

# **INVENTORY MANAGEMENT SYSTEM**

## **A MINI PROJECT REPORT**

**Submitted by**

**RISHI BALA P**

**220701224**

**SANJAY S**

**220701248**

**RAHUL KISHORE S**

**220701213**

In partial fulfillment for the award of the degree of

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE**

**RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)**

**THANDALAM**

**CHENNAI-602105**

**2023 – 24**

## **BONAFIDE CERTIFICATE**

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

“ **RISHI BALA P (220701224), SANJAY S (220701248), RAHUL KISHORE S (220701213)** ”

who carried out the project work under my supervision.

Submitted for the Practical Examination held on \_\_\_\_\_

**SIGNATURE**

**Dr. R. SABITHA**

**Professor and II Year Academic Head,  
Computer Science and Engineering,  
Rajalakshmi Engineering College  
(Autonomous),  
Thandalam, Chennai - 602 105**

**SIGNATURE**

**MRS. K. MAHESMEENA**

**Assistant Professor,  
Computer Science and Engineering,  
Rajalakshmi Engineering College,  
(Autonomous),  
Thandalam, Chennai - 602 105**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

# ABSTRACT

The Inventory Management System is a web-based application that provides a platform for businesses to manage their inventory efficiently. The application aims to simplify the process of inventory management and provide a user-friendly interface for businesses to access their inventory data. The Inventory Management System is divided into three modules, namely, Admin, Special User, and User Login. The admin has full control of the system, while the special user and user have their own respective roles in this system. The application is developed using PHP, MySQL, Bootstrap, HTML, CSS, and JavaScript.

The Inventory Management System provides a secure user authentication system, which ensures that only authorized users can access the application. The application enables businesses to manage their products and categories efficiently. Businesses can add, edit, and delete products and categories as per their requirements. The application also enables businesses to manage their sales records efficiently. Businesses can add, edit, and delete sales records as per their requirements. The application generates sales reports on a daily, weekly, and monthly basis, which helps businesses to analyze their sales data and make informed decisions.

The Inventory Management System has several benefits for businesses. It simplifies the process of inventory management, as businesses can manage their products, categories, and sales records efficiently on a single platform. The platform also provides a source of data analytics, as businesses can analyze their sales data and make informed decisions. The Inventory Management System is also a collaborative platform, which enables businesses to connect with their employees and manage their inventory data efficiently.

In conclusion, the Inventory Management System is a useful tool for businesses to manage their inventory efficiently. The application provides a secure user authentication system, enables businesses to manage their products, categories, and sales records efficiently, and generates sales reports on a daily, weekly, and monthly basis. The Inventory Management System has several benefits for businesses, such as simplifying the process of inventory management, providing a source of data analytics, and enabling businesses to connect with their employees and manage their inventory data efficiently.

## **TABLE OF CONTENTS**

<b>1. INTRODUCTION</b>	
1.1 INTRODUCTION.....	[5]
1.2 OBJECTIVES.....	[5]
1.3 MODULES.....	[6]
<b>2. SURVEY OF TECHNOLOGIES</b>	
2.1 SOFTWARE DESCRIPTION.....	[7]
2.2 LANGUAGES.....	[8]
<b>3. REQUIREMENT AND ANALYSIS</b>	
3.1 REQUIREMENT SPECIFICATION.....	[9]
3.2 HARDWARE AND SOFTWARE SPECIFICATION.....	[10]
3.3 ARCHITECTURE DIAGRAM.....	[11]
3.4 ER DIAGRAM.....	[13]
<b>4. PROGRAM CODE.....</b>	<b>[14]</b>
<b>5. RESULTS AND DICUSSION</b>	
5.1 FUNCTIONALITY OF THE PROJECT.....	[18]
5.2 USER FEEDBACK.....	[24]
5.3 CHALLENGES FACED DURING DEVELOPMENT.....	[25]
<b>6. CONCLUSION.....</b>	<b>[26]</b>
<b>7. REFERENCES.....</b>	<b>[27]</b>

# **1.INTRODUCTION**

## **1.1 INTRODUCTION**

The Inventory Management System is a critical aspect of any business that deals with the production, storage, and distribution of goods. It involves the tracking of inventory levels, orders, sales, and deliveries to ensure that the right products are available at the right time and in the right quantity. Manual inventory management methods are time-consuming, error-prone, and inefficient, which can lead to stockouts, overstocking, and increased holding costs. Therefore, businesses are increasingly adopting inventory management software to automate and streamline their inventory management processes.

