Stock Inventory Management System

By

Muntashir Arman Sameen

ID: CSE1801013014

Supervised by
Md. Ashfakur Rahman

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

May 2024

APPROVAL

The "**Stock Inventory Management System**" submitted by Muntashir Arman Sameen (CSE1801013014) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Md. Ashfakur Rahman
Lecturer
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Md. Ashfakur Rahman,** Lecturer, CSE**,** Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this **Inventory Management System** has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Md. Ashfakur Rahman)
Supervisor

Signature

Muntashir Arman Sameen
ID: CSE1801013014

ABSTRACT

Stock inventory management system is important for any business organization. The inventory management system (IMS) refers to the system and processes to manage the stock of an organization with the involvement of a technology system. This system can be used to store the details of the inventory, perform stock maintenance, update the inventory based on the sales details and generate sales and inventory reports yearly or weekly based. This project categorizes individual aspects of the sales and inventory management system. In this system, we are solving different problems affecting direct sales management and purchase management. An inventory management system is important to ensure quality control in businesses that handle transactions revolving around consumer goods. Without proper inventory control a large retail store may run out of stock on an important item. In the modern world every company wants to establish their business by profit and they can maintain their profit through software. Without proper inventory control, a large retail store may run out of stock on an important item. A good inventory management system will alert the wholesaler when it is time to record. Inventory management system is also an important means of automatically tracking large shipments. An automated inventory management system helps to minimize errors while recording the stock.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and strength to finish the task successfully within the scheduled time.

We are auspicious that we had the kind association as well as supervision of **Md. Ashfakur Rahman**, Lecturer, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Prof. Bulbul Ahamed,** Head of the Department, Computer Science and Engineering and **Brig. Gen. (Retd) Prof. Habibur Rahman Kamal, NDC, NDC, PSC,** Dean, Faculty of Science and Engineering, Sonargaon University, for their kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

CPU	Central Processing Unit
RAM	Random Access Memory
ROM	Read Only Memory

TABLE OF CONTENTS

STOCK INVENTORY MANAGEMENT SYSTEM Page No.

DECLARATION ……..…….. ii

ABSTRACT ……...……. iii

ACKNOWLEDGEMENT ……..…… vi

LIST OF ABBREVIATION ……..……. v

CHAPTER 1 1 – 2

INTRODUCTION

1.1 Introduction ---------- 1

1.2 Objectives ---------- 1

1.3 Motivation ---------- 1

1.4 Separate Category---------- 1

1.5 Tools ---------- 2

1.6 Methodology Methodology ---------- 2

1.7 Expected Outcome ---------- 2

CHAPTER 2 3 – 7

PROJECT SPECIFICATION

2.1 About Project ---------- 3

2.2 Feasibility Study ---------- 3

2.3 System Design ---------- 4

2.4 Requirement ---------- 4

2.5 Modeling ---------- 4 - 6

2.6 Challenges ---------- 6 - 7

CHAPTER 3 8 – 12

SYSTEM ANALYSIS AND DESIGN

3.1 Use Case Modeling And Description ---------- 8

3.2 Data flow Diagram ---------- 9

3.3 ER Diagram ---------- 10

3.4 Stock Inventory Management System Logo ---------- 11

3.5 Screenshot of Font-End ---------- 11 - 18

CHAPTER 4 20 - 21

IMPLEMENTATION AND TESTING

4.1 Implementation Requirements ---------- 20

4.2 Requirement Collection And Analysis ---------- 20

4.3 Design Analysis ---------- 20

4.4 Completion ---------- 20

4.5 Pass/Fail Criteria ---------- 21

CHAPTER 5 22 - 25

DEBUGGING AND TESTING

5.1 Purpose of Testing ---------- 22

5.2 Type of Testing ---------- 23

5.3 Units Test ---------- 24

CHAPTER 6 26 - 28

CONCLUSION AND FUTURE WORKS

6.1 Project Limitation ---------- 26

6.2 Conclusion ---------- 27

6.3 Lesson Learnt ---------- 28

REFERENCES ---------- 29

LIST OF FIGURES

Figure No.	Title	Page No.
Fig 3.1	Use Case Modeling	8
Fig 3.2	Data Flow Diagram	9
Fig 3.3	ER Diagram	10
Fig 3.4	Logo	11
Fig 3.5	Login Page	11
Fig 3.6	Dashboard Page	11
Fig 3.7	All Product	12
Fig 3.8	Category Product	12
Fig 3.9	All Adjustments	12
Fig 3.10	All Quotations	13
Fig 3.11	All Purchases	13
Fig 3.12	All Sells	13
Fig 3.13	POS	14
Fig 3.14	HRM Company	14
Fig 3.15	All Expenses	14
Fig 3.16	Customer Management	15
Fig 3.17	Suppliers Management	15
Fig 3.18	User Management	15
Fig 3.19	System Settings	16
Fig 3.20	POS Settings	16
Fig 3.21	Warehouse Page	16
Fig 3.22	Warehouse Report	17
Fig 3.23	Stock Report	17
Fig 3.24	Product Report	17
Fig 3.25	Sale Report	18
Fig 3.26	Best Customers	18
Fig 3.27	User Report	18
Fig 3.28	Dark Mode-1	19
Fig 3.29	Dark Mode-2	19
Fig 3.30	Dark Mode-3	19

CHAPTER 1

INTRODUCTION TO STOCK INVENTORY MANAGEMENT SYSTEM

1.1 Introduction

An inventory control system contains a list of orders to be filled and then prompts workers to pick the necessary items, and provides them with packaging and shaping information, inventory control may be used to automate a sales order fulfillment process and also manage in and outward material of hardware. Automation is the replacement of human workers by technology. For optional sales and inventory management processes, robust functionality is needed for managing logistic facilities. Warehouse management functions for inventory control cover internal warehouse movements and storage and its support helps in the recording and tracking of materials based on both quantity and value. This application takes care of all supply orders reducing costs for warehousing, and transportation while improving customer service. It significantly improves inventory turns, optimizes the flow of goods, and shortens routes within warehouse and distribution centers. it also improves cash flow, visibility, and decision providing efficient execution of tasks using this fast and reliable method.

1.2 Objectives

To supply the required materials continuously. The main objective of inventory management is to maintain the required inventory to run the production and sales process smoothly. An inventory management system helps you to track and control the company's supply so that you can optimize your inventory and manage it without spending extra time and money. The most integral part of the inventory management system is to evaluate your business regularly to ensure your path toward success track.

1.3 Motivation

The main purpose of inventory management is help to businesses easily and efficiently manage the ordering, stocking, storing, and using of inventory. By effectively managing your inventory, you'll always know what items are in stock, how many of them there are, and where they are located.

1.4 Separate Category

Raw Materials - The raw material is purchased by any company for its production purpose to transform it into a finished good. Work-in-progress inventory - refers to the process of transformation of raw material into a finished product. Finished goods - these are the complete goods that are now ready to be available for sale. Maintenance, repair, operation (MRO) goods -

items used for support of the production of finished goods as they will be purchased from the distributor for future resale.

1.5 Tools

1.HTML

2.CSS

3. Javascript

4. Bootstrap-5

5. Vue.js

6. PHP-8.2

7.MySQL

8. Ajax

9. Laravel-11

1.6 Methodology

Methodology is a collection of procedures, techniques, tools, and documentation that helps us in the system development process. To complete this project in time, we wanted to meet the goal in such a way.

1.7 Expected Outcome

We are expecting the outcome of our project will be according to our plan and it is going smoothly so far. We think that the way we are progressing the outcome will be satisfactory and users will get a platform where someone can manage inventory management-related issues using this management system.

CHAPTER 2

PROJECT SPECIFICATION

2.1 About Project

The main objective of our project is to maintain information about students, teachers, and other activities like attendance, student marks, class schedules, etc.

2.2 Feasibility Study

Market research studies are one of the most important sections of the feasibility study as they examine the marketability of the product or service and convince readers that there is a potential market for the product or service. The five principle areas of feasibility are technical (or technological), economic, legal, operational (or organizational) and scheduling, often denoted by the acronym Telos.

2.2.1. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building currently, this project is not technically feasible.

2.2.2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility— helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

2.2.3. Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That saved considerable time and effort by learning that their project was not feasible right from the beginning.

2.2.4. Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether—and how well —the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

2.2.5. Scheduling Feasibility

This assessment is the most important for project success after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

2.3 System Design

Systems design is the process of defining elements of a system like modules, architecture, components, and their interfaces and data for a system based on the specified requirements.

1. Create a Login Using HTML And CSS
2. Make a Website Homepage Using HTML and CSS
3. Admin Dashboard Sidebar Menu Using HTML And CSS
4. Connect to MySQL Database and Login Logout System Using PHP
5. User Registration From, Insert Data into MySQL Using PHP PDO+ Password Hashing
6. Fetch Data from MySQL Database and Display In HTML Table Using PHP And PDO
7. Delect Data from MySQL Database with Ajax, JavaScript and PHP
8. Update data from MySQL Database With Ajax/Javascript, PHP, Bootstrap Dialog

2.4 Requirement

Hardware Requirements: One PC/Laptop Web Browser Like Google Chrome, Mozilla Firefox, Opera Mini, Xampp, Composer

2.5 Modelling

Three of the most popular inventory management models are Economic Order Quantity (EOQ), Inventory Production Quantity, and ABC Analysis. Each of the inventory management models has an alternate way to deal with assisting you in knowing how much stock you ought to have available. Things you choose to utilize rely upon your business.

