

INVENTORY MANAGEMENT SYSTEM

A MINI PROJECT REPORT

SUBMITTED BY

SNEKA SORNA.P.S	221701056
TAMILARASI.R	221701060
SURYA.N	221701059
SOLOMON SAMUEL	221701058

In partial fulfillment for the award of the degree of

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND DESIGN**

**RAJALAKSHMI ENGINEERING COLLEGE
THANDALAM
CHENNAI – 602105**



ANNA UNIVERSITY: CHENNAI 600625

BONAFIDE CERTIFICATE

Certified that this project report “ **INVENTORY MANAGEMENT SYSTEM**” is the bonafide work of

**“SNEKA SORNA.P.S (221701056), TAMILARASI.R (221701060),
SURYA.N (221701059), SOLOMON SAMUEL (221701058)”** who carried
out the project work under my supervision.

SIGNATURE

Dr.P.Kumar M.E., Ph.D.,
Professor and Head,
Computer Science and Engineering,
Rajalakshmi Engineering College,
Thndalam, Chennai – 602105.

SIGNATURE

Mr.Vijaykumar M.Tech.,
Asst. Professor (SS),
Computer Science and Engineering,
Rajalakshmi Engineering College,
Thandalam, Chennai – 602105.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are highly obliged in taking the opportunity to thank our Chairman **Mr. S. Meganathan**, Chairperson **Dr.Thangam Meganathan** and our Principal **Dr.S.N.Murugesan** for providing all the facilities which are required to carry out this project work.

We are ineffably indebted to our H.O.D **Dr.P.Kumar M.E., Ph.D.**, for his conscientious guidance and encouragement to make this project a recognizable one.

We are extremely thankful to our faculty **Mr.Vijaykumar M.Tech.**, for his valuable guidance and indefatigable support and extend our heartfelt thanks to all the teaching and non-teaching staff of **Computer Science department** who helped us directly or indirectly in the completion of this project successfully.

At last but not least gratitude goes to our friends who helped us compiling the project and finally to god who made all things possible.

Any omission in this brief acknowledgement doesn't mean lack of gratitude.

SNEKA SORNA.P.S (221701056)

TAMILARASI.R (221701060)

SURYA.N (221701059)

SOLOMON SAMUEL (221701058)

ABSTRACT

The Inventory Management System (IMS) is a web-based application designed to streamline inventory management for businesses. It provides a user-friendly platform where businesses can manage products, categories, and sales records efficiently. The system is built using PHP, MySQL, Bootstrap, HTML, CSS, and JavaScript and features a secure user authentication system to ensure that only authorized users can access it.

The IMS is divided into three modules: Admin, Special User, and User Login. The Admin has full control of the system, while Special Users and regular Users have distinct roles. Businesses can add, edit, and delete products and categories, as well as manage sales records. The system also generates sales reports on a daily, weekly, and monthly basis, offering businesses valuable insights into their sales performance.

This platform benefits businesses by simplifying inventory management and enabling better organization of products and sales records on a single interface. The IMS also serves as a data analytics tool, allowing businesses to analyze sales trends and make informed decisions. Additionally, its collaborative features enable employees to connect and manage inventory data more effectively.

In summary, the Inventory Management System is a comprehensive tool that helps businesses manage inventory, track sales data, and improve operational efficiency with ease.

TABLE OF CONTENTS

1. INTRODUCTION

- 1.1 INTRODUCTION [6]
- 1.2 SCOPE OF WORK..... [6]
- 1.3 PROBLEM STATEMENT..... [6]
- 1.4 AIM AND OBJECTIVE..... [6]

2. SURVEY OF TECHNOLOGIES

- 2.1 SOFTWARE DESCRIPTION [7]
- 2.2 LANGUAGES [8]

3. REQUIREMENT AND ANALYSIS

- 3.1 REQUIREMENT SPECIFICATION [9]
- 3.2 HARDWARE AND SOFTWARE SPECIFICATION..... [10]

4. PROGRAM CODE.....[11]

5. RESULT AND DISCUSSION

- 5.1 FUNCTIONALITY OF THE PROJECT [15]
- 5.2 USER FEEDBACK [22]
- 5.3 CHALLENGES FACED DURING DEVELOPMENT [22]

6. CONCLUSION.....[24]

7. REFERENCES[25]

CHAPTER 1

1.INTRODUCTION

An Inventory Management System (IMS) is a software application designed to monitor and manage the stock of products in a business or organization. It helps streamline operations by automating stock tracking, reducing manual errors, and ensuring accurate and timely data for informed decision-making. Effective inventory management is essential for reducing costs, preventing stockouts, and improving customer satisfaction.

2.SCOPE OF WORK

The scope of this project covers the development of an Inventory Management System that enables businesses to track and control their inventory in real-time. The system will support core functionalities such as adding, updating, and deleting stock items; generating inventory reports; setting reorder levels; and providing alerts for low stock levels. The system will be designed to support a variety of businesses, from small retail operations to larger warehouses.

3.PROBLEM STATEMENT

Many businesses face challenges in managing their inventory effectively, leading to issues such as overstocking, stockouts, and inaccurate stock data. These issues can result in increased operational costs, lost sales, and inefficient resource utilization. Traditional, manual inventory management processes are prone to errors, time-consuming, and lack the capacity to provide real-time stock information.