The Inventory Management System software is a computer-based application that provides a platform for businesses to manage their inventory efficiently. It enables businesses to track their inventory levels in real-time, automate their order and delivery processes, and generate accurate inventory reports and analytics. The software is designed to meet the specific needs of businesses, ranging from small and medium-sized enterprises to large corporations. It is a crucial tool for businesses that aim to improve their inventory accuracy, reduce their holding costs, and enhance their customer satisfaction.

## **1.2 OBJECTIVES**

The main objectives of the Recipe Cookbook are:

- To provide a platform for users to store, search, and share recipes.
- To simplify the process of managing recipes.
- To provide a user-friendly interface for users to access their favorite recipes.
- To enable users to rate and review recipes.
- To enable users to add their own recipes.

## 1.3 MODULES

The Recipe Cookbook is divided into three modules, namely:

- Admin: The admin has full control of the system, where he/she can manage all inventory records. The admin can manage categories, products with their respective sales information. Also, this inventory system generates sales reports on basis of monthly, daily, and weekly reports.
- Special User: The special user can only have access to the product management section and media section.
- User: The user (Employee account) has access to manage sales records and view sales reports. Sales report includes monthly, weekly, and daily reports as mentioned earlier above.

## **2.SURVEY OF TECHNOLOGIES**

### **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.



## **3.REQUIREMENTS AND ANALYSIS**

### **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 REQUIREMENTS**

### **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

### 3.3 ARCHITECTURE DIAGRAM

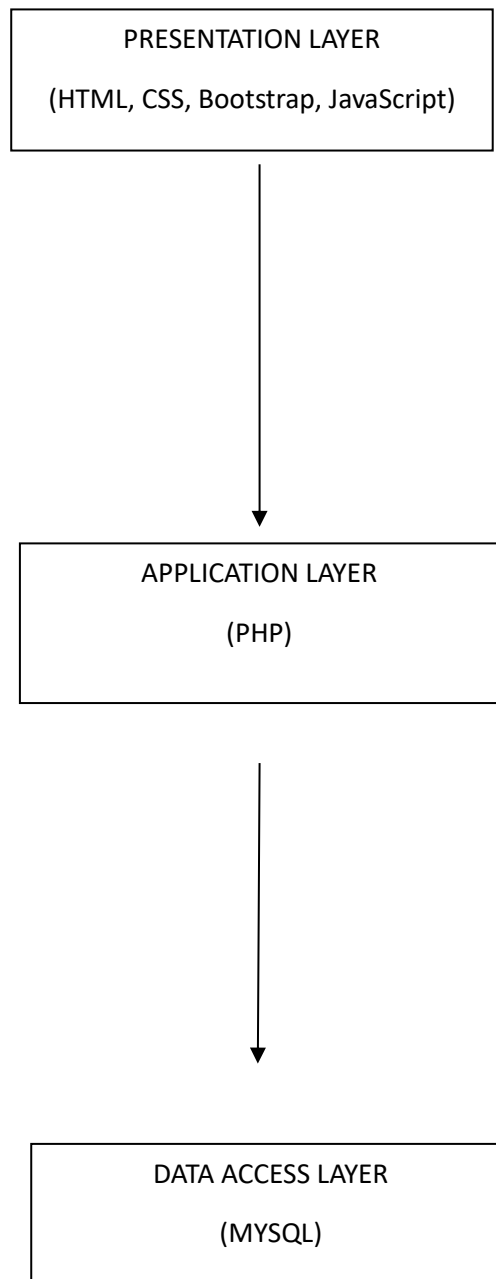


Fig 3.3.1 – Architecture Diagram

**1. Presentation Layer:**

- Responsible for the user interface.
- Technologies: HTML, CSS, Bootstrap, JavaScript