2.5.1 Economic Order Quantity (EOQ)

Best eCommerce traders and retail operators are driven by a prudent rationale to keep away from stockouts. Utilizing deterministic inventory management models – like the Economic Order Quantity (EOQ) model – will assist you with processing an ideal request

amount that limits stock expenses (and keeps away from the most dreaded part of e-commerce). The EOQ is a recipe that figures the most prudent number of things a business should request to limit costs and boost esteem when reloading stock. EOQ = 2DS/C D = Annual Demand C = Carrying Cost S = Ordering Cost In any case, the EOQ model expects a (consistent) request of an item and prompts accessibility of things to be restocked. It does not represent occasional or economic fluctuations. It expects fixed expenses of inventory units, requesting charges, and holding charges. No stockouts are allowed. This inventory model requires nonstop checking of inventory levels. The viability of the essential EOQ model is generally restricted by the presumption of a one-item business, and the recipe does not permit communication between items. Also, EOQ accepts a boundless arranging skyline and there is no restriction on capital availability

2.5.2 Inventory Production Quantity

Also called Economic Production Quantity, or EPQ, this kind of inventory management model discloses to you the number of items your business should arrange in a solitary cluster, to diminish holding expenses and arrangement costs. It expects that each request is conveyed by your provider in parts to your business, instead of in one full item. This model is an augmentation of the EOQ model. The distinction between the two models is the EOQ model expects providers are conveyed stock in full to your client or business. Here is how to calculate your Inventory Production Quantity: 2KDh(1-x) K = Setup (order) costs D = Demand Rate h = Yearly Holding Cost Per Product P = Yearly Production Rate x = D/P * Take the square root of (2SD)/Production Cost (1 – x) * S is your setup (order) costs * D is your demand rate (units) * X is your Demand Rate/Production Rate This model could be the best choice for your business if: * your business will generally request inventory from providers in parts as opposed to one full request, for example, for an automotive organization. * Interest for items is consistent over various periods.

2.5.3 Inventory Production Quantity

The more cash explicit inventory presents to you, the more significant it is to you. ABC examination arranges your inventory depending on degrees of significance. By realizing which stock is the main, you realize where to concentrate. To be best, ABC Analysis is often utilized with other inventory management models, like the Just in Time (JIT) technique. Category A: Inventory under this classification gets the most cash and is just a modest quantity of your all-out stock. It accounts for just 20% of your stock however gets 70% of complete income. Category A stock is given the most measure of consideration and has tight requesting controls set up. Category B: Unlike Category A stock, this Category B stock is not fundamental for your business to endure, however, it matters. It is 30% of your stock with 25% income. Category C: Inventory arranged under Category C

is 50% of your items with 5% income. This stock does not get as much benefit as A and B, however, it is steady. Inventory management is quite free here since it gets a modest quantity of pay. If your organization offers administrations or items that all shift significantly in cost, for example, an arranging organization, this will be a viable model for you. A genuine model is Amazon, whose items cover an enormous scope of costs. Few out of every odd thing you see recorded on their site is available. That would prompt extraordinarily high holding costs, making it hard to make money. So all things being equal they request stock dependent on what they see with their ABC classification rules. The drawback to this kind of inventory management model is that you need to classify the correct stock accurately with the goal for it to work. Or else you will be putting all your inventory management models into consideration for an item that is not presenting to you the most cash.

2.6 Challenges

1. Inaccurate Data: You need to know, at any given moment, exactly what inventory you have. Gone are the days when inventory could be counted once a year with an all-hands-on-deck approach.

2. Changing Demand: Customer demand is constantly shifting. Keeping too much could result in obsolete inventory you're unable to sell while keeping too little could leave you unable to fulfill customer orders. Order strategies for core items, as well as technology to create and execute an inventory plan, can help compensate for changing demand.

3. Limited Visibility: When your inventory is hard to identify or locate in the warehouse, it leads to incomplete, inaccurate, or delayed shipments. Receiving and finding the right stock is vital to efficient warehouse operations and positive customer experiences.

4. Problem Stock: Stock needs specialized plans for care and storage. And high-value inventory needs specific loss-prevention strategies and inventory controls.

5. Managing Warehouse Space: Efficiently managing space is an intimidating task. Planning and designing warehouse spaces with inventory management platforms helps you better control the timing of new stock deliveries. It can account for important factors, such as available space. Read more about the differences between warehouse management and inventory management.

6. Increasing Competition: Globalized supply chains are subject to unpredictable economic shifts and market forces that impact the competition for raw materials. Small

businesses are sometimes faced with choosing between competing for high-demand materials or holding enough inventory to control costs.

7. Overstocking: Keeping too much stock on hand can be as problematic as having too little. Overstock impacts business cash flow and leads to inventory-related problems, such as storage and loss.

8. Insufficient Order Management: One of the most common challenges to sound inventory management is preventing the overselling of products and running out of inventory. Using historical and seasonal data trends can help you accurately predict customer orders.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 Use Case Modeling And Description

A use case model describes the proposed functionality of a new system. A use case represents a discrete unit of interaction between a user (human or machine) and the system.