4.AIM AND OBJECTIVE

The aim of this project is to develop a user-friendly Inventory Management System that simplifies the tracking, management, and reporting of inventory. This system aims to support businesses in optimizing their inventory levels, reducing manual errors, and ensuring data accuracy.

Objectives:

- 1. Automate Inventory Tracking**
- 2. Reduce Manual Errors**
- 3. Optimize Stock Levels**
- 4. Generate Reports and Improve Efficiency**

CHAPTER 2

2.1 SOFTWARE DESCRIPTION

The Inventory Management System software is a computer-based application that is designed to provide a comprehensive solution for inventory management. The software is developed using various programming languages, tools, and technologies to ensure its efficiency, scalability, and security. The following is a description of the software used in the development of the Inventory Management System:

PHP: PHP is a server-side scripting language that is widely used in web development. It is used in the development of the Inventory Management System to handle the server-side logic and database interactions.

MySQL: MySQL is a popular relational database management system that is used to store and manage the data of the Inventory Management System. It is used in conjunction with PHP to handle the database interactions and queries.

Bootstrap: Bootstrap is a front-end framework that is used to design the user interface of the Inventory Management System. It provides a set of pre-designed CSS and JavaScript components that can be used to create responsive and mobile-friendly web pages.

HTML: HTML is a markup language that is used to structure the content of the Inventory Management System. It is used in conjunction with CSS and JavaScript to create the user interface.

CSS: CSS is a style sheet language that is used to design the visual appearance of the Inventory Management System. It is used in conjunction with HTML and JavaScript to create the user interface.

JavaScript: JavaScript is a client-side scripting language that is used to handle the client-side logic and user interactions of the Inventory Management System. It is used in conjunction with HTML, CSS, and PHP to create the user interface and handle the server-side logic.

These software components are integrated and tested to ensure the proper functioning of the Inventory Management System. The software is designed to be scalable, efficient, and secure, with regular updates and maintenance to ensure its continued performance.

2.2 LANGUAGES

2.2.1 SQL

Structured Query Language (SQL) is employed for database operations within the HMS. SQL is a standardized language used to manage relational databases and perform various operations on the data, such as querying, updating, and managing database schema creation and modifications. The use of SQL ensures that data is efficiently stored, organized, and accessed, which is critical for maintaining accurate and up-to-date hospital records.

2.2.2 PHP

PHP (Hypertext Preprocessor) is the core programming language used for the server-side scripting of the HMS. PHP's ability to embed within HTML, its compatibility with various databases, and its extensive set of built-in functions make it a powerful tool for developing web applications. In the HMS, PHP handles the logic behind user interactions, data processing, and communication with the MySQL database, ensuring that the system is responsive and interactive.

CHAPTER 3

3.1 REQUIREMENT SPECIFICATION

3.1.1 Functional Requirements:

1. User Management:

- Admin should be able to create, update, and delete user accounts.
- Special users should have limited access to product management and media sections.
- Users should be able to manage their profiles.

2. Product Management:

- Admin and special users should be able to add, update, and delete product details.
- Each product should have attributes like name, category, quantity, buying price, selling price, and associated media files.

3. Sales Management:

- Admin and users should be able to record sales transactions.
- Sales records should include product name, quantity, date, buying price, selling price, and total earnings.

4. Report Generation:

- The system should generate sales reports on a daily, weekly, and monthly basis.
- Reports should include total quantities sold, total earnings, and profit margins.

5. Media Management:

- Users should be able to upload and manage media files associated with products.
- Media files should be selectable when adding or updating product details.

6. Dashboard:

- Admin dashboard should display recent product sales, top-selling products, and highest selling quantities.
- Users should have a personalized dashboard based on their roles and permissions.

3.1.2 Non-Functional Requirements:

1. Security:

- Secure login and authentication for all users.

- Role-based access control to ensure only authorized users can access certain functionalities.

2. Performance:

- The system should be responsive and handle multiple transactions efficiently.
- Reports should be generated quickly, even with large datasets.

3. Usability:

- User interface should be intuitive and easy to navigate.
- Consistent layout and design across all pages for better user experience.

3.2 HARDWARE AND SOFTWARE REQUIREMENT

3.2.1 Hardware Requirements:

- **Server:**
 - Processor: Quad-Core CPU
 - RAM: 8 GB or higher
 - Storage: 500 GB SSD or higher
 - Network: High-speed internet connection
- **Client Devices:**
 - Any device capable of running a modern web browser (desktop, laptop, tablet, or smartphone)
 - Screen resolution: 1024x768 or higher

3.2.2 Software Requirements:

- **Server:**
 - Operating System: Linux (preferred), Windows Server, or macOS
 - Web Server: Apache HTTP Server or Nginx
 - Database Server: MySQL or MariaDB
 - Scripting Language: PHP 7.4 or higher
- **Client Devices:**
 - Web Browser: Latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari

CHAPTER 4

PROGRAM CODE

HOME PAGE:

```
<?php
    $page_title = 'Home Page';
    require_once('includes/load.php');
    if (!$session->isUserLoggedIn(true)) { redirect('index.php', false);}
?>
<?php include_once('layouts/header.php'); ?>
<div class="row">
    <div class="col-md-12">
        <?php echo display_msg($msg); ?>
    </div>
    <div class="col-md-12">
        <div class="panel">
            <div class="jumbotron text-center">
                <h1>Welcome User <hr> Inventory Management System</h1>
                <p>Browse around to find out the pages that you can access!</p>
            </div>
        </div>
    </div>
</div>
<?php include_once('layouts/footer.php'); ?>
```

LOGIN PAGE:

```
<?php
    ob_start();
    require_once('includes/load.php');
    if($session->isUserLoggedIn(true)) { redirect('home.php', false);}
?>
<?php include_once('layouts/header.php'); ?>
<div class="login-page">
    <div class="text-center">
        <h1>Login Panel</h1>
        <h4>Inventory Management System</h4>
    </div>
    <?php echo display_msg($msg); ?>
    <form method="post" action="auth.php" class="clearfix">
        <div class="form-group">
            <label for="username" class="control-label">Username</label>
            <input type="name" class="form-control" name="username">
        </div>
    </form>
</div>
```

```

placeholder="Username">
    </div>

    <div class="form-group">
        <label for="Password" class="control-label">Password</label>
        <input type="password" name="password" class="form-control"
placeholder="Password">
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-danger" style="border-
radius:0%">Login</button>
    </div>
</form>
</div>
<?php include_once('layouts/footer.php'); ?>

```

PROFILE PAGE:

```

<?php

$page_title = 'My profile';
require_once('includes/load.php');
// Checkin What level user has permission to view this page
page_require_level(3);
?>

<?php
$user_id = (int)$GET['id'];
if(empty($user_id)):
    redirect('home.php',false);
else:
    $user_p = find_by_id('users',$user_id);
endif;
?>
<?php include_once('layouts/header.php'); ?>
<div class="row">
    <div class="col-md-4">
        <div class="panel profile">
            <div class="jumbotron text-center bg-red">
                
                <h3><?php echo first_character($user_p['name']); ?></h3>
            </div>
            <?php if( $user_p['id'] === $user['id']):?>
                <ul class="nav nav-pills nav-stacked">
                    <li><a href="edit_account.php"> <i class="glyphicon glyphicon-
edit"></i> Edit profile</a></li>
                </ul>
            <?php endif;?>

```

```

        </div>
    </div>
</div>
<?php include_once('layouts/footer.php'); ?>

```

USER PAGE:

```

<?php
    $page_title = 'All User';
    require_once('includes/load.php');
    ?>
<?php
    // Checkin What level user has permission to view this page
    page_require_level(1);
    //pull out all user form database
    $all_users = find_all_user();
    ?>
<?php include_once('layouts/header.php'); ?>
<div class="row">
    <div class="col-md-12">
        <?php echo display_msg($msg); ?>
    </div>
</div>
<div class="row">
    <div class="col-md-12">
        <div class="panel panel-default">
            <div class="panel-heading clearfix">
                <strong>
                    <span class="glyphicon glyphicon-th"></span>
                    <span>Users</span>
                </strong>
                <a href="add_user.php" class="btn btn-info pull-right">Add New
User</a>
            </div>
            <div class="panel-body">
                <table class="table table-bordered table-striped">
                    <thead>
                        <tr>
                            <th class="text-center" style="width: 50px;">#</th>
                            <th>Name </th>
                            <th>Username</th>
                            <th class="text-center" style="width: 15%;">User Role</th>
                            <th class="text-center" style="width: 10%;">Status</th>
                            <th style="width: 20%;">Last Login</th>
                            <th class="text-center" style="width: 100px;">Actions</th>
                        </tr>
                    </thead>

```

```

<tbody>

<?php foreach($all_users as $a_user): ?>
    <tr>
        <td class="text-center"><?php echo count_id();?></td>
        <td><?php echo remove_junk(ucwords($a_user['name']))?></td>
        <td><?php echo remove_junk(ucwords($a_user['username']))?></td>
        <td class="text-center"><?php echo
remove_junk(ucwords($a_user['group_name']))?></td>
        <td class="text-center">
            <?php if($a_user['status'] === '1'): ?>
                <span class="label label-success"><?php echo "Active"; ?></span>
            <?php else: ?>
                <span class="label label-danger"><?php echo "Deactive"; ?></span>
            <?php endif;?>
        </td>
        <td><?php echo read_date($a_user['last_login'])?></td>
        <td class="text-center">
            <div class="btn-group">
                <a href="edit_user.php?id=<?php echo (int)$a_user['id'];?>"
class="btn btn-xs btn-warning" data-toggle="tooltip" title="Edit">
                    <i class="glyphicon glyphicon-pencil"></i>
                </a>
                <a href="delete_user.php?id=<?php echo (int)$a_user['id'];?>"
class="btn btn-xs btn-danger" data-toggle="tooltip" title="Remove">
                    <i class="glyphicon glyphicon-remove"></i>
                </a>
            </div>
        </td>
    </tr>
<?php endforeach;?>
</tbody>
</table>
</div>
</div>
</div>
<?php include_once('layouts/footer.php'); ?>