**2. Application Layer:**

- Handles the business logic of the application.
- Technologies: PHP

**3. Data Layer:**

- Manages data storage and retrieval.
- Technologies: MySQL

### 3.4 ER DIAGRAM

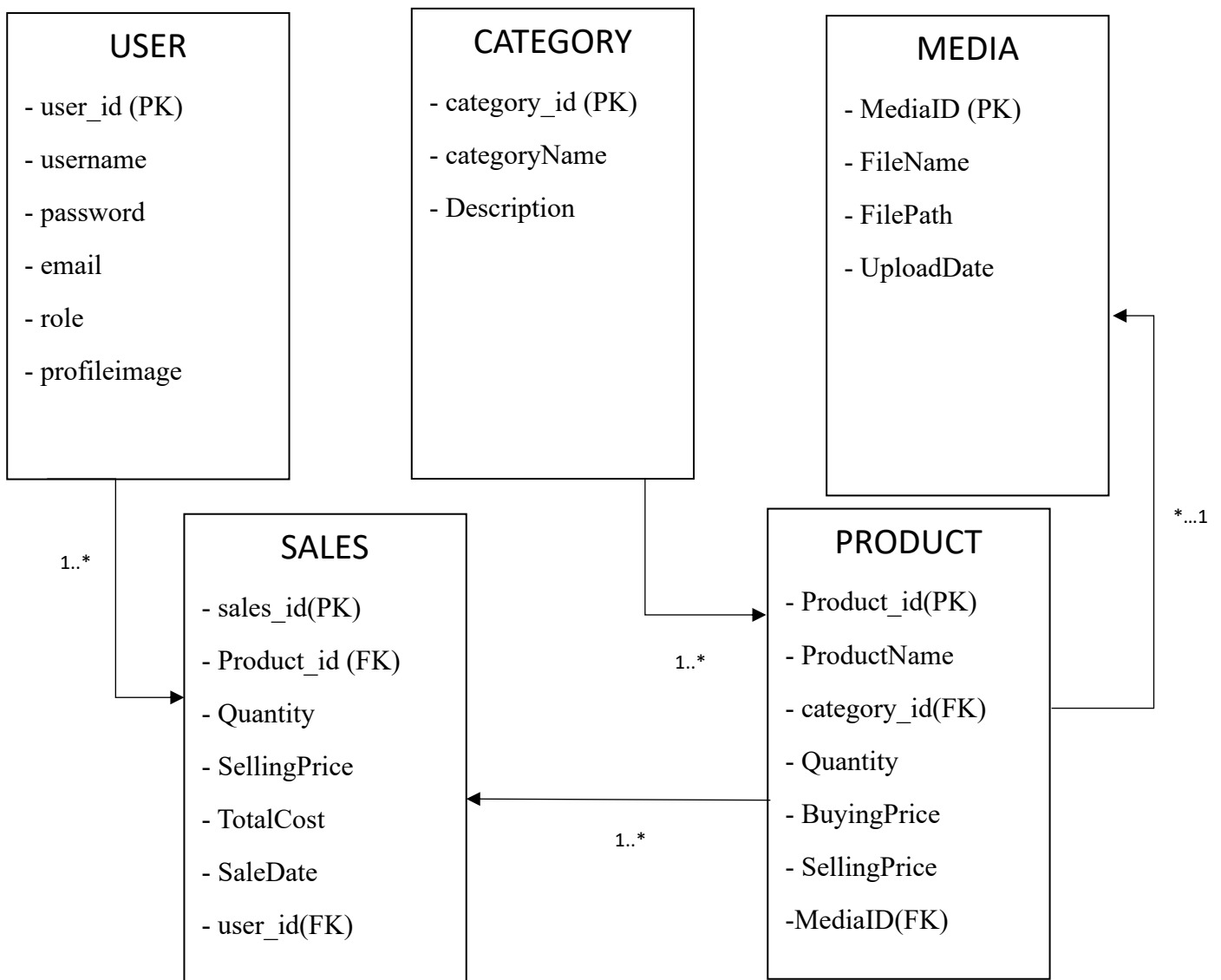


Fig 3.4.1 – ER Diagram

## 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="text" class="form-control" name="username"
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>

```

