**Inventory Management System**

**FINAL YEAR PROJECT**

**JENISHA MUNIKAR**

**LC00017000520**

**TEXAS COLLEGE OF MANAGEMENT & IT**





**DEPARTMENT OF INFORMATION TECHNOLOGY**

**LINCOLN UNIVERSITY COLLEGE**

**June 5th , 2021**

**Declaration**

I hereby declare that the project work entitled “INVENTORY MANAGEMENT SYSTEM” submitted to the Faculty of Science, Lincoln University College, Kathmandu is an original piece of work under the supervision of Mr. Durganand Panjiyar and is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Technology (BIT). This project work report has not been submitted to any other university or institution for the award of any degree.

Signature:  
Name of Student:   
Date:

**Supervisor’s Recommendation**

The project work report entitled ‘INVENTORY MANAGEMENT SYSTEM’ submitted by JENISHA MUNIKAR of Texas College of Management and IT, is prepared under my supervision as per the procedure and format requirements laid by the Faculty of Science, Lincoln University College, as partial fulfillment of the requirements for the degree of Bachelors of Information Technology (BIT). I, therefore, recommend the project work report for evaluation.

Signature:

Name of Supervisor:

Date:

**Acknowledgement**

I would like to express my special gratitude towards Lincoln University College and Texas College of Management and IT for providing me a golden opportunity to work on the Final Year Project which also helped me strengthen my technical and research skills.

I would also like to offer my special thanks to my supervisors, lecturers and tutors for giving proper assistance, support, encouragement and useful critiques of this project work during the project time period. The provision of all the required resources for the project by my college is greatly appreciated. I would finally like to thanks all the college team members who helped me during this project.

**Abstract**

The project is aimed to build a web application named ‘Inventory Management System’ which helps in managing the inventory of any small grocery stores. The project ‘Inventory Management System’ is the system for tracking the availability of the products to ensure whether there is enough product stock or whether there is shortage of the product, for managing the category on which the products belong to, for adding stocks of products if not available and for managing the sales of the products. This system is for those who wants to manage the inventory of their stores in a proper and efficient way, who wants to keep the details of all their products and the sales appropriately, who wants to keep the stock details of products in an organized way, who do not want to waste their time searching for products throughout the shop and checking the products availability. Through this system, all the details of the products, its availability status, its stock details and its sales details can be stored and can also be updated.

The main aim of the report is to illustrate the project’s subject matter, background knowledge of the subject, project aim and objectives, development and progress till present date. At the very beginning of this report in chapter 1, this report provides information about the introduction to subject matter, problem statement, proposed system for problem statement, scope of the project and project aim and objectives. After that, in literature review section, the background of the project has been elaborated along with the review about similar existing system and review of technical aspect. The project ‘Inventory Management System’ has been developed using PHP programming language (Laravel framework) which is explained in chapter 2.

All the methodologies considered for the development of the system has been clarified in chapter 2 along with its advantages and disadvantages. Also, the detail information about the development methodology to be implemented in the system is also presented in chapter 2. Moreover, the report also includes appendices which contains the wireframe designs, various diagrams of the system, Gantt chart, coding and testing of the system.

**TABLE OF CONTENTS**

[**CHAPTER 1: INTRODUCTION** 1](#_Toc85566612)

[1.1 Introduction to Subject Matter 1](#_Toc85566613)

[1.2 Objectives of this chapter 2](#_Toc85566614)

[1.3 Problem Statement 2](#_Toc85566615)

[1.4 Proposed system for the problem statement 2](#_Toc85566616)

[1.5 Aim and Objectives 3](#_Toc85566617)

[1.6 Scope and Boundaries 3](#_Toc85566618)

[**CHAPTER 2: LITERATURE REVIEW** 4](#_Toc85566619)

[2.1 Background 4](#_Toc85566622)

[2.2 Objectives of this chapter 5](#_Toc85566623)

[2.3 Related Work 5](#_Toc85566624)

[2.4 Tools used for development 5](#_Toc85566629)

[2.3.1. Laravel Framework 5](#_Toc85566635)

[2.3.2. MySQL 5](#_Toc85566636)

[2.3.3. XAMPP 6](#_Toc85566637)

