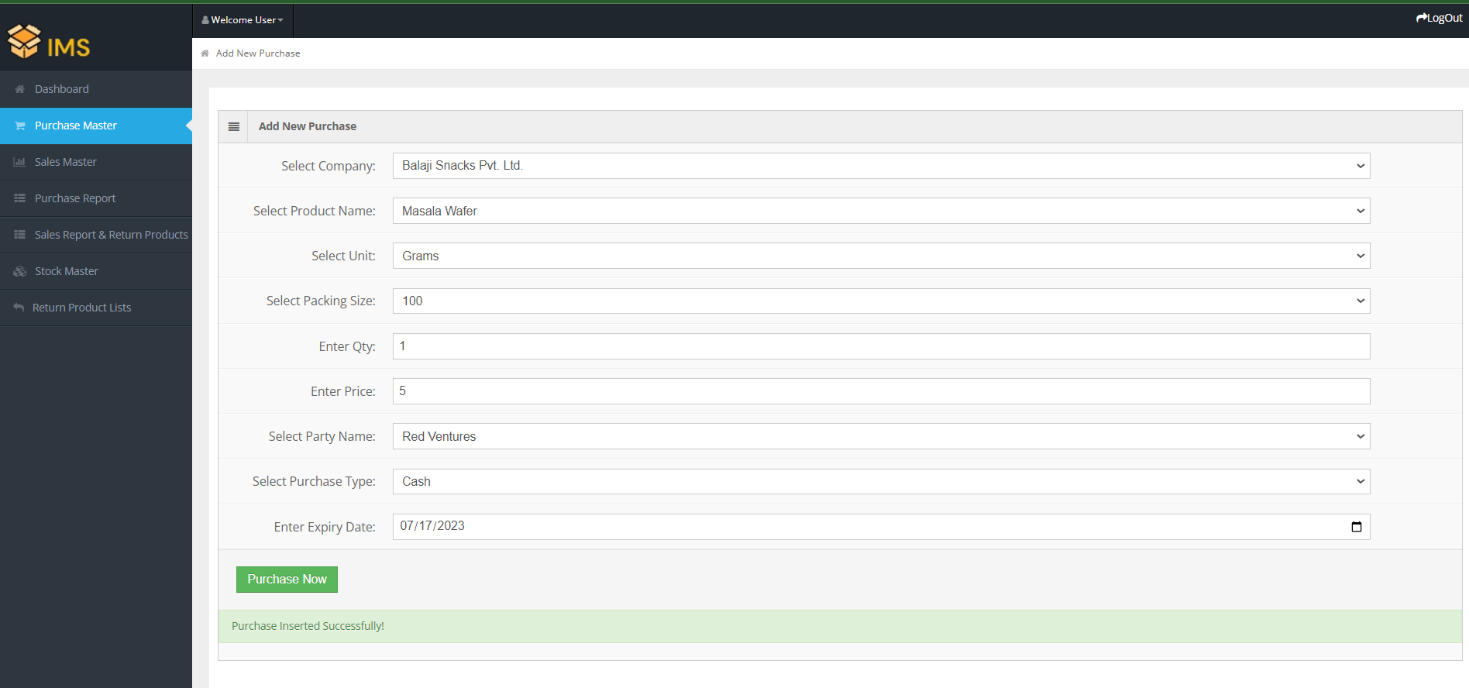
****

**Purchase Master:**

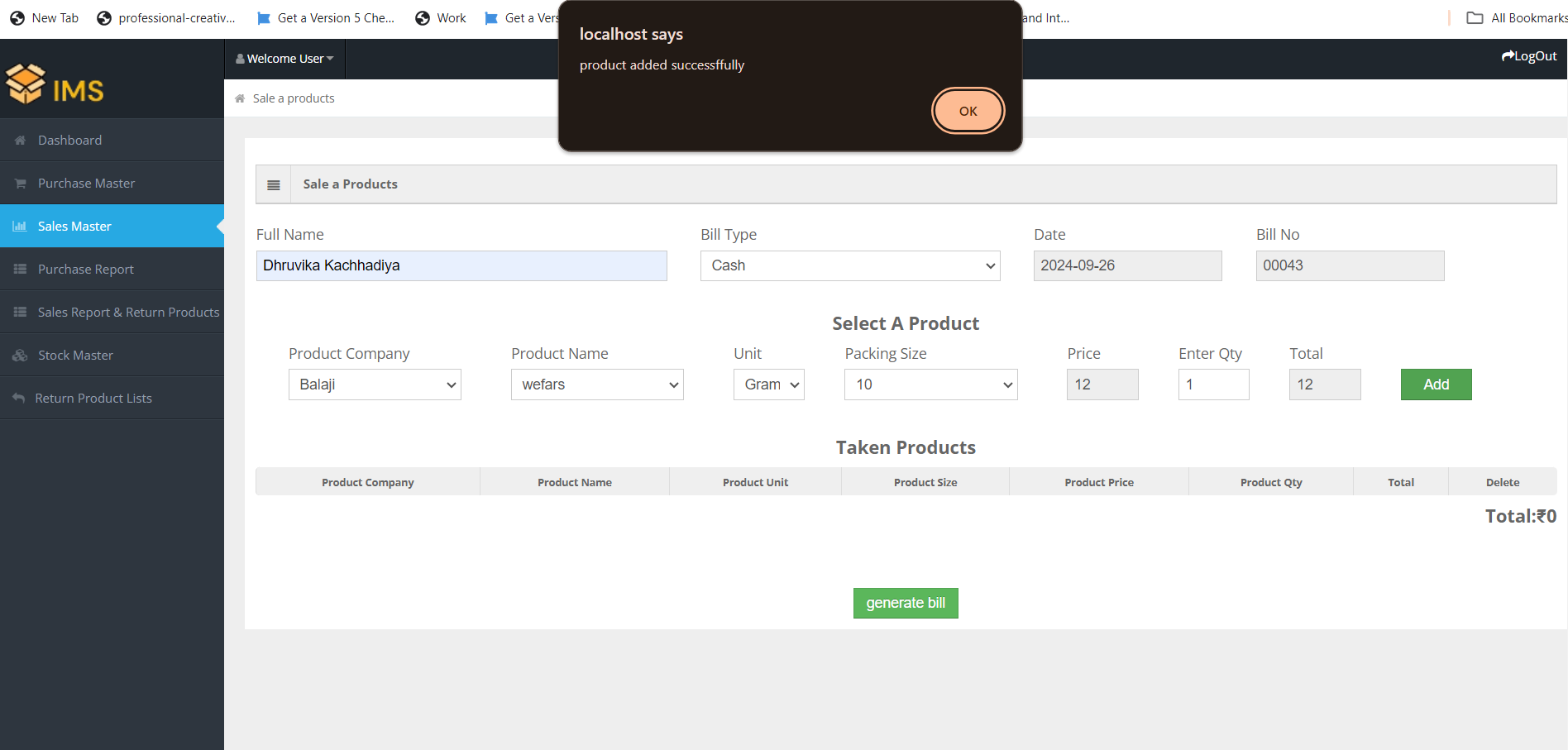
* validations

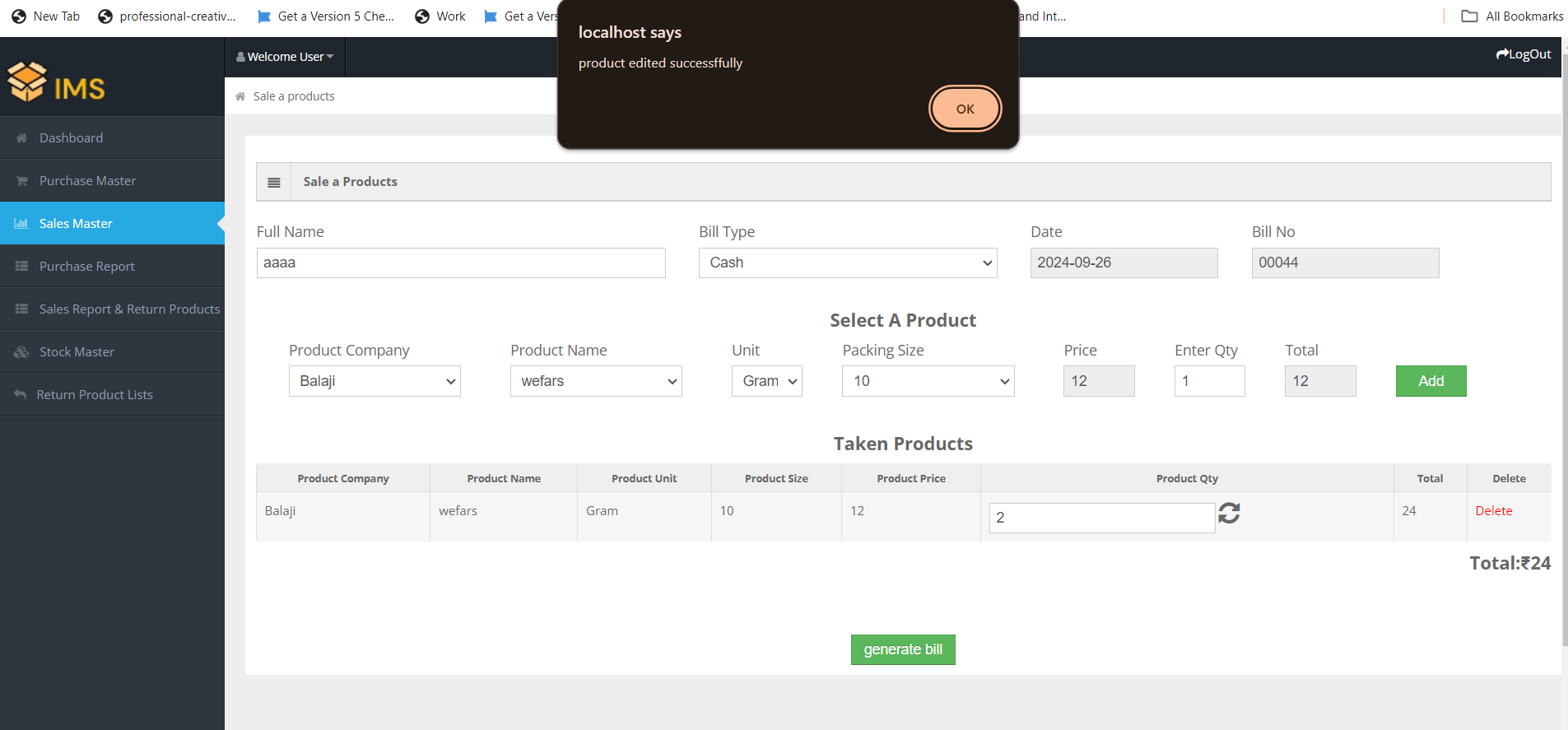
****

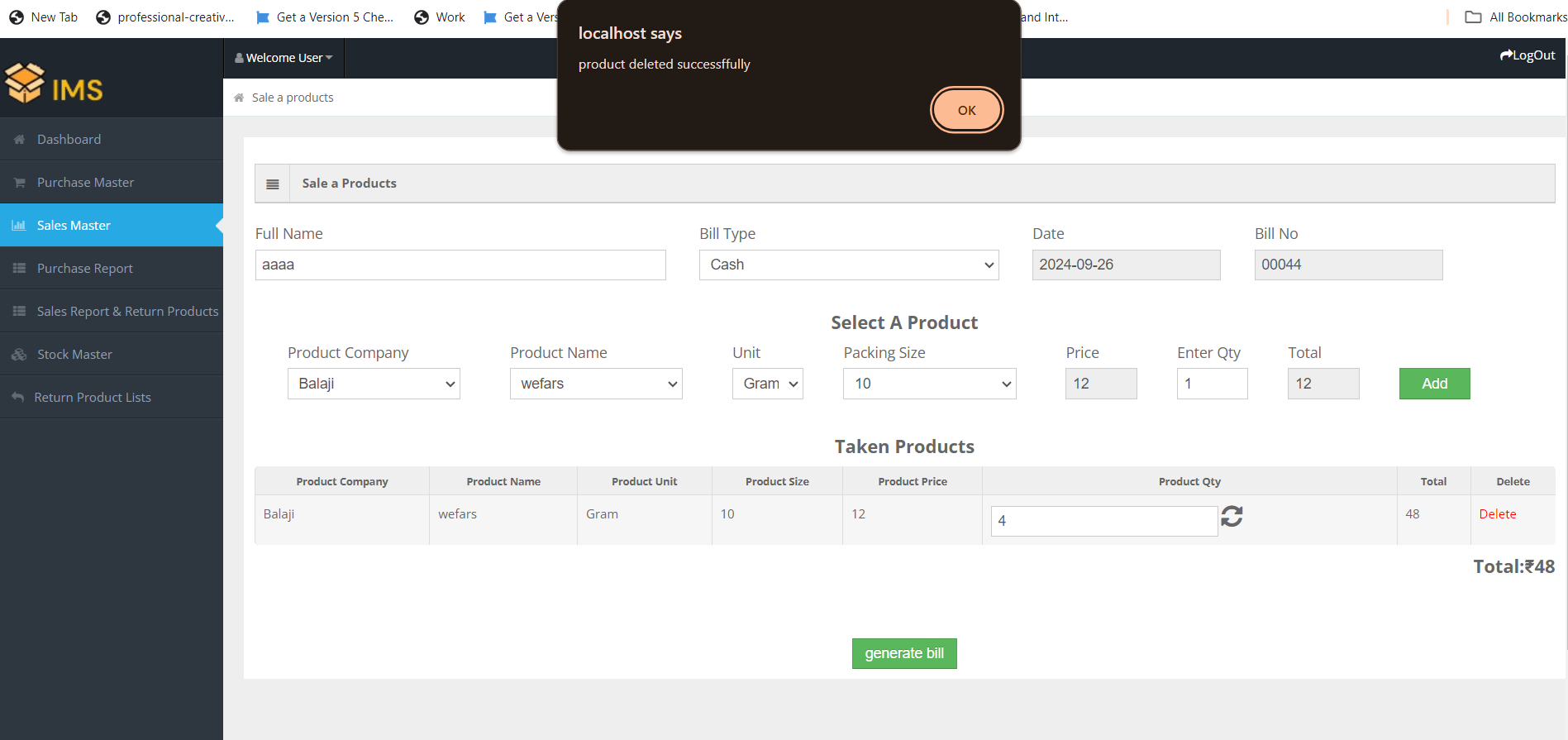
* Insert

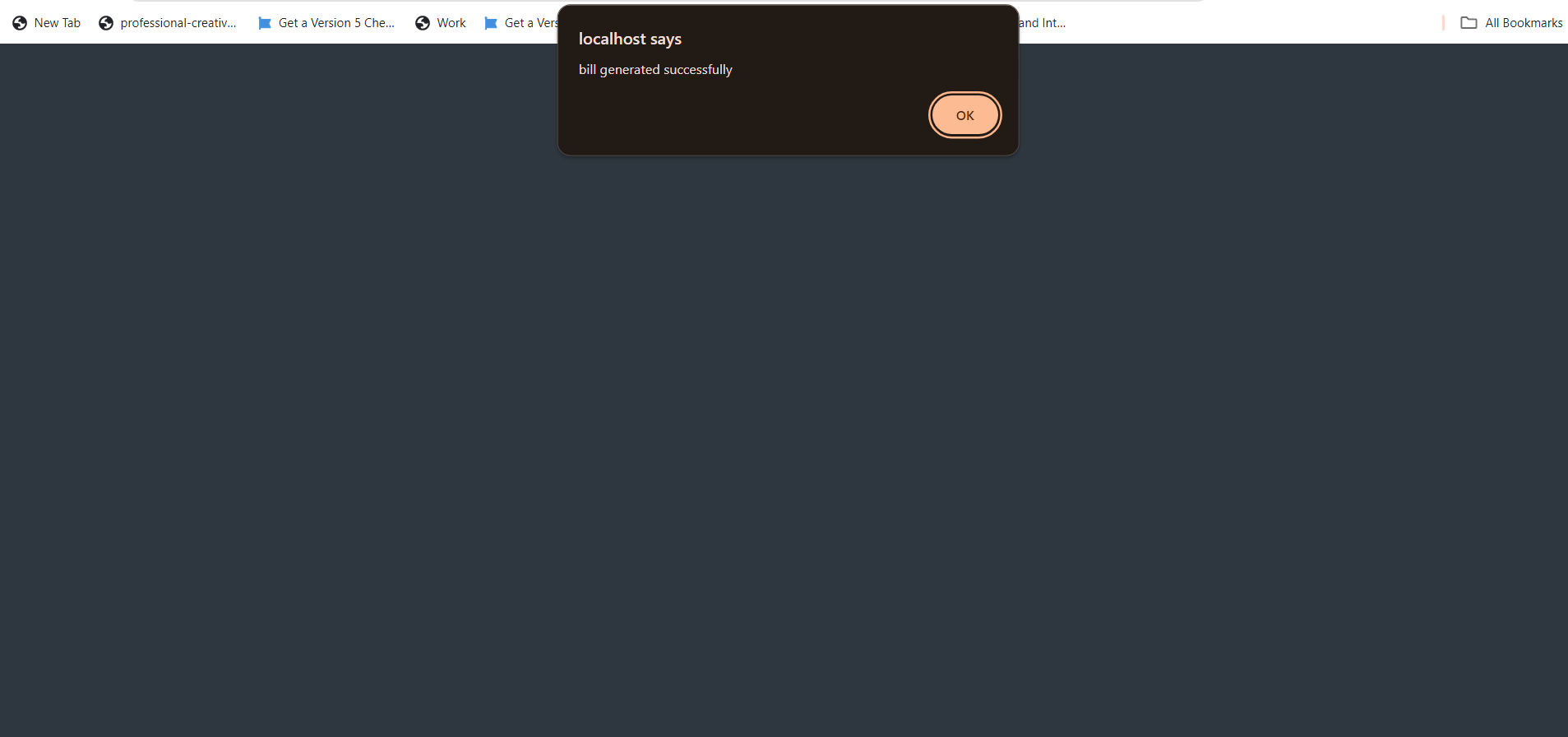
****

**Sales Master**

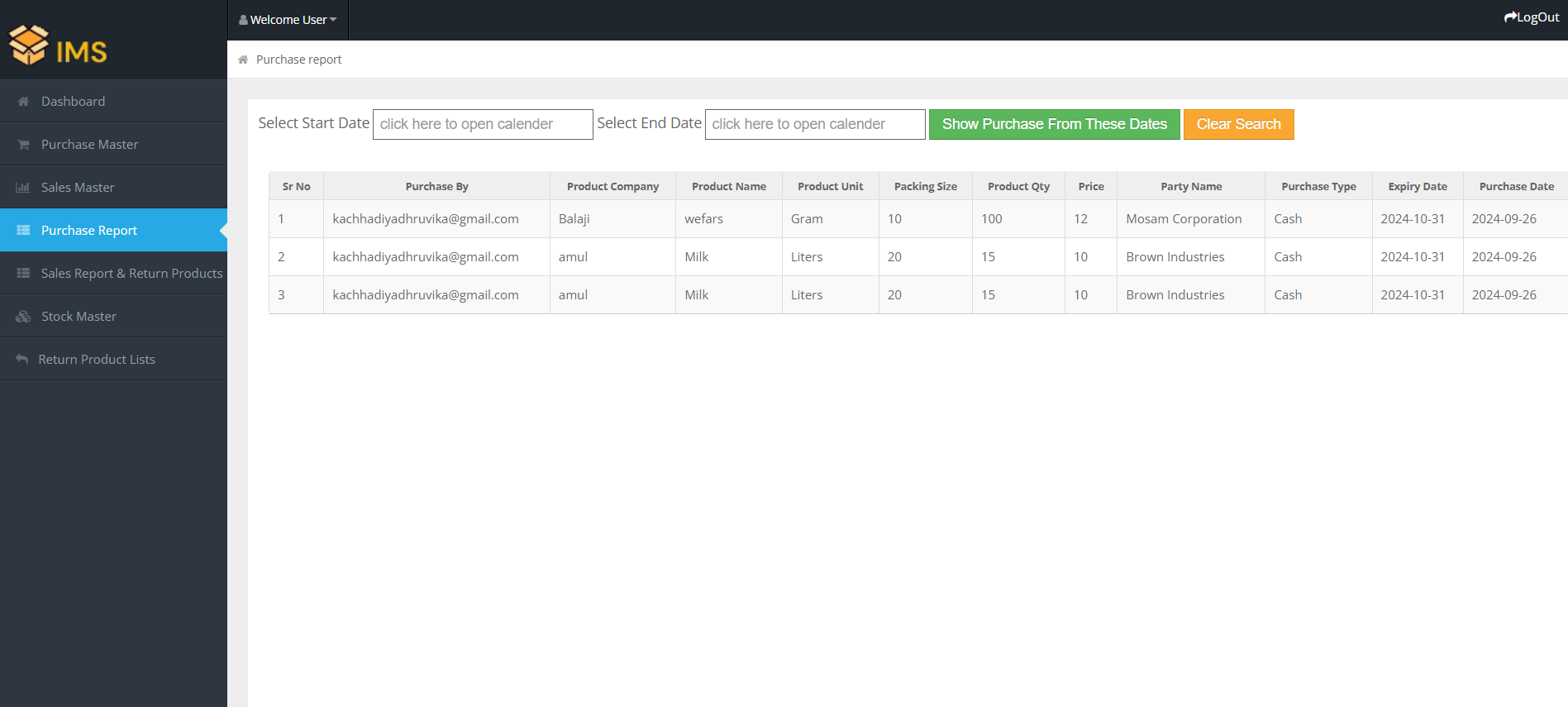
****

****

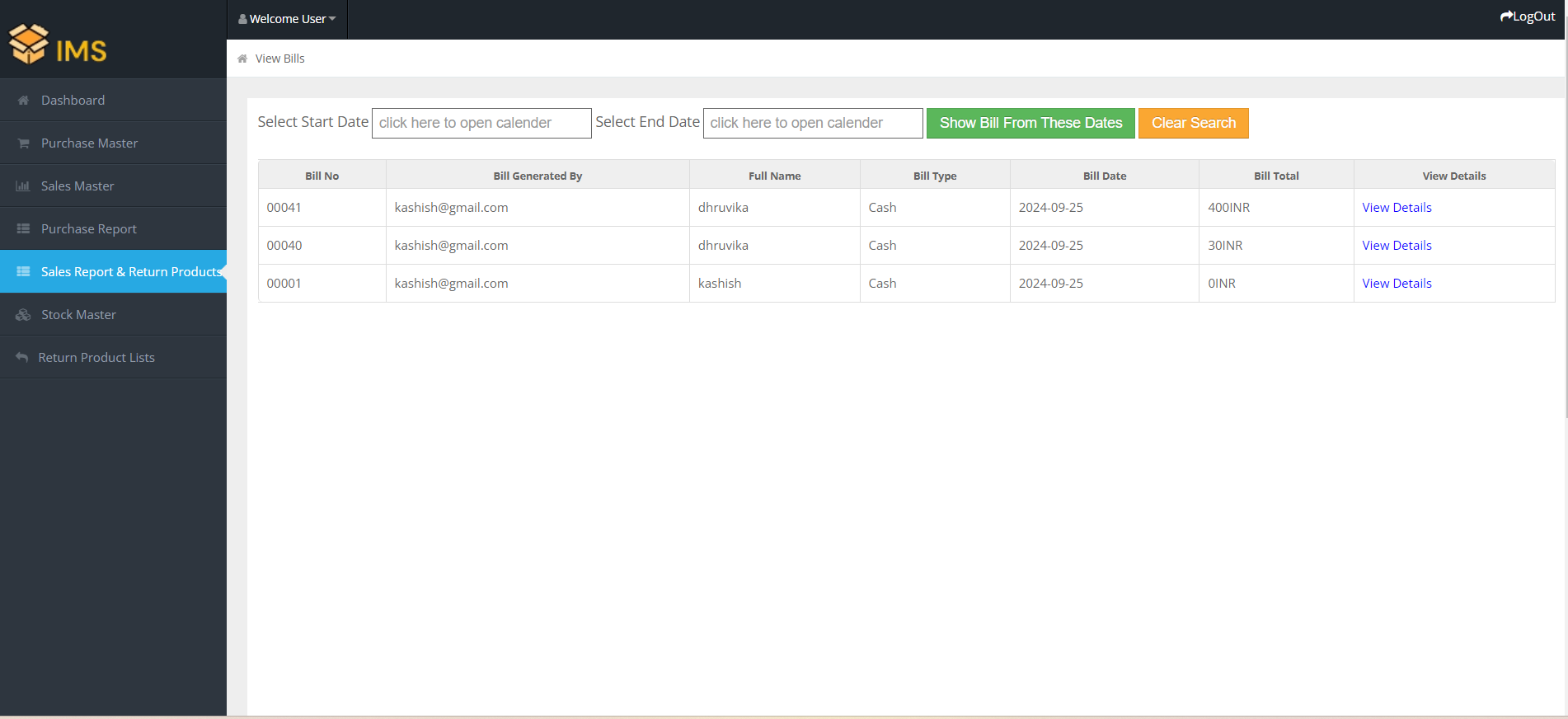
****

****

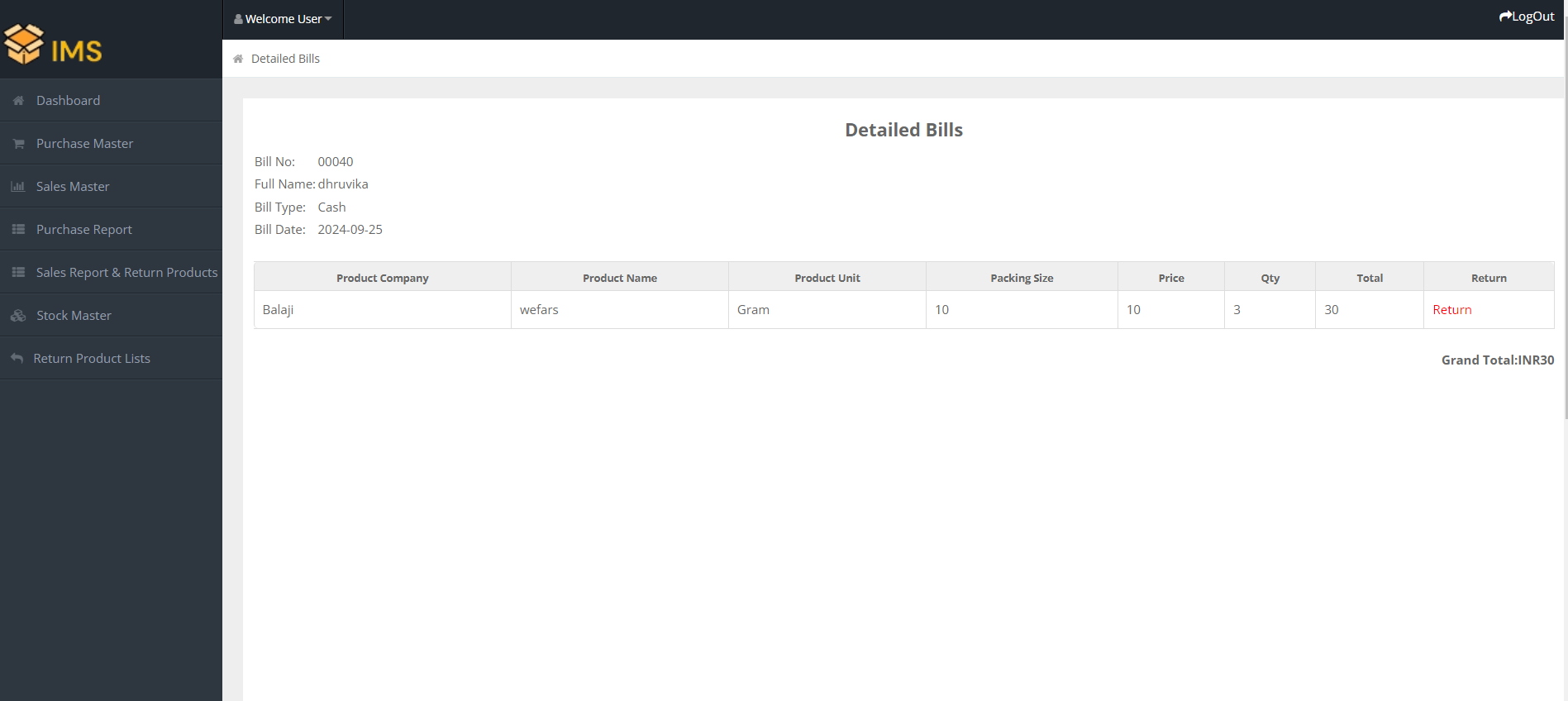
**Purchase Report**

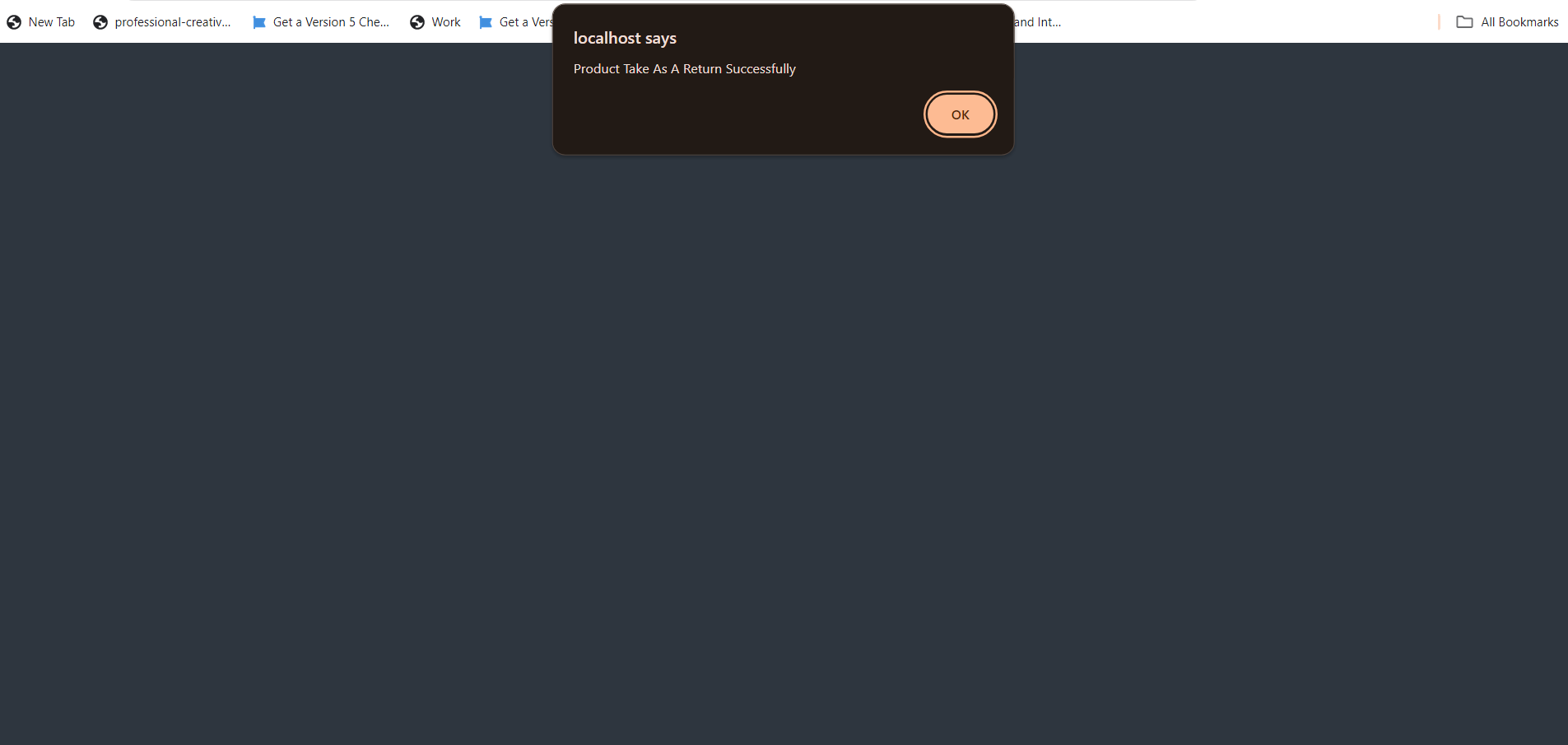
****

**Sales Report**

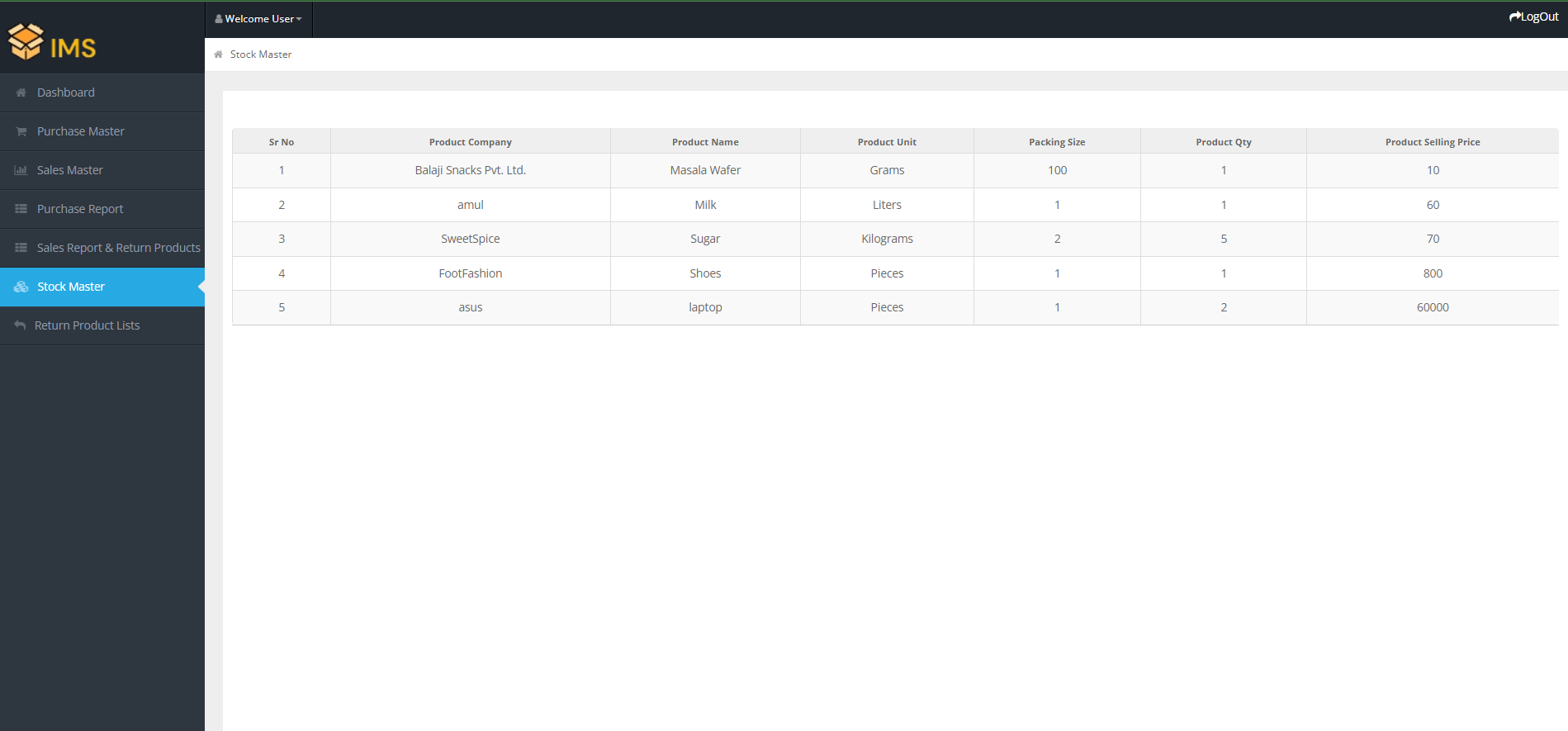
****

* View details

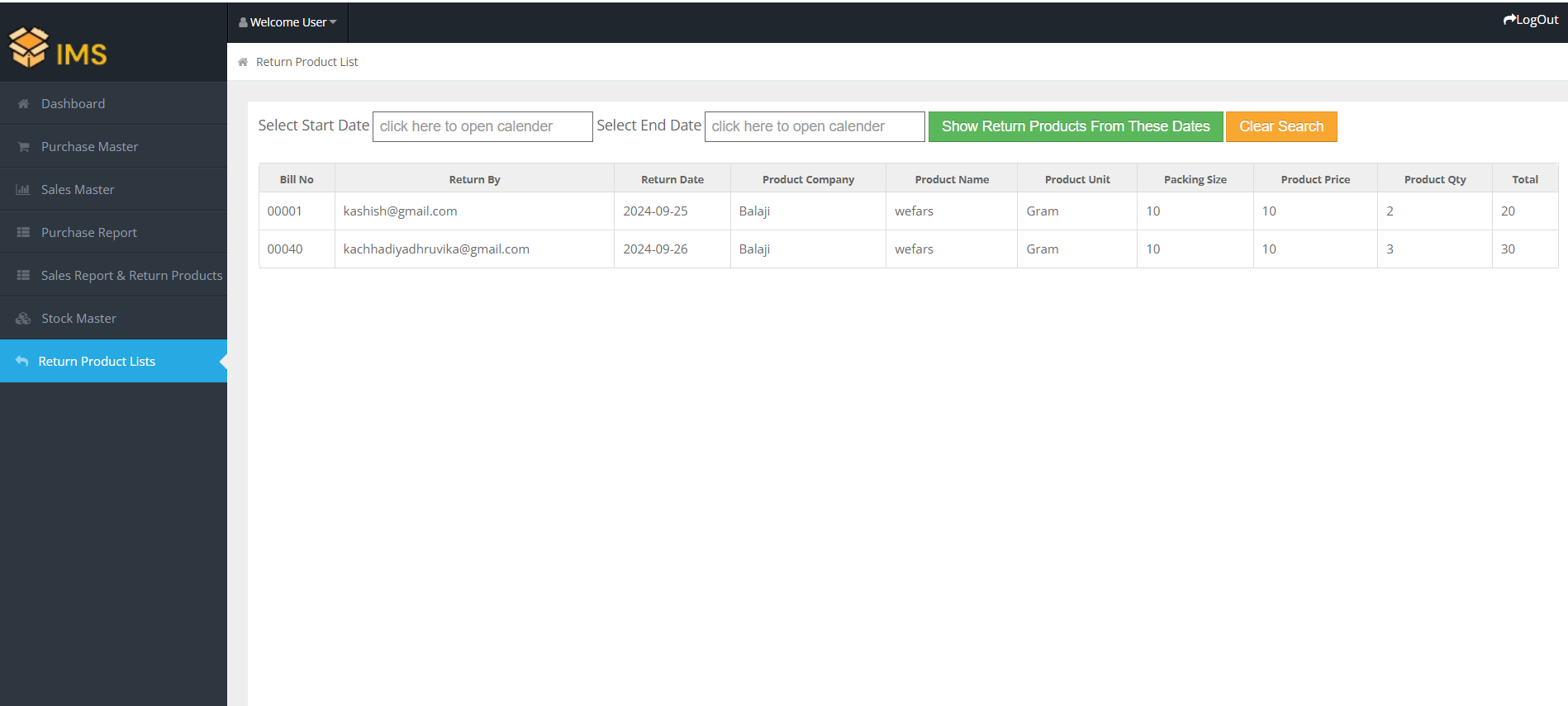
****

****

**Stock Master**

****

**Return Product Lists**

****

**8. SOFTWARE TESTING**

* **Black-Box Testing Requirements:**

1. **Functional Testing**:
   * Validate all admin and user functionalities, such as adding/editing/deleting units, purchase party info, products, and more.