```

CHAPTER 5

RESULT AND DISCUSSION

5.1 Functionality of the Project

The Inventory Management System was developed to streamline and automate the process of managing products, sales, and user roles within an organization. The system comprises several key functionalities:

1. User Management:

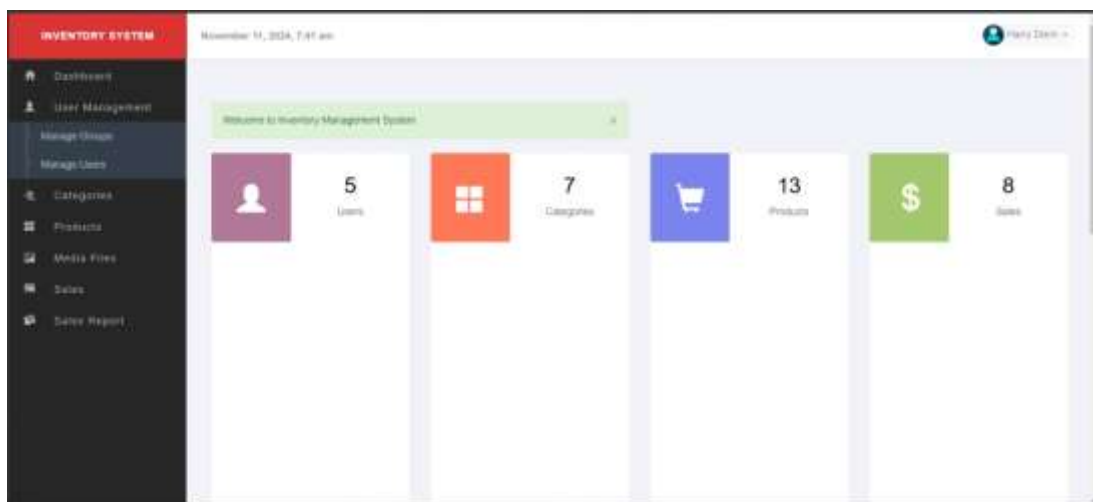


Fig 5.1.1 – User Management

- **Admin Panel:** Admins have full control over the system, including managing users, categories, products, and sales records.
- **Special User Panel:** Special users have access to manage product details and media files.
- **Employee Panel:** Employees can manage sales records and view sales reports.

2. Category Management:

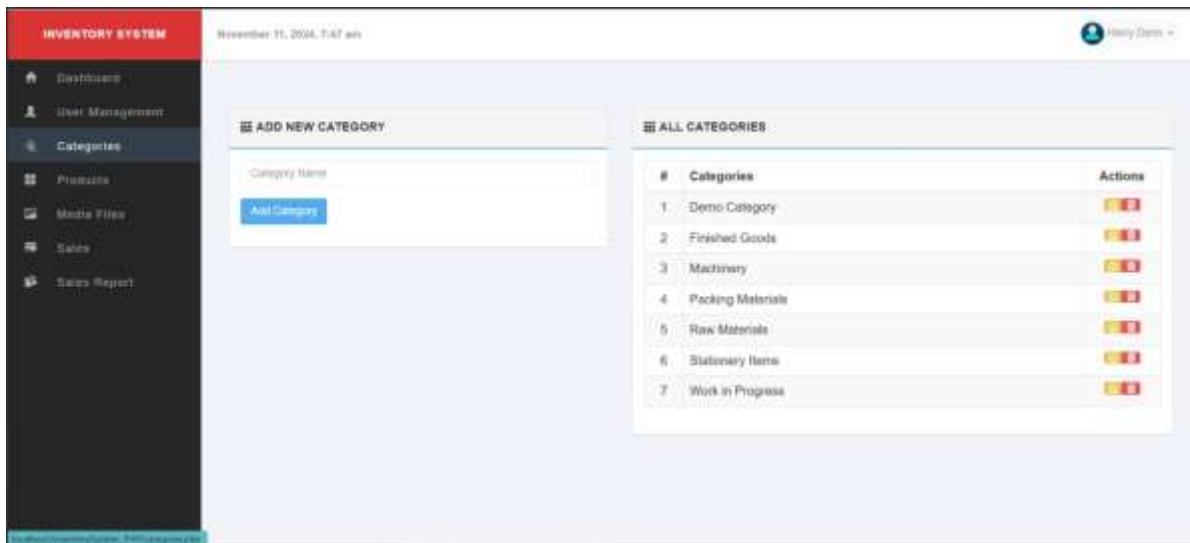
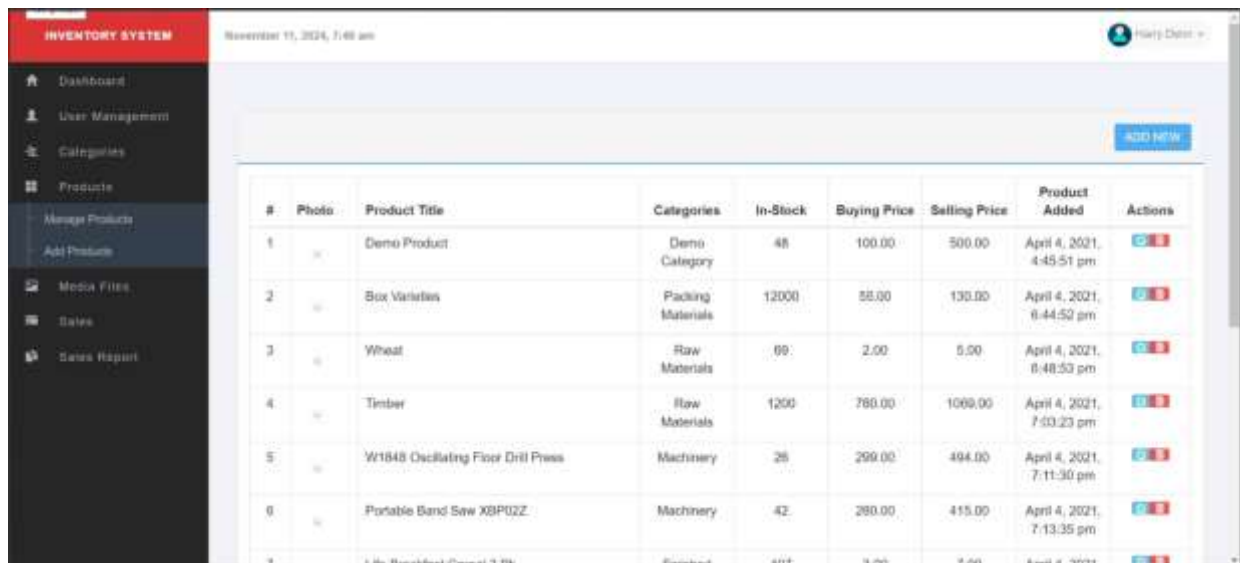