[2.3.4. Visual Studio Code 6](#_Toc85566638)

[2.5 Comparison of various methodologies 6](#_Toc85566639)

[2.5.1 Waterfall Model 7](#_Toc85566640)

[2.5.2 Spiral Model 8](#_Toc85566641)

[2.5.3 Prototyping Model 9](#_Toc85566642)

[2.6 Development Methodology to be implemented 9](#_Toc85566643)

[**CHAPTER 3: SYSTEM REQUIREMENT SPECIFICATION** 12](#_Toc85566644)

[3.1 Introduction 12](#_Toc85566648)

[3.1.1 Purpose 12](#_Toc85566649)

[Intended Audience and Reading Suggestions 12](#_Toc85566650)

[Intended Audience 12](#_Toc85566651)

[Reading Suggestions 13](#_Toc85566652)

[3.1.2 Project Scope 13](#_Toc85566653)

[3.2 Overall Description 13](#_Toc85566654)

[3.2.1 Product Perspective 13](#_Toc85566655)

[3.2.2 User classes and characteristics 14](#_Toc85566656)

[3.2.3 Operating Environment 14](#_Toc85566657)

[3.2.4 Design and Implementation Constraints 14](#_Toc85566658)

[3.2.5 Assumptions and Dependencies 15](#_Toc85566659)

[3.3 External Interface Requirements 15](#_Toc85566660)

[3.3.1 User Interfaces 15](#_Toc85566661)

[3.3.2 Hardware Interfaces 15](#_Toc85566662)

[3.3.3 Software Interfaces 15](#_Toc85566663)

[3.3.4 Communication Interfaces 15](#_Toc85566664)

[3.4 System Features 15](#_Toc85566665)

[ Login Feature 16](#_Toc85566666)

[3.5 Other Non-functional Requirements 16](#_Toc85566667)

[3.5.1 Performance Requirements 16](#_Toc85566668)

[3.5.2 Safety and Security Requirements 16](#_Toc85566669)

[3.5.3 Software Quality Attributes 17](#_Toc85566670)

[**CHAPTER 4: SYSTEM DESIGN** 17](#_Toc85566671)

[4.1 Introduction 17](#_Toc85566676)

[4.2 Use Case Diagram 18](#_Toc85566677)

[4.2.1 Use Case Description for actor “Admin” 19](#_Toc85566678)

[4.3 Entity Relationship Diagram 20](#_Toc85566679)

[4.4 Data Flow Diagram 20](#_Toc85566680)

[4.4.1 Context Level DFD 21](#_Toc85566681)

[4.4.2 Level 1 Data Flow Diagram 21](#_Toc85566682)

[4.4.3 Level 2 Data Flow Diagram 21](#_Toc85566683)

[4.5 Architecture Diagram 21](#_Toc85566684)

[**CHAPTER 5: SYSTEM DEVELOPMENT** 22](#_Toc85566685)

[5.1 Introduction 22](#_Toc85566686)

[5.2 Code Screenshots 22](#_Toc85566687)

[5.2.1 Development of Front home page 22](#_Toc85566688)

[5.2.2 Development of Admin panel 27](#_Toc85566689)

[**CHAPTER 6: SYSTEM TESTING** 27](#_Toc85566690)

[6.1 Purpose of System Testing 28](#_Toc85566692)

[6.2 Objectives of System Testing 28](#_Toc85566693)

[6.3 Importance of System Testing 28](#_Toc85566694)

[6.4 Overall Test Cases 28](#_Toc85566695)

[6.5 Validation Testing 29](#_Toc85566696)

[6.6 Test Case for Main Page 33](#_Toc85566697)

[**CHAPTER 7: System Evaluation** 34](#_Toc85566698)

[7.1 Introduction 34](#_Toc85566699)

[7.2 Evaluation Overview 34](#_Toc85566700)

[7.3 Evaluation of Project Deliverables 34](#_Toc85566701)

[7.4 System Evaluation 34](#_Toc85566702)

[7.5 Criteria 34](#_Toc85566703)

[7.6 Evaluation of Project Practices 34](#_Toc85566704)

[7.7 Conclusion 34](#_Toc85566705)

[**CHAPTER 8: Project Conclusion** 34](#_Toc85566706)