   * Test user registration, login, and authentication processes.
   * Verify that items can be added to the system, sold, returned, and bills generated correctly.
2. **Usability Testing**:
   * Assess the user interface for its ease of use, clarity, and responsiveness.
   * Ensure that users can navigate through the system intuitively.
3. **Compatibility Testing**:
   * Check the compatibility of your system with various web browsers and devices to ensure it works consistently.
4. **Boundary Testing**:
   * Test the system's limits by checking how it handles extreme inputs or large quantities of data.
5. **Error Handling**:
   * Verify that the system handles errors gracefully, providing clear error messages to users when issues occur.
6. **Security Testing**:
   * Conduct security testing to identify and address vulnerabilities, including SQL injection and unauthorized access attempts.

* **White-Box Testing Requirements:**

1. **Code Review**:
   * Review the source code to ensure it follows best practices, is well-structured, and adheres to coding standards.
2. **Code Analysis Tools**:
   * Utilize code analysis tools to identify potential issues in the codebase.
3. **Unit Testing**:
   * Develop and execute unit tests for individual functions and methods to validate their correctness.
4. **Integration Testing**:
   * Test the interactions between different modules to ensure they work together seamlessly.
5. **Database Testing**:
   * Verify that the database schema, queries, and transactions function correctly and efficiently.
6. **Performance Testing**:
   * Evaluate system performance under various conditions, including load testing and stress testing.