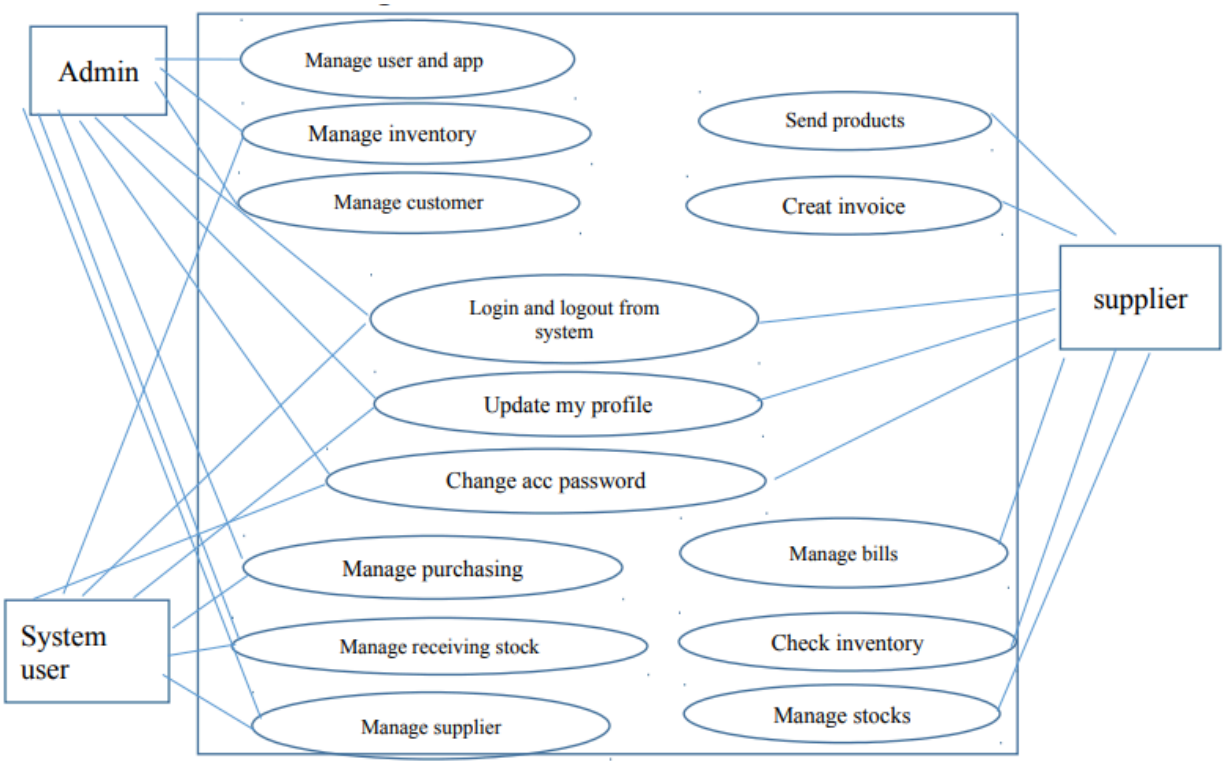


Fig 3.1: Use Case Modeling

3.2 Data Flow Diagram

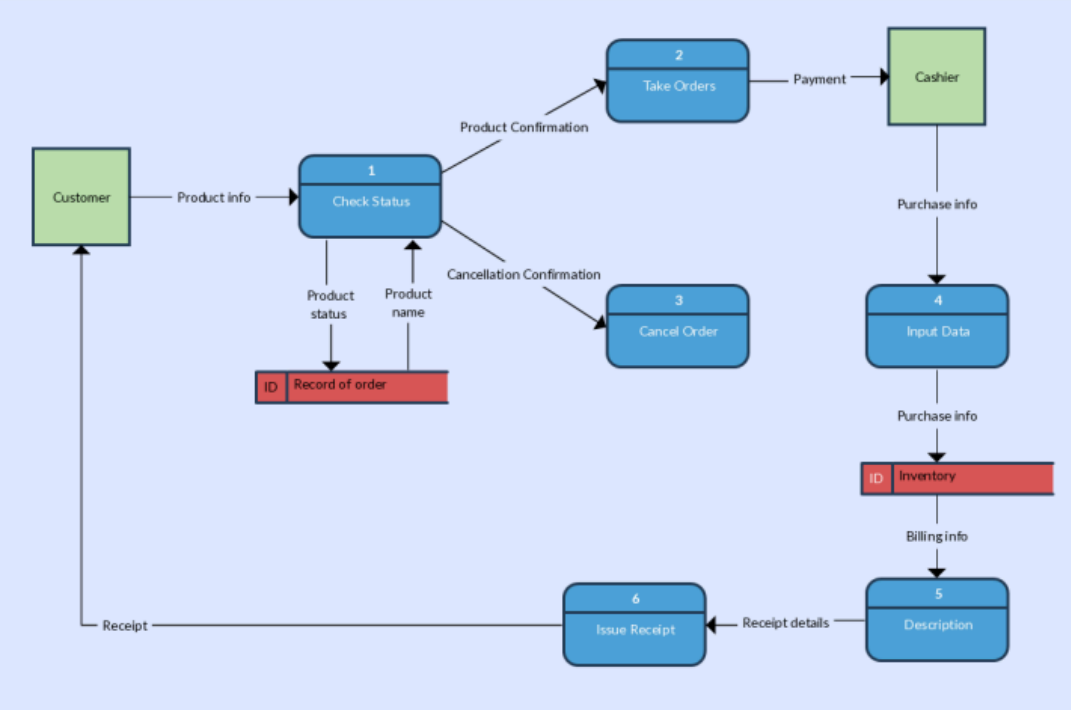


Fig 3.2: Data Flow Diagram

3.3 ER Diagram

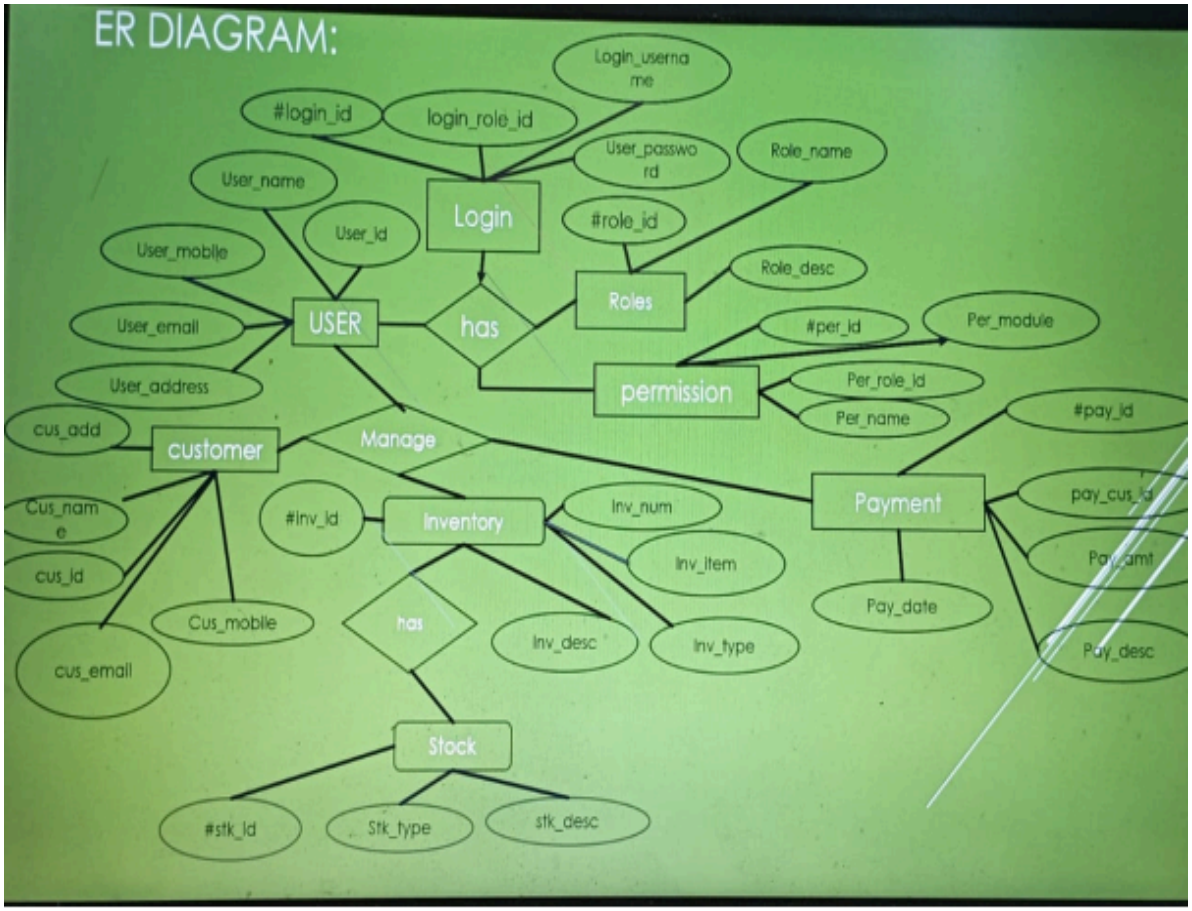


Fig 3.3: ER Diagram

3.4 Stock Inventory Management System Logo

Fig 3.4: Logo

3.5 Screenshot of Front End

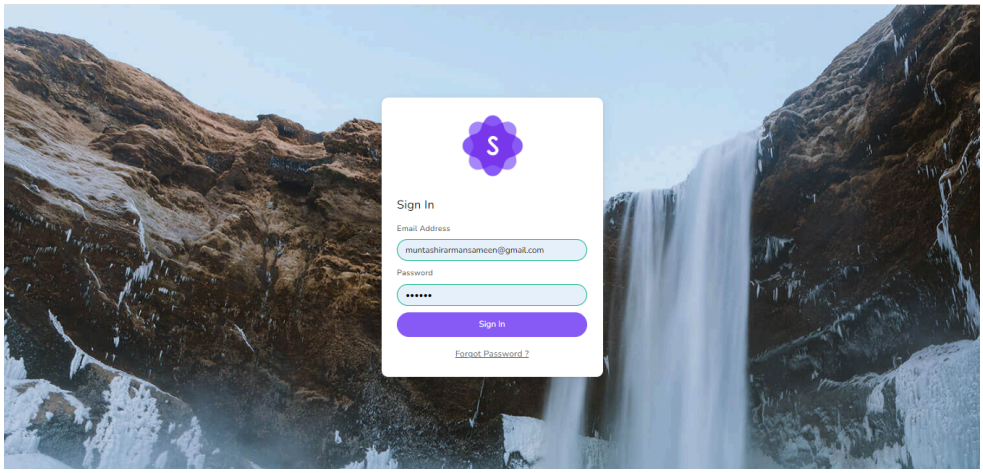


Fig 3.5: Login Page

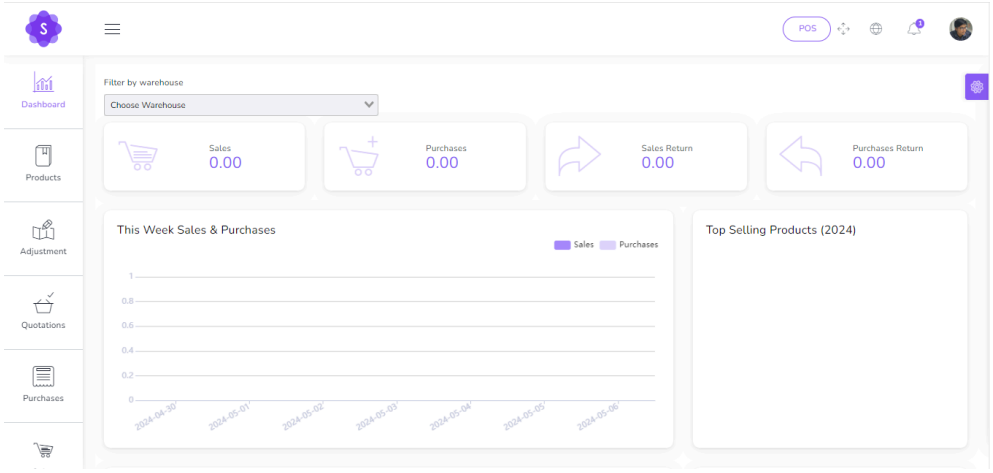


Fig 3.6: Dashboard Page

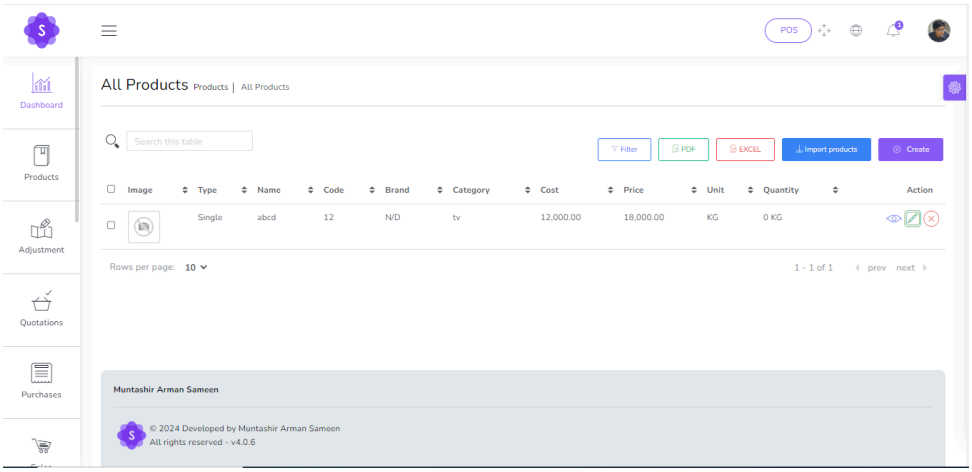


Fig 3.7: All Product

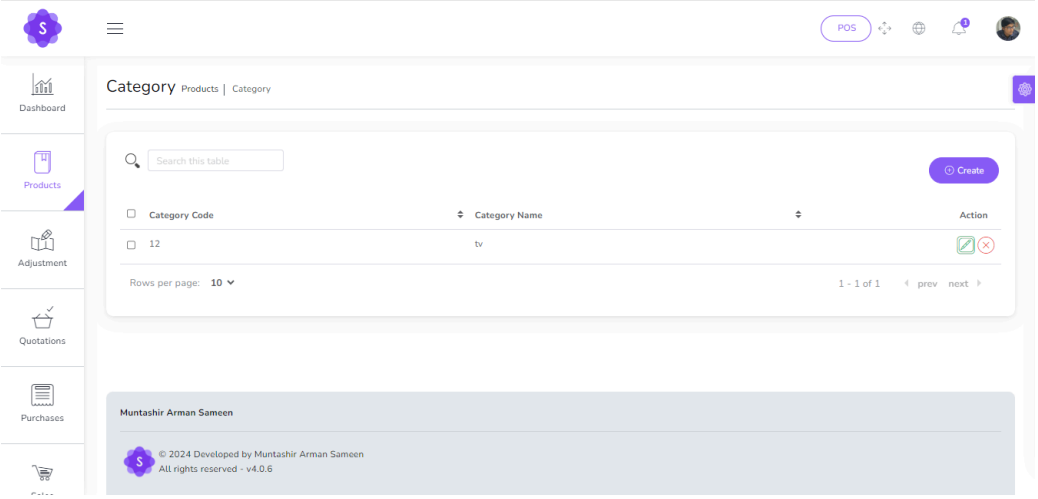


Fig 3.8: Category Product

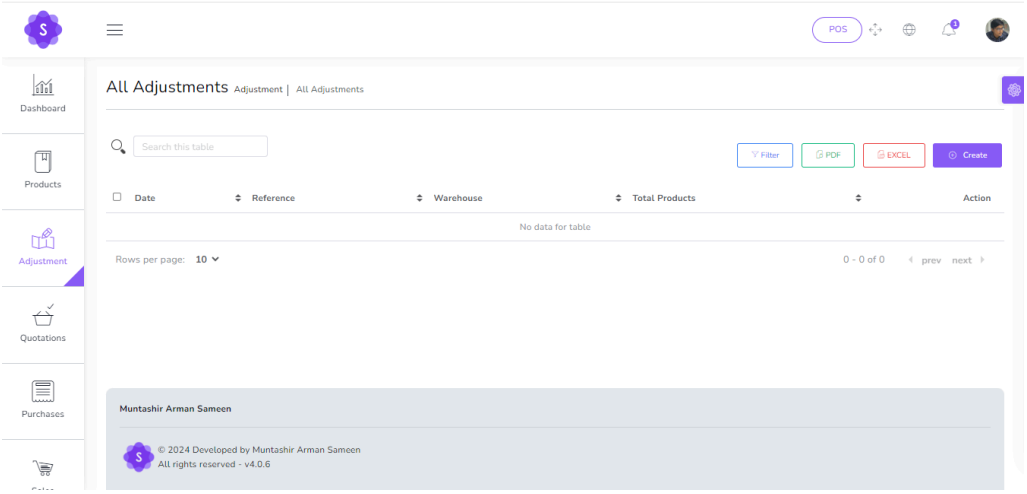


Fig 3.9: All Adjustments

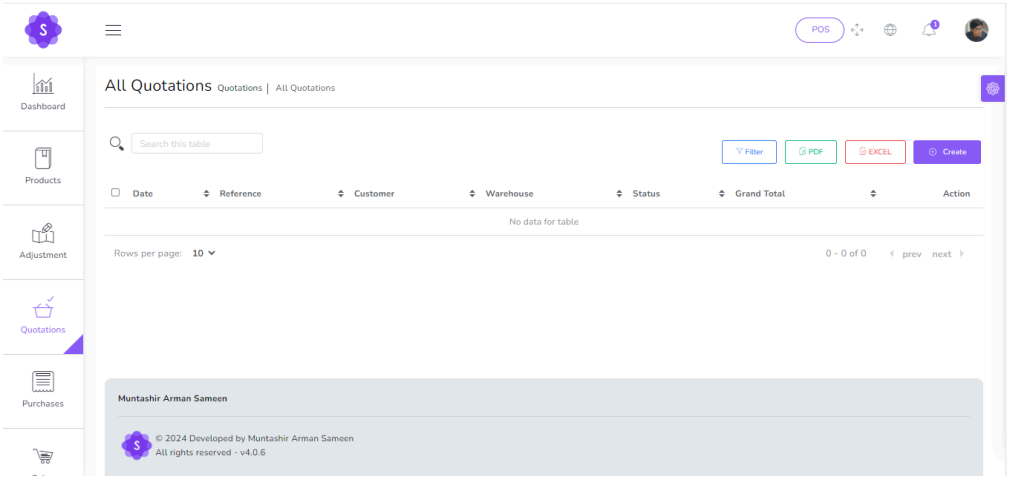


Fig 3.10: All Quotations

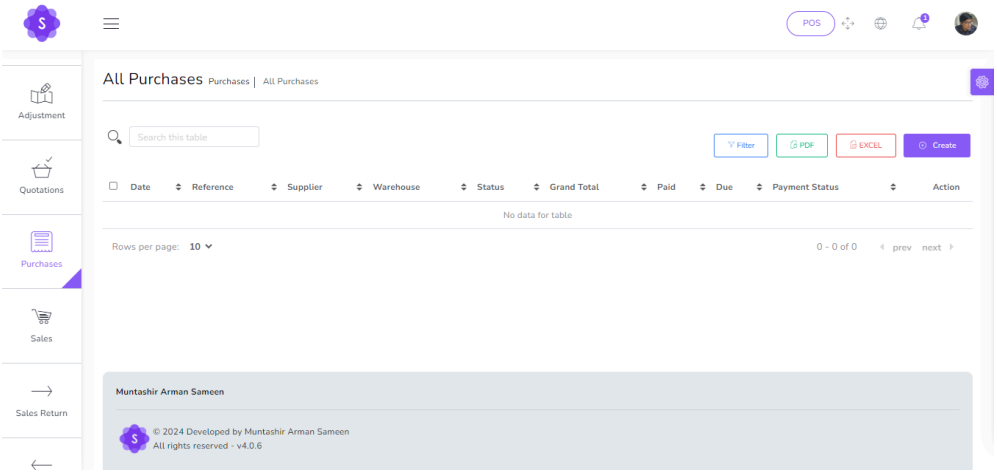


Fig 3.11: All Purchases

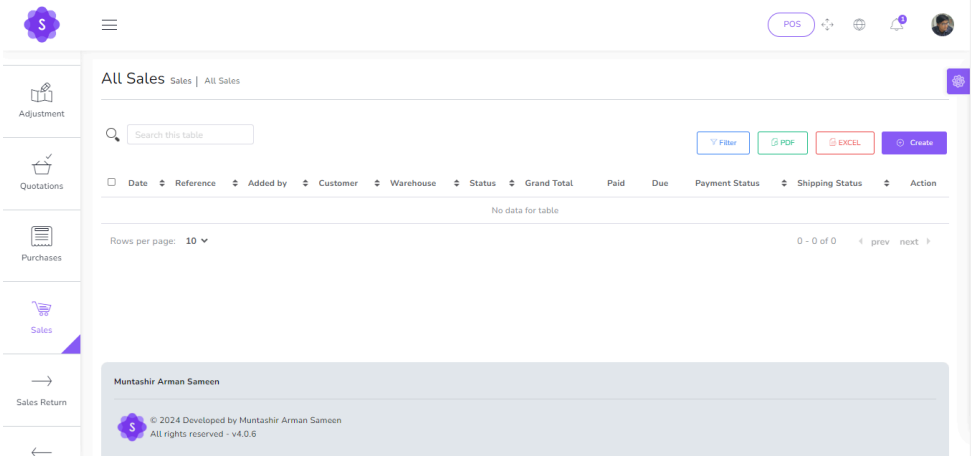


Fig 3.12: All Sells

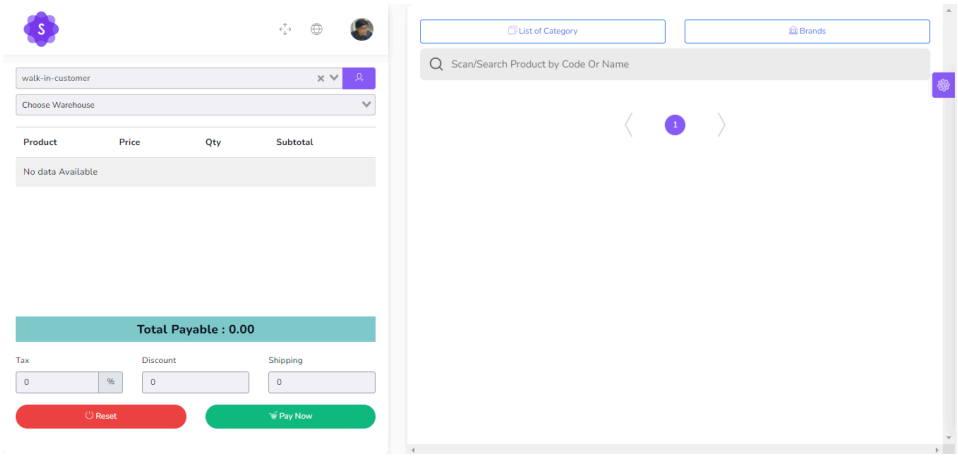


Fig 3.13: POS

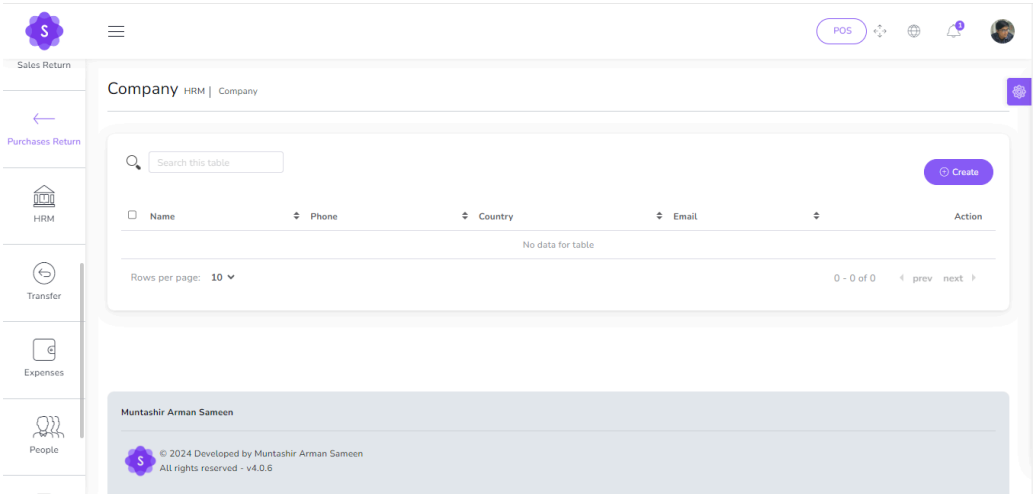


Fig 3.14: HRM Company

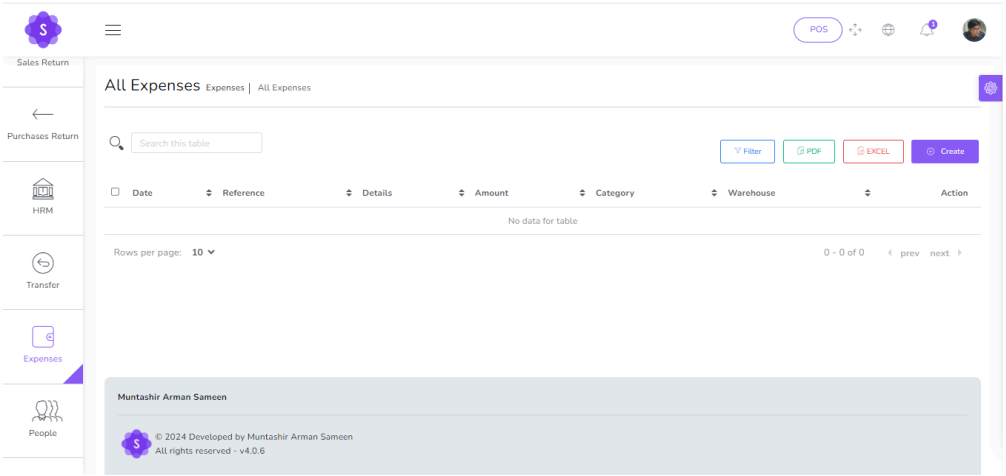


Fig 3.15: All Expenses

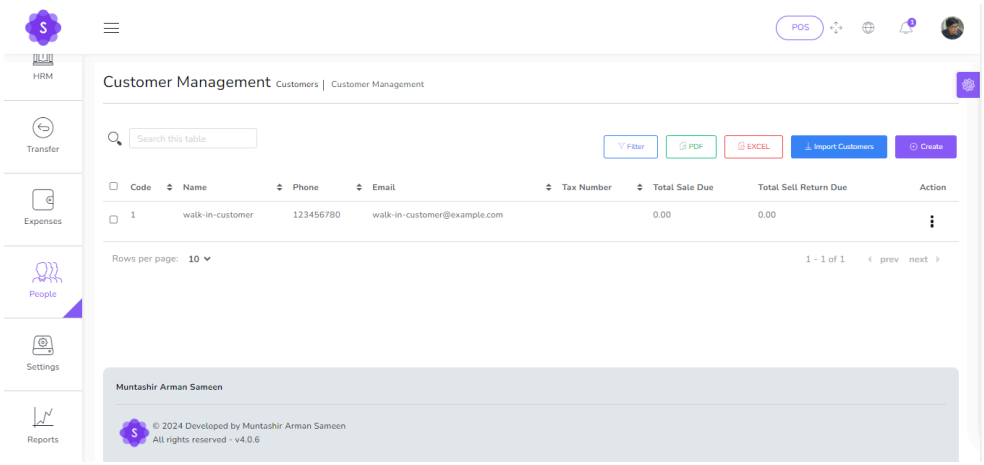


Fig 3.16: Customer Management

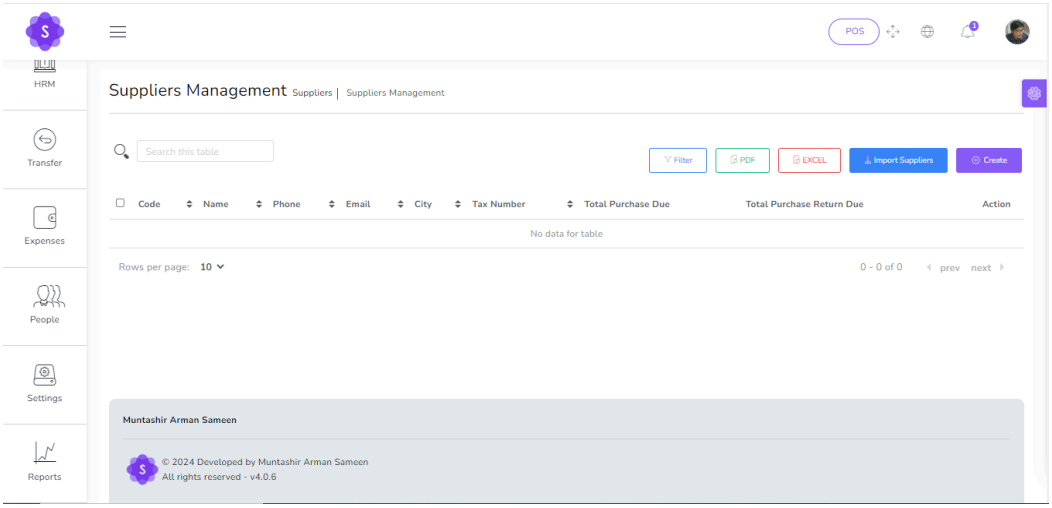


Fig 3.17: Suppliers Management

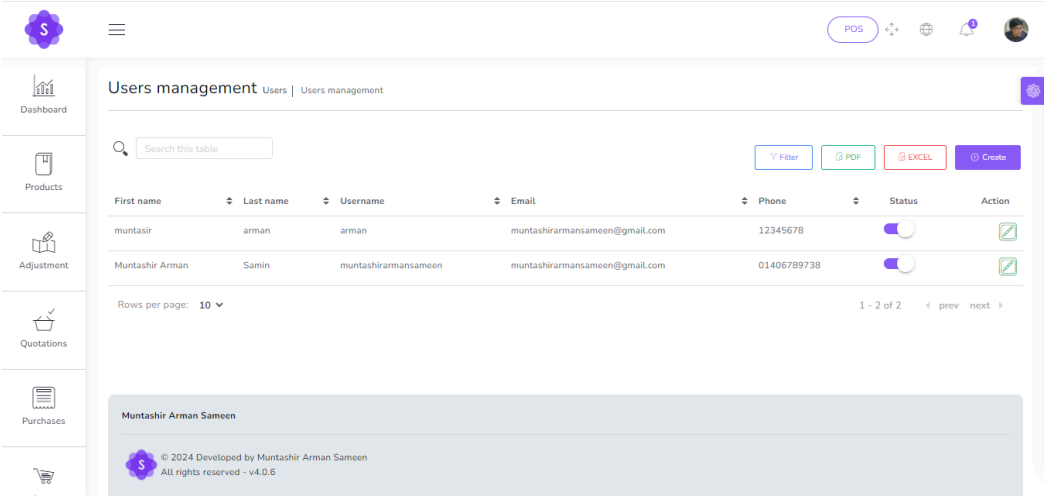


Fig 3.18: User Management

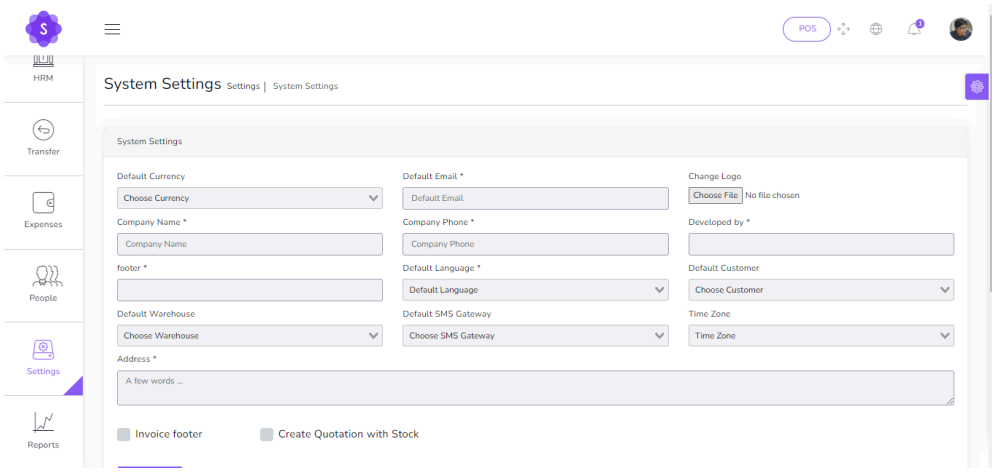


Fig 3.19: System Settings

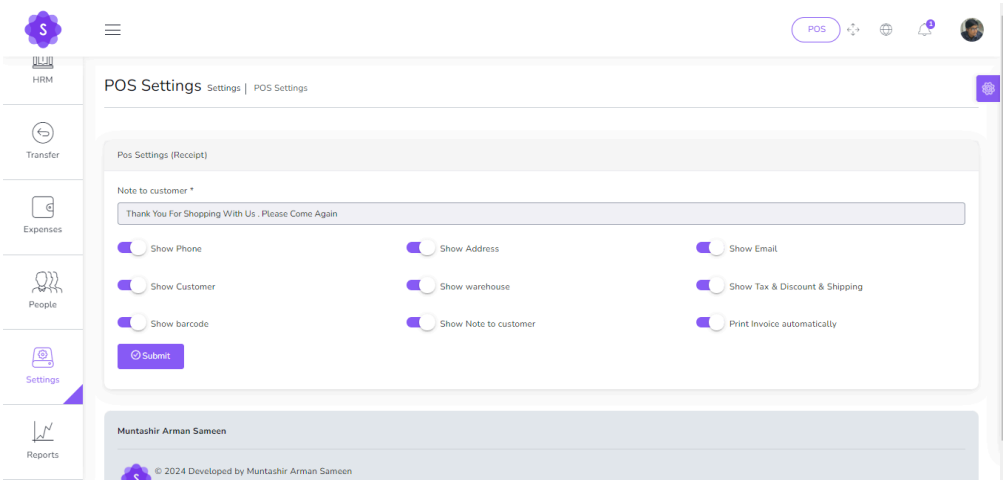


Fig 3.20: POS Settings

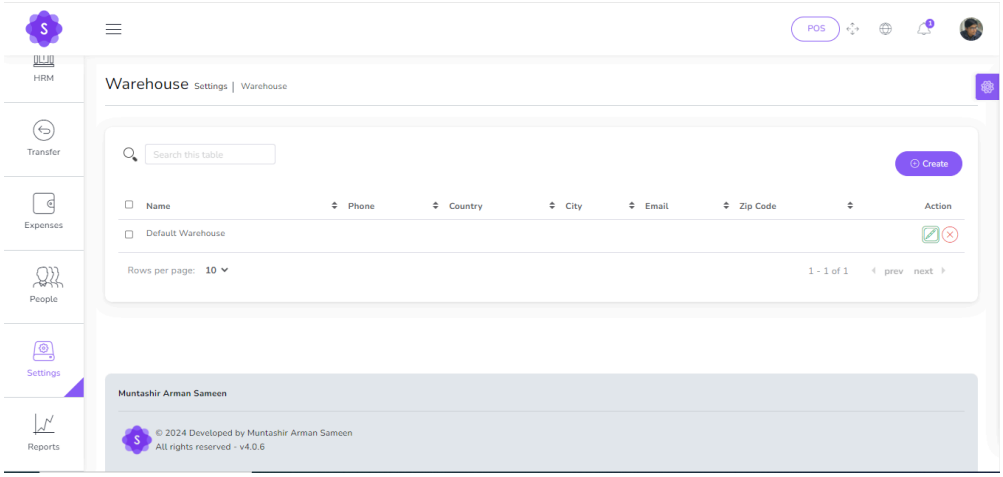


Fig 3.21: Warehouse Page

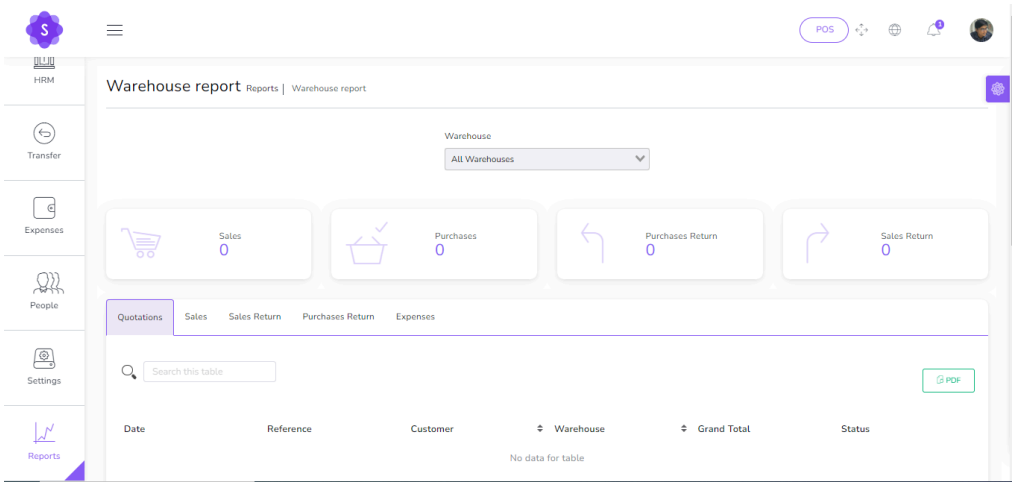


Fig 3.22: Warehouse Report

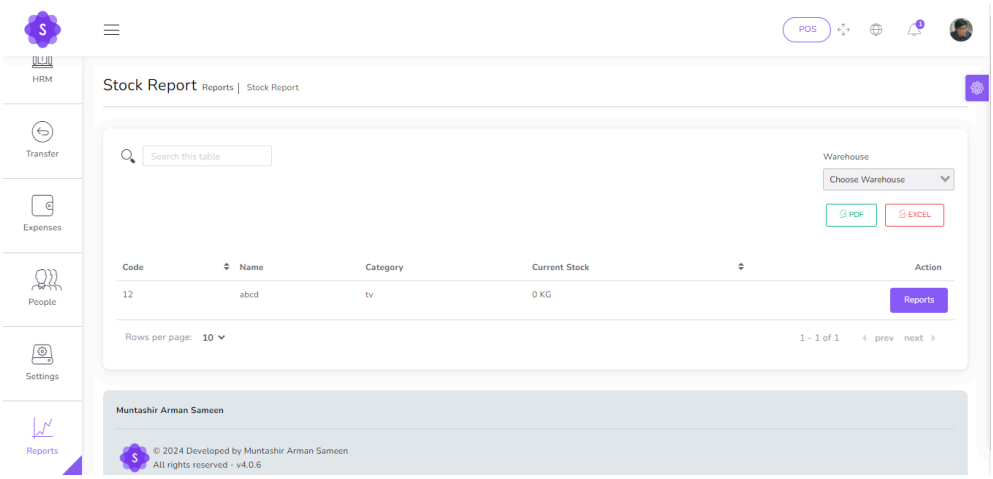


Fig 3.23: Stock Report

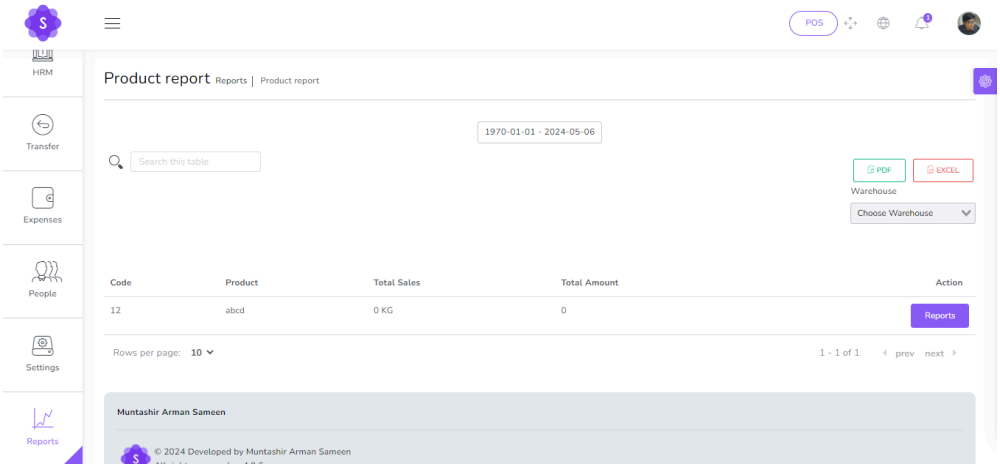


Fig 3.24: Product Report

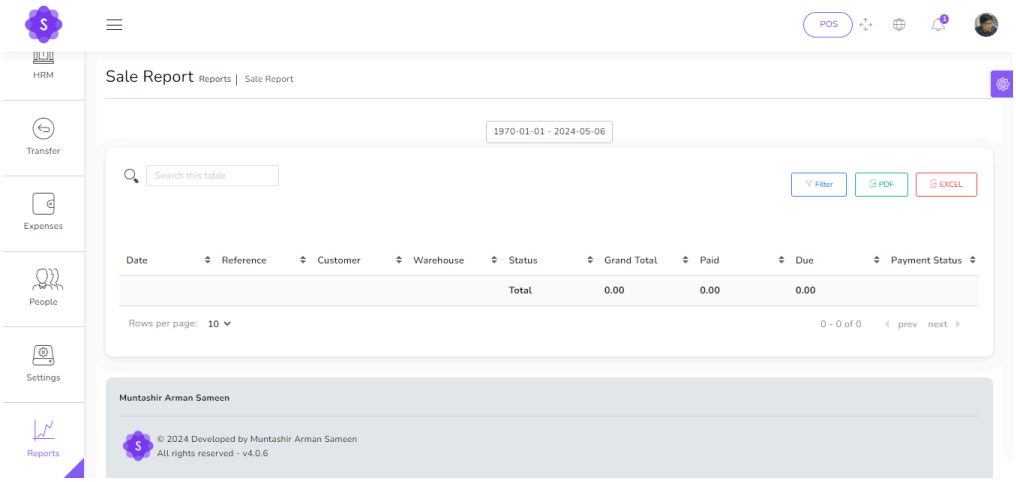


Fig 3.25: Sale Report

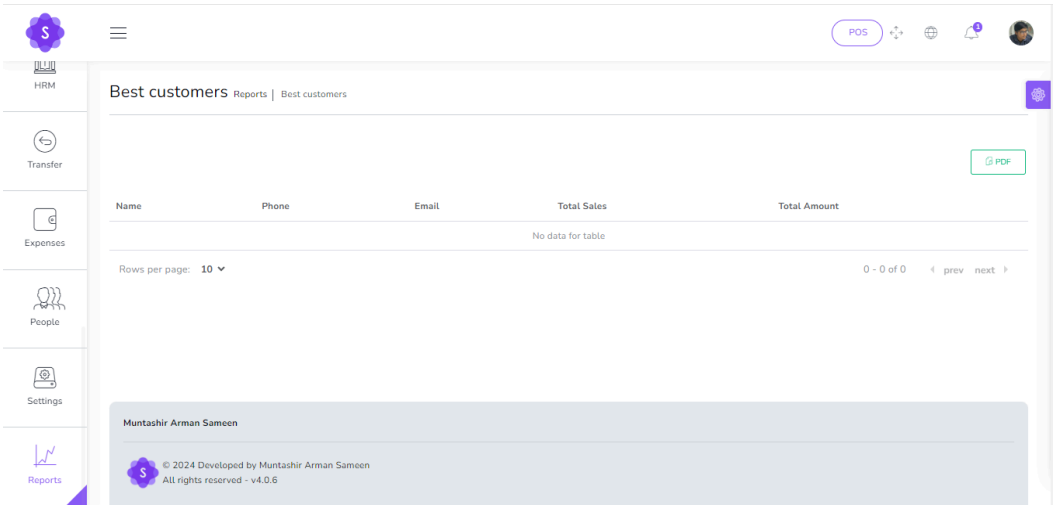


Fig 3.26: Best Customers