[8.1 Introduction 34](#_Toc85566707)

[8.2 Project Achievement 34](#_Toc85566708)

[8.3 Future Escalation 34](#_Toc85566709)

[8.4 Critical Appraisal 34](#_Toc85566710)

[8.5 Personal Achievement 34](#_Toc85566711)

[8.6 Summary 34](#_Toc85566712)

[**CHAPTER 9: Conclusion** 34](#_Toc85566713)

[**CHAPTER 10:** **References** 35](#_Toc85566714)

**TABLE OF FIGURES**

[Figure 1: Showing various types of software methodologies (Wikimedia Foundation, Inc., 2021) 7](#_Toc85615760)

[Figure 2: Use case diagram of Inventory Management System 19](#_Toc85615761)

[Figure 3: ER diagram of Inventory Management System 21](#_Toc85615762)

[Figure 4: Architecture of the system 22](#_Toc85615763)

[Figure 5: Login form page with correct username and password 42](#_Toc85615764)

[Figure 6: Test result for logging with correct admin username and password 43](#_Toc85615765)

[Figure 7: Test result for leaving the fields empty while log in 44](#_Toc85615766)

[Figure 8: Test result for entering wrong email and password 45](#_Toc85615767)

[Figure 9: Test result for checking whether error message displays when new password and confirm new password does not match 46](#_Toc85615768)

[Figure 10: Test result for checking whether error message displays when minimum password of at least 8 characters is not set in new password field 47](#_Toc85615769)

[Figure 11: Test result for checking whether password changes or not after clicking Update button. 48](#_Toc85615770)

[Figure 12: Test result for checking whether the photos of the products are displayed if photos are inserted 49](#_Toc85615771)

[Figure 13: Test result for checking if a new stock of the product added in stock page increases the quantity of the product in products page. 51](#_Toc85615772)

[Figure 14: Test result for checking whether the quantity of the product in products page changes according to the update made in the product stock quantity in stock 52](#_Toc85615773)

[Figure 15: Test result for checking if a new sales of the product added in sales page decreases the quantity of the product in products page 54](#_Toc85615774)

[Figure 16: Test result for checking whether the quantity of the product in products page changes according to the update made in the sold product quantity in sales 55](#_Toc85615775)

[Figure 17: Test result for checking whether result displays in search inventory page of admin panel after entering product name 56](#_Toc85615776)

[Figure 18: Test result for checking whether delete confirmation modal pops up after clicking delete icon 57](#_Toc85615777)

[Figure 19: Test result for checking whether admin logouts after clicking logout button\ 58](#_Toc85615778)

[Figure 20: Test result for checking whether back button of the browser redirects to login page after logout. (for preventing access to admin panel after logout) 59](#_Toc85615779)

[Figure 21: Test result for checking whether all the details along with availability status of specific product displays after searching by entering product name and search by ‘available option’ in front homepage. 60](#_Toc85615780)

[Figure 22: Test result for checking whether all the sales details of specific product displays after searching by entering product name and search by ‘sales option’ in front homepage. 61](#_Toc85615781)

# **INTRODUCTION**

## Introduction to Subject Matter

The proposed system ‘Inventory Management System’ is a web-based application which deals with the management of the inventory of small grocery stores in an appropriate and convenient way. The main aim of this project is to build an inventory management system which portrays all the information regarding the products, helps to check the availability of the products, manage the product stocks, manage the category and sales details of the products and the remaining quantity of the products.

In the context of Nepal, there are many small grocery stores that are operating. These small grocery stores purchase the products and sells these products to the customers. Most of these grocery stores keep these purchase and sales records of the products either in a paper-based form or in the spreadsheets or in the notes form in the mobile. They manage their inventory in these forms which can be quite time consuming, difficult and inappropriate as well. These stores do not have proper management of their inventory. There is no efficient way of managing their products, their product stocks, the purchase details, the sales details and checking the remaining product stock whether they have enough stock or shortage of products. Thus, with the purpose of providing these small grocery stores a proper, efficient and convenient way of managing their inventory, this project ‘Inventory Management System’ is carried out.