Fig 5.1.2 – Category management

- Admins can create, update, and delete product categories.
- Each product must be associated with a specific category

3. Product Management:



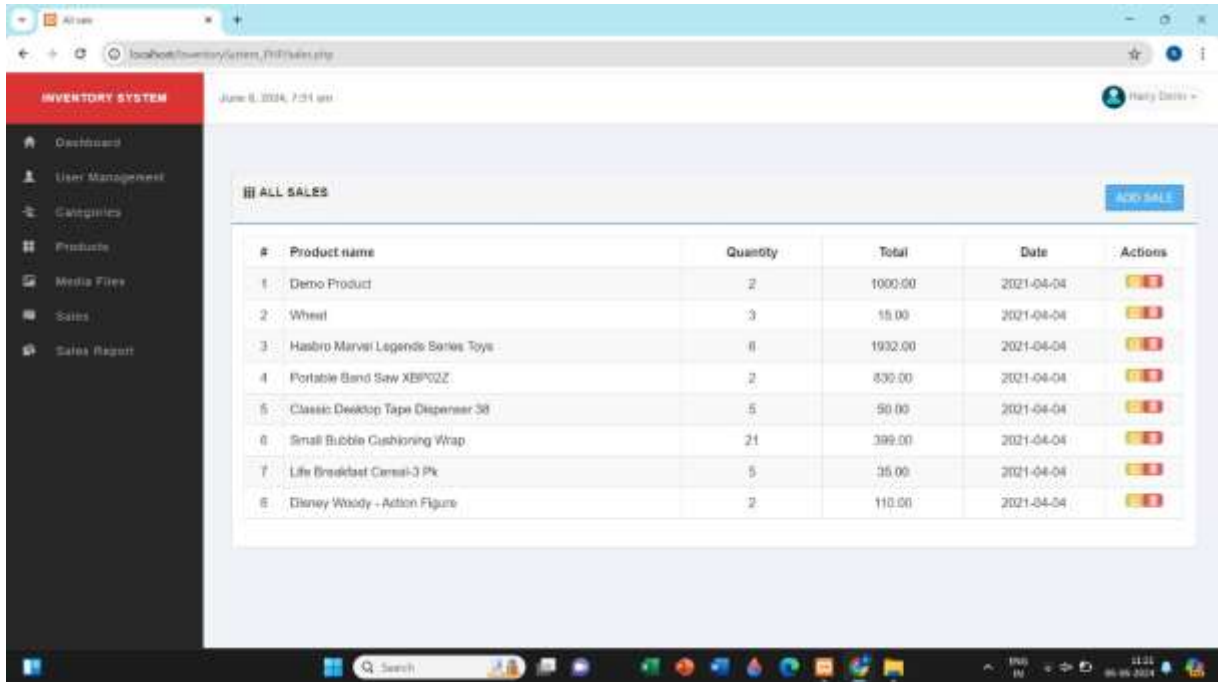
The screenshot displays the 'INVENTORY SYSTEM' interface. On the left is a dark sidebar with navigation links: Dashboard, User Management, Categories, Products, Manage Products (highlighted), Add Products, Media Files, Sales, and Sales Report. The main area shows a table of products with columns: #, Photo, Product Title, Categories, In-Stock, Buying Price, Selling Price, Product Added, and Actions. A blue 'ADD NEW' button is in the top right. The table lists six products, including 'Demo Product', 'Box Varieties', 'Wheat', 'Timber', 'W1848 Oscillating Floor Drill Press', and 'Portable Band Saw XBP02Z'. Each row has a corresponding 'Actions' column with edit and delete icons.