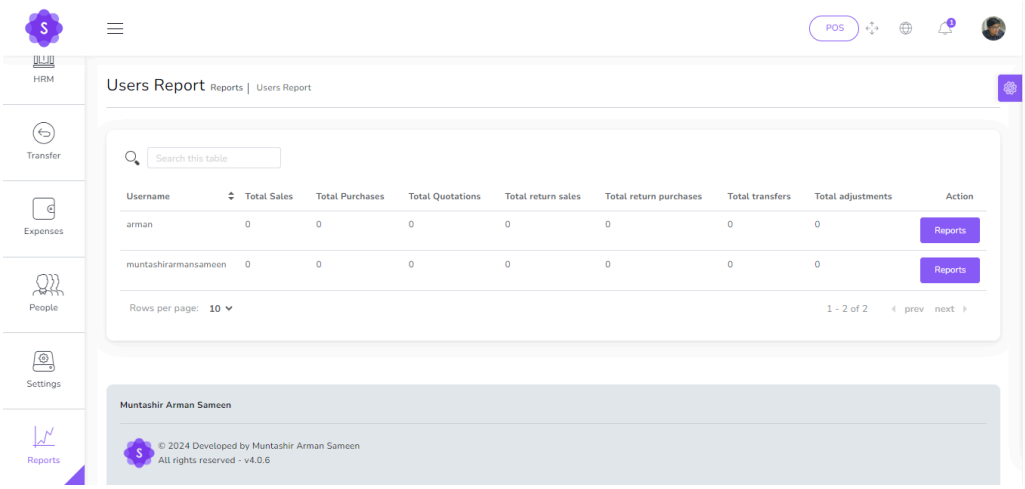


Fig 3.27: User Report

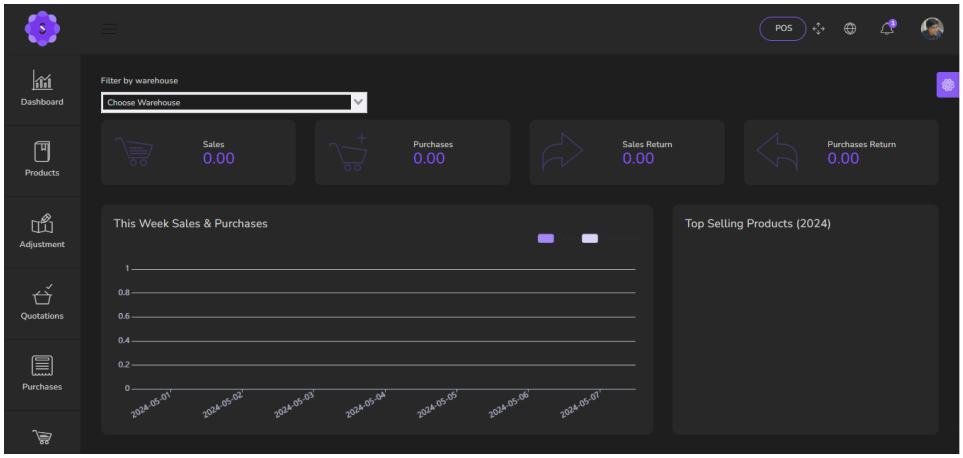


Fig 3.28: Dark Mode-1

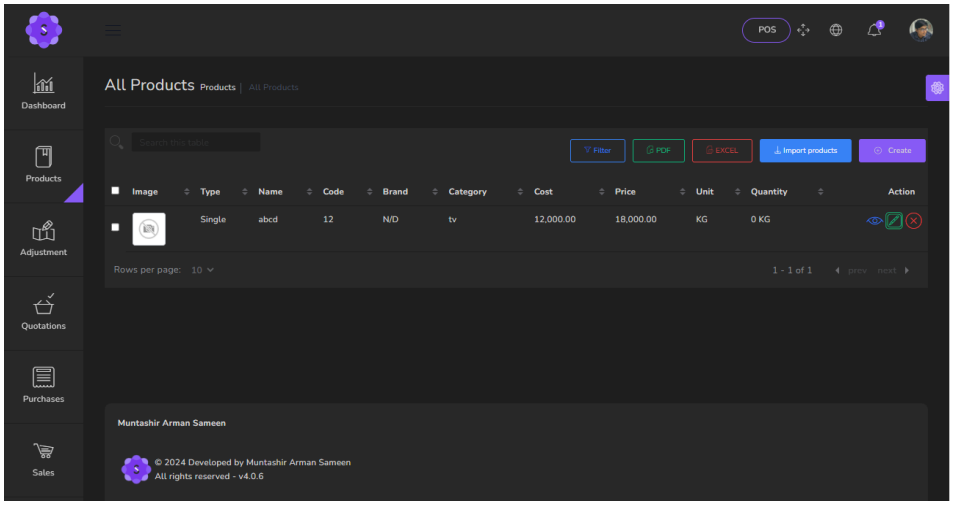


Fig 3.29: Dark Mode-2

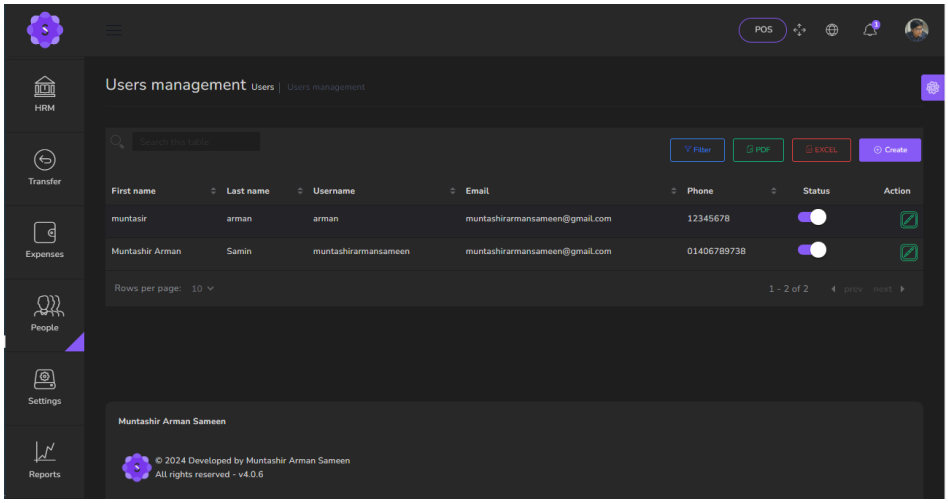


Fig 3.30: Dark Mode-3

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Implementation Requirements

An easy-to-use interface that doesn't require advanced training, support, or documentation. Automation for eliminating manual processes of business functions related to inventory management. A reliable, secure database that provides accurate, real-time data.

4.2 Requirement Collection and Analysis

Requirement analysis is critical to the success or failure of a systems or software project. We analyzed the requirements of these activities with the help of our honorable project supervisor.

Here is the main requirement analysis:

> Admin, System User, and Supplier Log Into The System

> Admin Manages The System

> System Users Can Purchasing Manage Supplier

> Supplier Can Manage Bills, Stock, And Check Inventory

> Logout

4.3 Design Analysis

The website design and development procedure for our project has been shown using a use case diagram & use case description, E-R diagram. The whole design is user-friendly and any nontechnical person will understand the design procedure easily just by viewing the described diagram.

4.4 Completion

This project is all about inventory management properly. This website is easy to use. We have more features and updates in our minds. In the future, we will try to add those features.

4.5 Pass/Fail Criteria

Generally, test engineers set the pass or fail criteria. They prepare the pass/fail criteria according to which input data worked and which did not work. Pass criteria are the worked data and the rest of the input data are considered as fail criteria.