```

</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'); ?>

```

# 5.RESULTS 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:

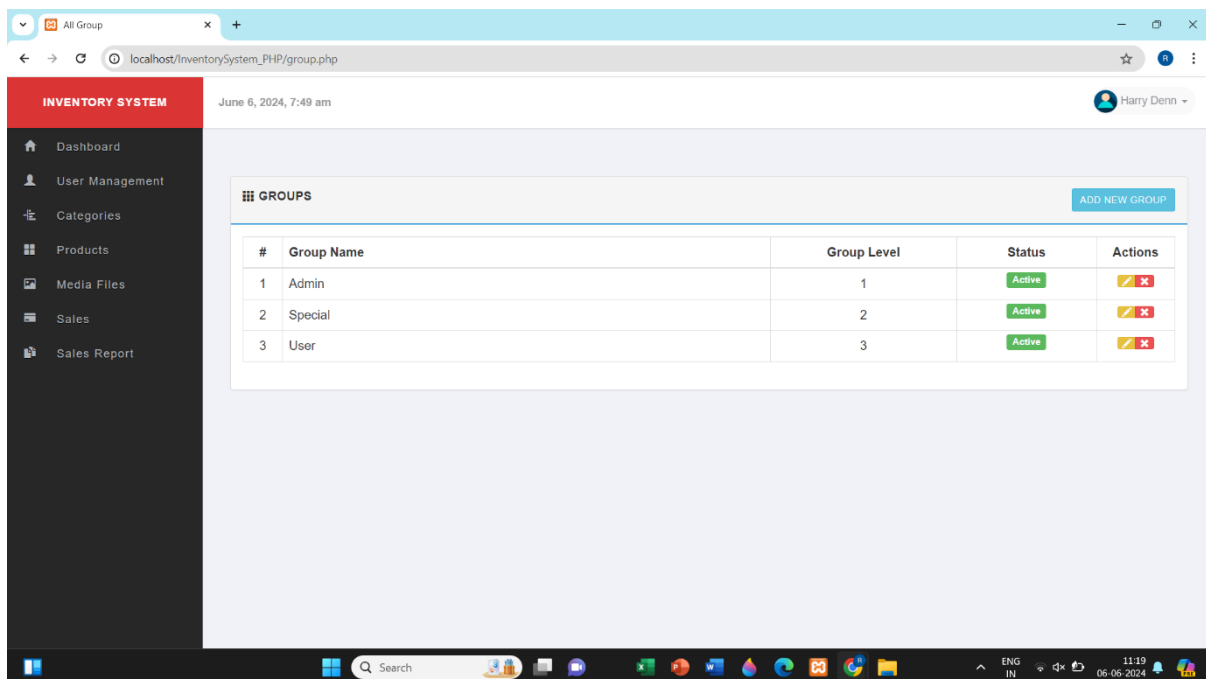


Fig 5.1.1 – User Management

- **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:

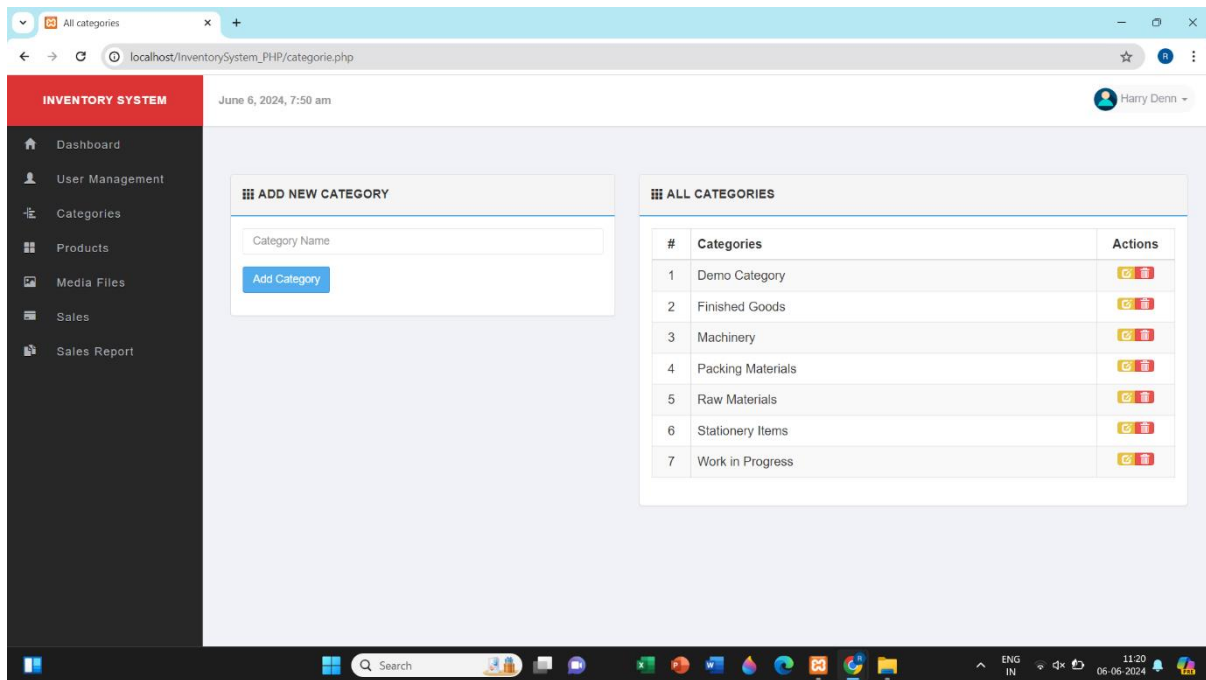


Fig 5.1.2 – Category management

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