7. **Regression Testing**:
   * Continuously retest existing features after updates to prevent new issues from emerging.
8. **Documentation**:
   * Document test cases, including input data, expected outcomes, and actual results for all testing phases. Also, maintain records of any discovered issues and their resolutions.
9. **LIMITATIONS AND FUTURE SCOPE OF ENHANCEMENT**

* **Limitations:**

1. **Limited Scalability**: Your system may have limitations on handling a massive number of products, users, or transactions. As the system grows, performance issues might arise.
2. **Security Concerns**: While efforts have been made to secure the system, there could still be potential security vulnerabilities that need to be addressed.
3. **Usability**: User interfaces can always be improved for better usability and user experience. Conducting usability testing can reveal areas for enhancement.
4. **Compatibility**: The system may not be fully compatible with all web browsers and devices. Compatibility testing may be needed to improve cross-browser support.
5. **Limited Reporting**: The system's reporting capabilities may be basic. Advanced reporting and data visualization features could enhance decision-making.
6. **Lack of Advanced Analytics**: The system may lack advanced analytical features such as predictive analytics or trend analysis, which can provide valuable insights.
7. **Mobile Accessibility**: While the system is accessible via web browsers, there may not be a dedicated mobile application. Developing a mobile app could improve accessibility.
8. **Data Backup and Recovery**: Robust data backup and recovery mechanisms may not be in place. Implementing a comprehensive data backup strategy is essential for data integrity.