For This System Pass/Fail Criteria Are Given Below:

1. One Criterion Should Always Be Passing Accurately To Be Considered as Pass Criteria
2. System Crush Will Be Considered as a Fail Criterion
3. Not Showing Expected Results Will Be Considered as a Fail Criteria

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Purpose of Testing

The purpose of software testing is to access or evaluate the capabilities or attributes of a software program's ability to adequately meet the applicable standards and application needs. Testing does not ensure quality and the purpose of testing is not to find bugs. Testing can be verification and validation or reliability estimation. Testing serves several critical purposes in the development and maintenance of software systems, including inventory management systems:

1. **Identifying Defects:** Testing helps in identifying defects or bugs in the software. These defects can range from simple logic errors to more complex issues related to integration, performance, or security. By detecting and fixing defects early in the development process, testing helps prevent larger problems down the line.

2. **Ensuring Quality:** The primary purpose of testing is to ensure the quality of the software. Quality assurance activities, including testing, aim to verify that the software meets specified requirements, performs as expected, and delivers value to users. By rigorously testing the software against predefined criteria, developers can ensure that it meets quality standards.

3. **Validating Requirements:** Testing helps validate that the software meets the requirements specified by stakeholders. By comparing the actual behavior of the software against expected outcomes, testing helps ensure that the software fulfills its intended purpose and meets user needs.