#	Photo	Product Title	Categories	In-Stock	Buying Price	Selling Price	Product Added	Actions
1		Demo Product	Demo Category	48	100.00	500.00	April 4, 2021, 4:45:51 pm	
2		Box Varieties	Packing Materials	12000	55.00	130.00	April 4, 2021, 6:44:52 pm	
3		Wheat	Raw Materials	69	2.00	5.00	April 4, 2021, 6:48:53 pm	
4		Timber	Raw Materials	1200	780.00	1000.00	April 4, 2021, 7:53:23 pm	
5		W1848 Oscillating Floor Drill Press	Machinery	26	290.00	494.00	April 4, 2021, 7:11:30 pm	
6		Portable Band Saw XBP02Z	Machinery	42	280.00	415.00	April 4, 2021, 7:13:35 pm	

Fig 5.1.3 – Product management

- Admins and special users can add, update, and delete products.
- Products are categorized, and each product has details such as quantity, buying price, selling price, and associated media files.

4. Sales Management:



The screenshot displays a web application interface for an inventory system. The top navigation bar includes the title 'INVENTORY SYSTEM', the date 'June 8, 2024, 7:24 am', and a user profile 'Harry Davis'. A sidebar on the left lists various system components: Dashboard, User Management, Categories, Products, Media Files, Sales, and Sales Report. The main content area is titled 'ALL SALES' and features a table with the following data:

#	Product name	Quantity	Total	Date	Actions
1	Demo Product	2	1000.00	2021-04-04	[Edit] [Delete]
2	Wheel	3	15.00	2021-04-04	[Edit] [Delete]
3	Hasbro Marvel Legends Series Toys	6	1932.00	2021-04-04	[Edit] [Delete]
4	Portable Band Saw XBP02Z	2	836.00	2021-04-04	[Edit] [Delete]
5	Classic Desktop Tape Dispenser 38	5	50.00	2021-04-04	[Edit] [Delete]
6	Small Bubble Cushioning Wrap	21	399.00	2021-04-04	[Edit] [Delete]
7	Life Breakfast Cereal-3 Pk	5	35.00	2021-04-04	[Edit] [Delete]
8	Disney Woody - Action Figure	2	110.00	2021-04-04	[Edit] [Delete]

Fig 5.1.4 – Sales management

- Employees can record sales transactions by selecting products and entering quantities.
- The system calculates total cost, selling price, and profit margins automatically

5. Media Management:

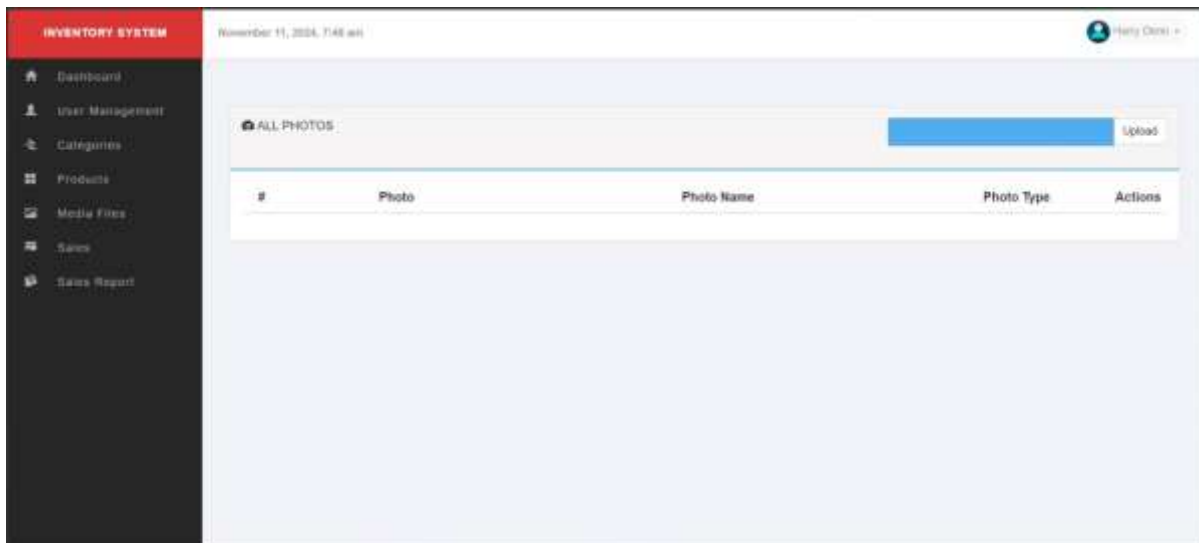
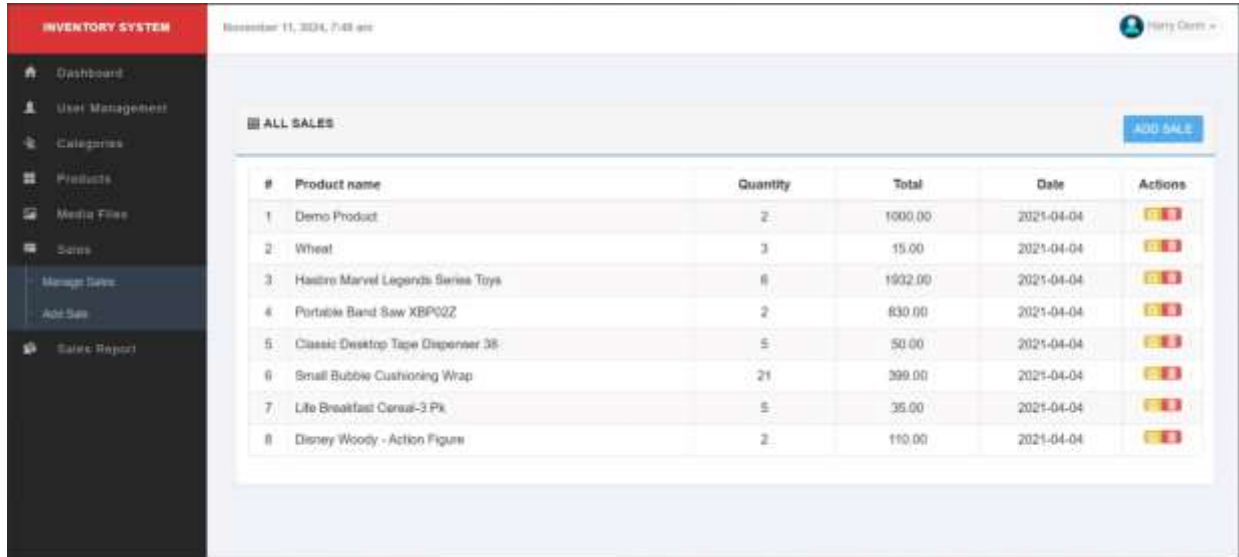