9. **Internationalization**: The system may not support multiple languages or currencies. Expanding to international markets may require localization efforts.

* **Future Scope of Enhancement**:

1. **Enhanced Reporting**: Implement advanced reporting features, including customizable reports, graphical data representation, and export options (PDF, Excel).
2. **Predictive Analytics**: Incorporate predictive analytics to help users forecast demand, plan inventory, and make data-driven decisions.
3. **Mobile Application**: Develop a mobile app to provide users with on-the-go access to inventory management and sales features.
4. **Integration with E-commerce**: Integrate the system with e-commerce platforms to manage online sales seamlessly.
5. **Barcode/QR Code Scanning**: Implement barcode or QR code scanning for efficient product management and sales processes.
6. **AI and Machine Learning**: Utilize AI and machine learning to automate tasks like demand forecasting, suggesting reorder points, and identifying trends.
7. **Multi-Language Support**: Add support for multiple languages and currencies to cater to a global user base.
8. **Inventory Optimization**: Incorporate algorithms for inventory optimization to reduce carrying costs while ensuring product availability.
9. **Supplier Management**: Enhance the system to manage supplier relationships, track supplier performance, and streamline procurement processes.
10. **User Training and Documentation**: Develop comprehensive user training materials and documentation to assist users in maximizing the system's potential.
11. **Data Security Enhancements**: Continuously monitor and improve security measures to protect against emerging threats and vulnerabilities.
12. **User Feedback Mechanism**: Implement a feedback mechanism to collect user suggestions and requirements for ongoing improvements.
13. **Scalability**: Plan for scalability by optimizing database structures and code to handle larger datasets and user loads.
14. **AI-Driven Stock Recommendations**: Provide AI-driven recommendations for stock replenishment based on historical data and trends.
15. **IoT Integration**: Explore the integration of IoT devices for real-time monitoring of inventory levels and conditions.

**10. Reference**

* <https://www.youtube.com/@amitandipara11981>
* https://www.youtube.com/watch?v=r2N5PpVKkVQ&list=PLsdZGHZMYvk0oTVQ7L2X0t7xEEK92OR8V
* <https://chat.openai.com/>
* <https://lucid.app/>
* <https://www.visual-paradigm.com/>
* <https://www.freelogodesign.org/>