4. **Mitigating Risks:** Testing helps mitigate risks associated with software development and deployment. By identifying potential issues early in the development process, testing allows developers to address them proactively, reducing the likelihood of costly failures or system downtime in production environments.

5. **Improving Reliability:** Testing contributes to improving the reliability of software systems. Through various testing activities, such as unit testing, integration testing, and system testing, developers can verify the correctness and robustness of the software, increasing its reliability and stability in real-world usage scenarios.

6. **Enhancing User Satisfaction:** Testing plays a crucial role in enhancing user satisfaction by ensuring that the software meets user expectations and provides a positive user experience. By detecting and fixing defects that impact usability, performance, or functionality, testing helps create software that users can rely on and enjoy using.

7. **Facilitating Maintenance:** Testing helps facilitate the maintenance and evolution of software systems over time. By providing a safety net for making changes and updates to the software, testing allows developers to confidently modify existing functionality or add new features without introducing regressions or unintended consequences.

8. **Complying with Standards and Regulations:** In certain industries, such as healthcare or finance, testing is essential for ensuring compliance with regulatory requirements and industry standards. By demonstrating that the software meets specified quality and security standards, testing helps organizations mitigate legal and financial risks associated with non-compliance.

Overall, testing is a critical part of the software development lifecycle, helping to ensure that software systems are reliable, high-quality, and meet the needs of users and stakeholders.

5.2 Type of Testing

We have used one type of testing to ensure the error-free features of our software application. Testing is a crucial aspect of ensuring the quality and reliability of an inventory management system. Here are some types of testing commonly employed in such systems:

1. **Unit Testing:** This involves testing individual components or modules of the system in isolation to ensure they work correctly. In an inventory management system, unit testing might involve testing functions related to adding, updating, or deleting inventory items.

2. **Integration Testing:** Integration testing verifies that different modules or components of the system work together as expected. For example, integration testing in an inventory management system would ensure that the inventory database integrates seamlessly with the user interface.

3. **System Testing:** This type of testing evaluates the system as a whole to ensure that it meets the specified requirements. System testing in an inventory management system might involve testing end-to-end scenarios such as adding new inventory items, updating quantities, and generating reports.

4. **User Acceptance Testing (UAT):** UAT involves testing the system with real users to ensure that it meets their needs and expectations. In an inventory management system, UAT might involve users performing common tasks such as adding new products, creating purchase orders, or running inventory reports.

5. **Performance Testing:** Performance testing assesses the system's responsiveness, scalability, and stability under various load conditions. For an inventory management

system, performance testing might involve simulating a large number of concurrent users or transactions to evaluate system performance.