Fig 5.1.5 – Media management

- Users can upload and associate media files (e.g., product images) with products.
- Media files are stored and managed systematically.

6. Reports and Analytics:



The screenshot displays the 'INVENTORY SYSTEM' dashboard. On the left is a dark sidebar with navigation links: Dashboard, User Management, Categories, Products, Media Files, Sales, Manage Sales, Add Sale, and Sales Report. The main content area shows the date 'December 11, 2024, 7:48 am' and a user profile 'Harry Gern'. Below this is a section titled 'ALL SALES' with an 'ADD SALE' button. A table lists sales data with columns for ID, Product name, Quantity, Total, Date, and Actions. The table contains 8 rows of data.

















#	Product name	Quantity	Total	Date	Actions
1	Demo Product	2	1000.00	2021-04-04	 
2	Wheat	3	15.00	2021-04-04	 
3	Hasbro Marvel Legends Series Toys	8	1902.00	2021-04-04	 
4	Portable Band Saw XBP02Z	2	830.00	2021-04-04	 
5	Classic Desktop Tape Dispenser 38	5	50.00	2021-04-04	 
6	Small Bubble Cushioning Wrap	21	399.00	2021-04-04	 
7	Life Breakfast Cereal-3 Pk	5	35.00	2021-04-04	 
8	Disney Woody - Action Figure	2	110.00	2021-04-04	 

Fig 5.1.6 – Report and analytics

- The system generates sales reports on a daily, weekly, and monthly basis.
- Reports include details such as product names, quantities sold, total earnings, and profit margins.

7. Authentication and Authorization:

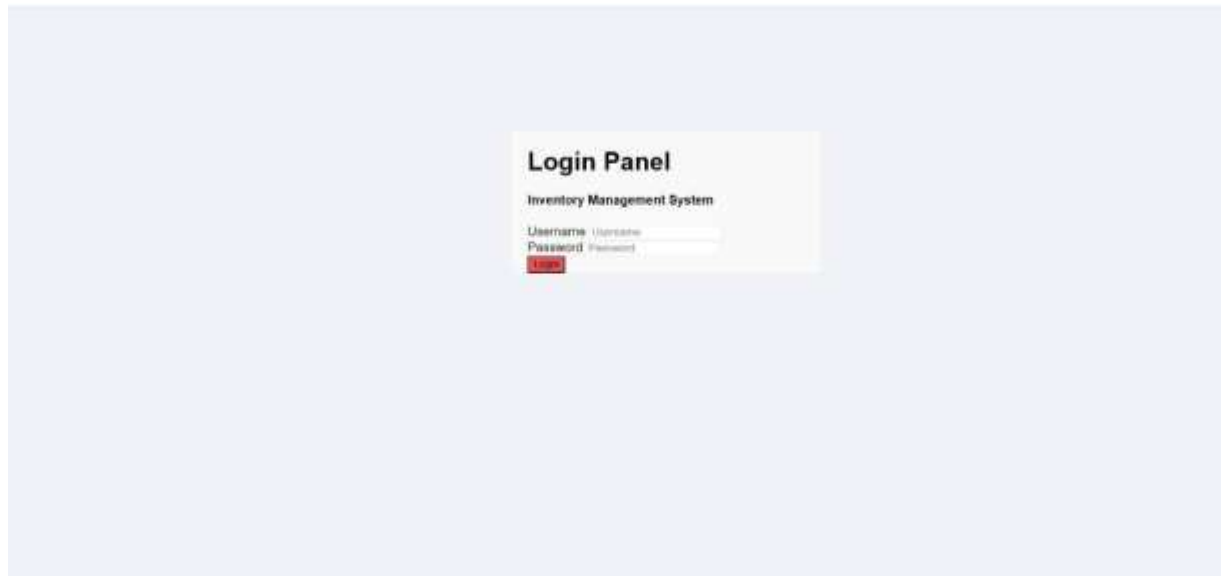


Fig 5.1.7 – Authentication and Authorization

- The system includes robust user authentication to ensure secure access.
- Different user roles have specific permissions, ensuring that users can only access functionalities relevant to their roles.