### 3. Product Management:

INVENTORY SYSTEM June 6, 2024, 7:50 am Harry Denn

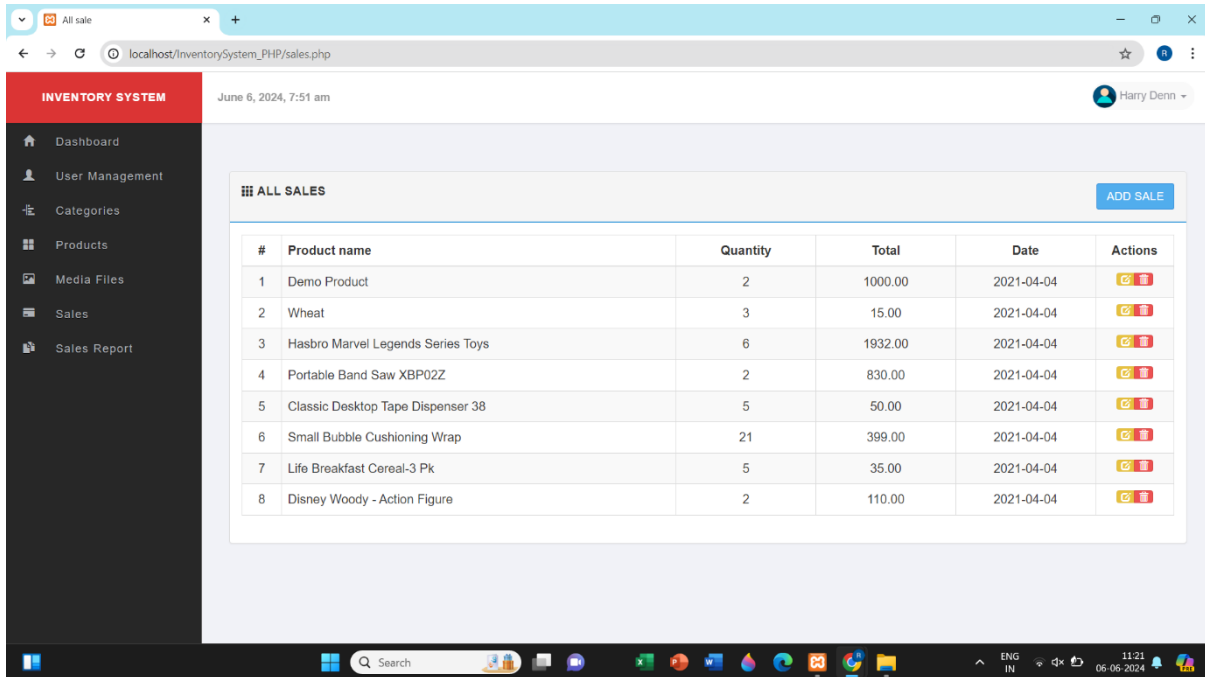
ADD NEW

#	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	1069.00	April 4, 2021, 7:03:23 pm	
5		W1848 Oscillating Floor Drill Press	Machinery	26	299.00	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	
7		Life Breakfast Cereal-3 Pk	Finished Goods	107	3.00	7.00	April 4, 2021, 7:15:38 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:



INVENTORY SYSTEM June 6, 2024, 7:51 am Harry Denn

ALL SALES ADD SALE

















#	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	6	1932.00	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 5.1.4 – 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:

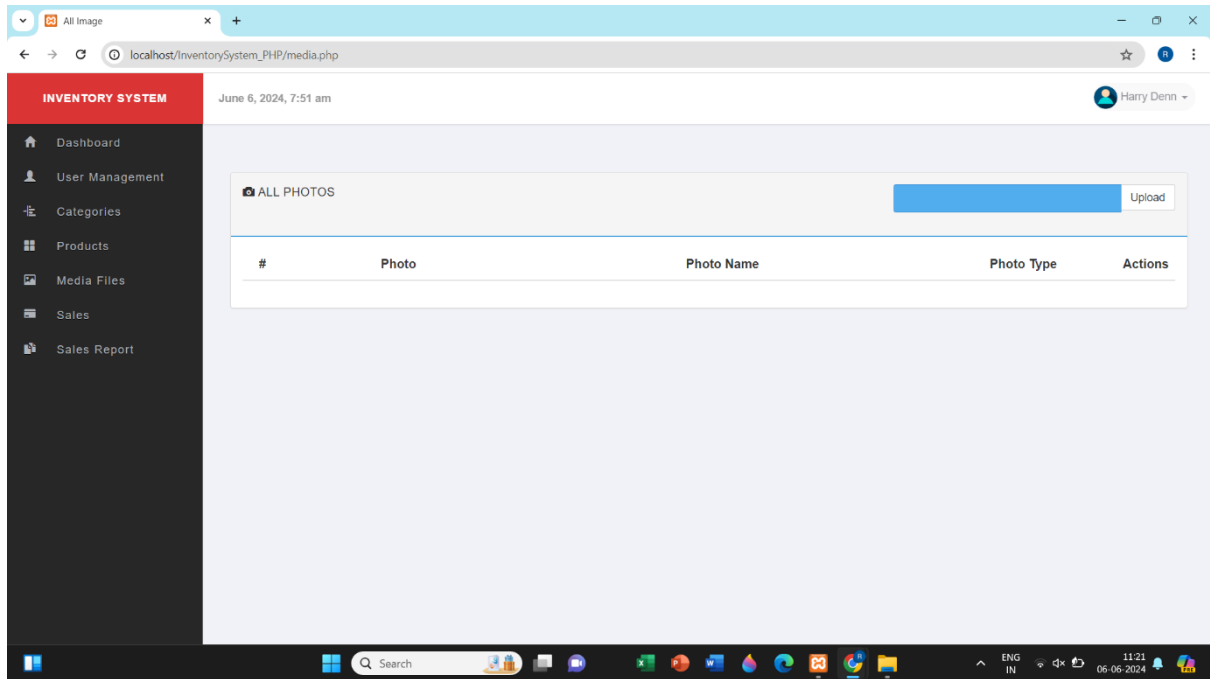


Fig 5.1.5 – Media management

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

## 6. Reports and Analytics:

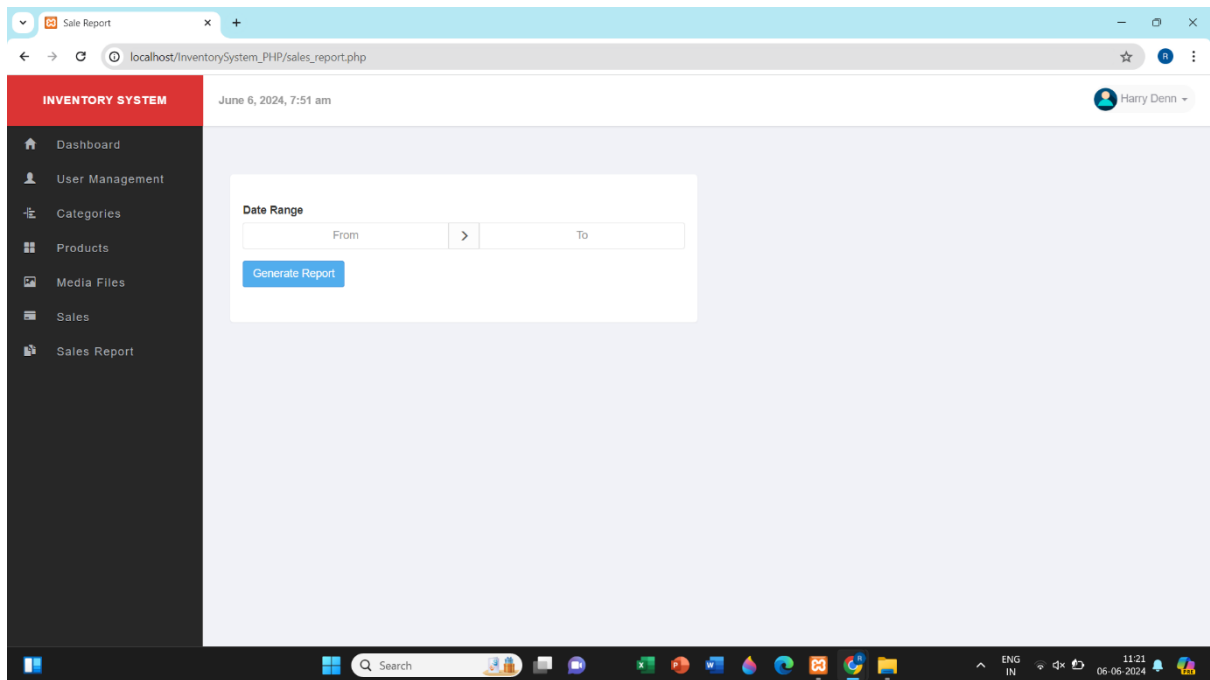