6. **Security Testing:** Security testing evaluates the system's ability to protect sensitive data and prevent unauthorized access. In an inventory management system, security testing might involve testing user authentication mechanisms, data encryption, and access controls.

7. **Regression Testing:** Regression testing ensures that new changes or updates to the system do not adversely affect existing functionality. In an inventory management system, regression testing might involve retesting core inventory management features after a software update or bug fix.

8. **Usability Testing:** Usability testing assesses the system's ease of use and user interface design. In an inventory management system, usability testing might involve observing users as they perform common tasks and gathering feedback on the system's navigation, layout, and overall user experience.

By conducting these various types of testing, developers can identify and address any issues or defects in the inventory management system, ensuring that it meets quality standards and delivers value to users.

5.3 Units Test

This type of testing is the testing of individual software components. It is typically done by the programmer and not by the testers. It requires details information and knowledge about the internal program design and code to perform this. During unit testing, we carried out various testing tasks such as the reflection of the unit data on the database and its interface. Various types of bugs associated with the component were identified and fixed. We use various functional keys to test our software. In our software unit testing is concerned with the stock units, opening stock units, and product unit validation as well as the validation of product.

Unit testing in an inventory management system focuses on testing individual units or components of the system in isolation. Here's how you might approach unit testing for different units within the system:

1. **Inventory Item Unit Test:**
    - Test adding a new inventory item to the system and verify that it is correctly stored in the database.
    - Test updating the quantity or other attributes of an existing inventory item and verify that the changes are applied correctly.

- Test deleting an inventory item and verify that it is removed from the database.

2. **Inventory Manager Unit Test:**
   - Test the functionality to retrieve a list of all inventory items and verify that the correct data is returned.
   - Test searching for inventory items by name, category, or other criteria and verify that the search results are accurate.
   - Test generating reports such as inventory levels or stock movements and verify that the reports are generated correctly.
3. **Order Management Unit Test:**
   - Test creating a new purchase order and verify that it is correctly saved in the system.
   - Test updating the status of a purchase order (e.g., from pending to fulfilled) and verify that the status change is reflected accurately.
   - Test generating invoices or receipts for completed orders and verify that the documents are generated correctly.
4. **User Authentication Unit Test:**
   - Test registering a new user account and verify that the account is created successfully.
   - Test logging in with valid credentials and verify that the user is authenticated and granted access to the system.
   - Test logging in with invalid credentials and verify that the appropriate error message is displayed.
5. **Notification Unit Test:**
   - Test sending notifications to users for events such as low stock alerts or order status updates and verify that the notifications are sent correctly.
   - Test handling different types of notification triggers and verify that the appropriate notifications are generated.
6. **Audit Logging Unit Test:**
   - Test logging actions performed by users (e.g., adding, updating, or deleting inventory items) and verify that the log entries are recorded accurately.
   - Test querying the audit log for specific actions or periods and verify that the relevant log entries are retrieved.
7. **Data Validation Unit Test:**
   - Test input validation for various forms and fields (e.g., quantity, price) and verify that invalid inputs are rejected with appropriate error messages.
   - Test boundary cases for numeric fields (e.g., minimum and maximum values) and verify that the system handles them correctly.
8. **Error Handling Unit Test:**
   - Test scenarios where the system encounters errors (e.g., database connection failure, invalid input) and verify that the errors are handled gracefully with appropriate error messages.

CHAPTER 6
CONCLUSION AND FUTURE WORKS

6.1 Project Limitation

Since this is our first project it has some limitations. Due to less knowledge in particular fields and limited time, we were not able to fulfill all our expectations that we expected we could do while the project got started. We hope this limitation is considerable. Project limitations define the boundaries within which the project operates and can include constraints on resources, scope, time, and technology. Here are some common project limitations:

1. **Scope Limitations:** The project scope defines the functionalities and features included in the inventory management system. Limitations on scope might arise due to budget constraints, time constraints, or the complexity of the desired features. For example, the project might be limited to basic inventory tracking and reporting functionalities, with more advanced features deferred to future phases.

2. **Not For Large Organizations:** This application is not suitable for those organizations where there is a large quantity of products and different levels of warehouses.

3. **Resource Constraints:** Resource limitations can include constraints on budget, manpower, equipment, or infrastructure. For instance, the project might have a limited budget for software development, necessitating careful prioritization of features and efficient use of resources.

4. **Time Constraints:** Time limitations can affect the project schedule and deadlines for completion. Tight deadlines might require streamlining the development process, reducing the scope of work, or allocating additional resources to meet project milestones.

5. **Technology Limitations:** Technology constraints can arise from limitations in the available tools, platforms, or programming languages. For example, the project might be constrained to using specific software frameworks or databases due to compatibility requirements or organizational standards.

6. **Data Limitations:** Data limitations refer to constraints on the availability, quality, or quantity of data needed for the project. For instance, the project might be limited by the amount of historical inventory data available for testing and analysis.

7. **Regulatory Compliance:** Regulatory constraints can arise from legal or industry requirements that must be adhered to in the development of the inventory management system. For example, the system might need to comply with data protection regulations such as GDPR or industry-specific standards like HIPAA in healthcare.

8. **Security Concerns:** Security limitations relate to the protection of sensitive data and the prevention of unauthorized access. The project might be limited by the need to implement robust security measures to safeguard inventory information from breaches or cyberattacks.

9. **User Limitations:** User limitations refer to constraints related to the knowledge, skills, or preferences of end-users. For example, the system might need to accommodate users with varying levels of technical expertise or accessibility requirements.

By acknowledging and documenting project limitations, stakeholders can manage expectations, prioritize tasks effectively, and make informed decisions to ensure the successful completion of the inventory management system within the defined constraints.

6.2 Conclusion

To conclude, Stock Inventory Management System is a web application suitable for small organizations. It has every basic item that is used for the small organization. I am successfuly making the application where we can update, insert, and delete the item as per the requirement. This application also provides a simple report daily and weekly to know the daily sales and purchase details. This application matches small organizations where there small limit. Though it has some limitations, I strongly believe that implementation of this system will surely benefit the organization.

In conclusion, the development of an inventory management system represents a significant endeavor with multifaceted considerations. Throughout this project, we have embarked on a journey to conceptualize, design, implement, and test a robust solution tailored to meet the specific needs of inventory management in our context.

We began by identifying the importance of efficient inventory management for businesses, recognizing the pivotal role it plays in optimizing operations, minimizing costs, and maximizing profitability. With clear objectives in mind, we defined the scope of our project, outlining the functionalities and features to be included while acknowledging the limitations and constraints within which we operate.

Through meticulous planning and execution, we formulated system requirements, designed an appropriate architecture, and developed an intuitive user interface to facilitate seamless interaction. Leveraging cutting-edge technologies and best practices, we implemented core functionalities such as inventory item management, order processing, user authentication, and reporting, ensuring alignment with stakeholder expectations and industry standards.

Testing emerged as a cornerstone of our project, enabling us to validate requirements, identify defects, and ensure the quality, reliability, and security of our system. From unit testing

individual components to system-wide integration testing and user acceptance testing, we meticulously scrutinized every aspect of our solution to deliver a robust, dependable, and user-friendly inventory management system.

As we reflect on our journey, we recognize the collaborative effort and dedication of all stakeholders involved in bringing this project to fruition. While this iteration of the inventory management system represents a significant milestone, we acknowledge that software development is an iterative process, and there are always opportunities for enhancement and refinement.

In conclusion, our web application stands as a testament to our commitment to innovation, efficiency, and customer satisfaction. Moving forward, we remain vigilant in monitoring system performance, addressing user feedback, and embracing new technologies to ensure continued success and relevance in an ever-evolving landscape.

Together, we have laid the foundation for streamlined inventory management, empowering businesses to thrive in today's dynamic marketplace. With a shared vision and relentless pursuit of excellence, we look forward to the continued evolution and success of our inventory management system.

6.3 Lesson Learnt

Doing something for long periods always gives good lessons. Some of the things that our team learned are listed below:

1. How to Develop a Web Application
2. Learn About Some Web Technologies
3. Learn Front-end: HTML, CSS, Bootstrap, Javascript, Vue.js
4. Learn Back-end: PHP, Laravel, (Database-MySQL)

REFERENCE

[1] Inventory Management Systems IMS,(2023) Asia Pacific University of Technology,Kual a Lumpur, Malaysia https://www.researchgate.net/publication/372653862_Inventory_Ma nagement_Systems_IMS

[2] Aakashpawar (2019) Laravel Controller Basics https://www.geeksforgeeks.org/laravel-co ntroller-basics/

[3] Aamir Khan, Aasif Ansari, MD Ghalib, H. P. (2019). IRJET- A Web Application Inventory Management System for Server Center of AIKTC. International Research Journal of Engineering and Technology, 06 (02), 2203 – 2205 https://www.irjet.net/archives/V6/i2/IRJET-V6I2436.pdf

[4] Arina Ramlee, David Henry, B. C. (2019). Inventory Management System Finance & Management Engineering Journal of Africa, 1 (6), 22–37 https://damaacademia.c om/fmeja/wp-content/uploads/2019/07/FMEJ-06-019-002.pdf

[5] Bajjouk, M., Rana, M. E., Ramachandiran, C. R., & Chelliah, S. (2021). Software testing for reliability and quality improvement. Journal of Applied Technology and Innovation, 5(2), 40-46.

[6] Jallow, A. (2018). DZIGUA ENTERPRISE INVENTORY MANAGEMENT SYSTEM.

[7] https://air.ashesi.edu.gh/bitstream/handle/20.500.11988/436/Jallow_Alieu_2018_CS_App liedProject.pdf?sequence=1&isAllowed=y

[8] Kausar Bagwan, M. I., & Swati Ghule, P. D. (2019). A Modern Review on Laravel-PHP Framework. IRE Journals, 2(12).

[9] Oracle (2022), MySQL https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html Otwell, T. (2022). Release Notes of Laravel 9 https://laravel.com/docs/9.x/releases

[10] Patil, H., & Divekar, B. R. (2014). Inventory Management Challenges for B2C E-commerce Retailers. Procedia Economics and Finance, https://doi.org/10.1016/s2212-5671(14)00221-4. PhpStorm. (2022)