5.2 User Feedback

During the testing phase, the Inventory Management System was subjected to user feedback to evaluate its performance and usability. Here are some highlights from the feedback:

- **Ease of Use:** Users found the interface intuitive and easy to navigate. The use of Bootstrap for the front-end provided a clean and responsive design, enhancing the overall user experience.
- **Functionality:** Users appreciated the comprehensive set of features, particularly the detailed sales reports and the ability to manage products and categories efficiently.
- **Performance:** The system performed well under various scenarios, with quick response times and accurate data handling.
- **Security:** Users felt confident in the security measures implemented, including user authentication and role-based access control.

5.3 Challenges Faced During Development

Developing the Inventory Management System presented several challenges, which were addressed through careful planning and problem-solving:

1. Database Design:

- **Challenge:** Designing a scalable and efficient database schema that could handle various relationships between entities.
- **Solution:** Careful normalization of the database and creation of appropriate foreign key constraints to ensure data integrity and avoid redundancy.

2. User Authentication and Authorization:

- **Challenge:** Implementing a secure authentication system and managing different user roles with specific permissions.
- **Solution:** Utilization of PHP session management and role-based access control to ensure secure and appropriate access for each user type.

3. Front-End Design:

- **Challenge:** Creating a user-friendly and responsive interface that works well on different devices and screen sizes.
- **Solution:** Leveraging Bootstrap to design a responsive layout and conducting user testing to refine the interface.

4. Sales Report Generation:

- **Challenge:** Developing a system to generate accurate and detailed sales reports on a daily, weekly, and monthly basis.
- **Solution:** Writing efficient SQL queries to aggregate sales data and present it in a readable format in the reports.

5. Error Handling and Validation:

- **Challenge:** Ensuring robust error handling and data validation to maintain system stability and data integrity.
- **Solution:** Implementing comprehensive validation checks and error handling mechanisms throughout the application.

Overall, the development process was a valuable learning experience, and the final system met the project objectives, providing a robust solution for inventory management. Continuous feedback and iterative improvements were key to the successful implementation of the system.

CONCLUSION

In conclusion, the Inventory Management System project successfully achieved its primary objective of creating an efficient and user-friendly platform for managing products, sales, and user roles. The system's three-tier architecture, encompassing the Presentation Layer, Application Layer, and Data Layer, ensured a well-structured and scalable solution. By implementing features such as comprehensive user management, detailed sales reports, and robust authentication and authorization mechanisms, the project addressed the essential needs of inventory management in a seamless manner. The use of PHP and MySQL for backend development, along with Bootstrap for the front-end, contributed to the system's responsiveness and overall usability.

Throughout the development process, several challenges were encountered and overcome, particularly in areas of database design, user authentication, and report generation. User feedback highlighted the system's ease of use, functionality, and security, confirming that the project met its goals effectively. The iterative development and continuous improvements based on user input were crucial in refining the system. Overall, this project not only provided a practical solution for inventory management but also served as a valuable learning experience in software development, database management, and user interface design. The resulting system stands as a testament to the team's effort and dedication in creating a reliable and efficient inventory management solution.

REFERENCES

- a. Official PHP Documentation: <https://www.php.net/docs.php>
- b. Official MySQL Documentation: <https://dev.mysql.com/doc/>
- c. Official Bootstrap Documentation: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- d. MDN Web Docs for HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- e. MDN Web Docs for CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- f. MDN Web Docs for JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- g. jQuery Documentation: <https://api.jquery.com/>
- h. AJAX Introduction on W3Schools: https://www.w3schools.com/xml/ajax_intro.asp
- i. CodexWorld Rating System: <https://www.codexworld.com/star-rating-system-in-php-mysql-jquery/>
- j. PHP File Upload Documentation: <https://www.php.net/manual/en/features.file-upload.php>
- k. OWASP Top Ten Security Risks: <https://owasp.org/www-project-top-ten/>
- l. Prepared Statements in PHP: <https://www.php.net/manual/en/mysqli.quickstart.prepared-statements.php>
- m. Nielsen Norman Group: <https://www.nngroup.com/>
- n. Smashing Magazine: <https://www.smashingmagazine.com/>
- o. Visual Studio Code: <https://code.visualstudio.com/>
- p. XAMPP for PHP and MySQL: <https://www.apachefriends.org/index.html>
- q. Stack Overflow: <https://stackoverflow.com/>
- r. W3Schools: <https://www.w3schools.com/>
- s. GitHub for version control: <https://github.com/>