Fig 5.1.6 – Report and analytics

- 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:

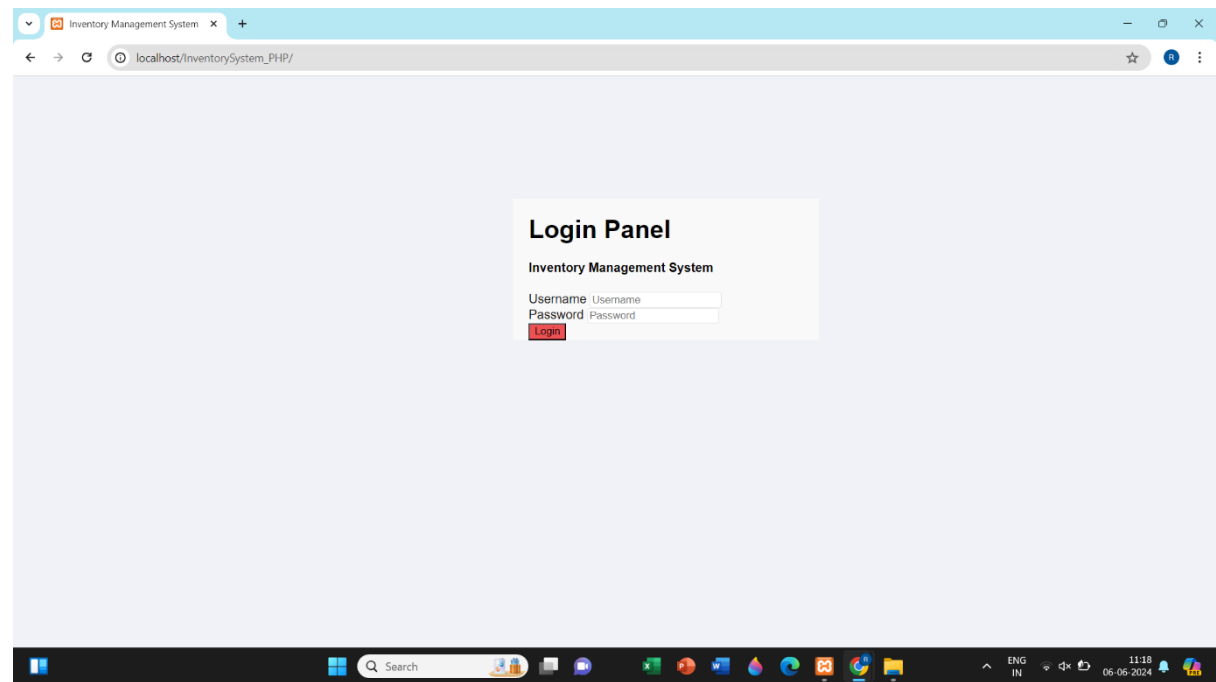


Fig 5.1.7 – Authentication and Authorization

- 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.

## 6.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.

## 7. 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](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/>