This system is going to provide an easy way of managing the inventory; an easy way of storing the products and sales details, managing and adding product stocks and tracking these products and handling the inventory in an organized way. The staff can view the category and its products, the availability status of the products and the sales records of the products. No login is required for viewing these pages. However, for making any changes in the system, one should be logged in. For the login purpose, a login page is created. Through this page, only admin i.e., the shopkeeper can log into the system. It is done so for the security of any misuse of the inventory and for the prevention of any change in the information regarding inventory.

## Objectives of this chapter

The objectives of this chapter are:

* To introduce the problem statement and the proposed system for the problem domain.
* To portray the aim and objectives of the project.
* To describe the scope and boundaries of the project.

## Problem Statement

## Proposed system for the problem statement

In section 1.3 Problem Statement, I have explained about the problem. So now what can be the solution for this problem statement? Do we even have any solutions for this problem statement? If yes, what is the solution? And many other such questions may arise.

The answer is yes. There is solution for the above-mentioned problem statement. The simple solution is a web-based application named ‘Inventory Management System’. In this system, I have tried to solve the problem statement and minimize the difficulties as much as possible. This system is easy to understand and use and is also not difficult in its interface; even a normal layman can use it easily. Through this system, various small grocery stores can easily manage their inventory and also in a proper and organized way. I hope this system will help various small grocery stores for managing their inventory.

In order to achieve this system, following objectives needs to fulfilled.

* Requirement gathering and analysis
* Planning the time period for the project by choosing a software development methodology.
* Research
* Creating prototype of the system
* Developing and testing the system
* Completing the requirements and documentation

## Aim and Objectives

The aim of this project is to develop an inventory management system which allows the proper management of the inventory of the small grocery stores.

The objectives of the project are as follows:

* To provide easier way to keep track of all the products for small grocery stores.
* To store all the details regarding the products in a proper and efficient way.
* To manage the stock and sales details of the products in an organized way.
* To track the availability of the products i.e., whether enough product stock is available or not, whether there is shortage of any product or not.
* To save time as there is no need searching the products in every part of the shop, counting the number of products available and checking for its availability.

## Scope and Boundaries

The scope of my project ‘Inventory Management System’ is large. There are many small grocery stores that are operating in Nepal i.e., grocery stores that is operated by single person or shopkeeper. So, my project can provide all these small grocery stores a proper and organized way of managing their inventory. All these grocery stores can use this system to keep the product details, the product stock details and the sales details in appropriate way and can also keep the track of the product availability whether there is enough product stock or whether there is shortage. Only single person i.e., the shopkeeper(admin) can add and modify all the details. Internet is required for access since it is web based.

# **LITERATURE REVIEW**



## Background

This project is about the development of a web application ‘Inventory Management System’. This system deals with the proper management of inventory, proper management of products, product stocks, category of products and sales of the products and proper tracking of product availability. In Nepal, there are various small grocery stores that are operating in every areas. Most of these stores require proper management of inventory and appropriate management of products so that they can keep track of these products. However, most of them keep their inventory records either in a paper-based form or in the spreadsheets or in the notes form in the mobile. They manage their inventory in these forms which can be quite time consuming, difficult and inappropriate as well. So, for making inventory management easier, organized and appropriate, this project is done.

The development of this system first begins with selection of one fixed prototype and then according to that prototype every other steps will be followed accordingly. PHP Laravel framework will be used for the development purpose. Every code will be written in this platform and other supportive languages such as HTML5, CSS, Bootstrap and Javascript will also be used for making the appearance of the system much better. Here, the primary role will be of analysis part as everything will be defined and planned on this part. The rest of the other such as design, coding, etc. are only the implementation of that analysis part. The system will be ready in the similar way.

## Objectives of this chapter

The main objective of this chapter is to provide detailed information of the system, to describe the project background, to portray the development tools used for the creation of the website, and to define the software methodology used in the project.

## Related Work



## Tools used for development

The following development tools are used for building this system:

* Laravel Framework
* HTML, CSS, JS
* Bootstrap
* Apache
* MySQL
* XAMPP
* Visual Studio Code



### **Laravel Framework**

Laravel is a web application framework with expressive, elegant syntax. (Laravel, 2021)

The whole system is developed using Laravel framework 8.62.0 including user interface and admin panel. The reason behind choosing Laravel is: it provides various features like clarity in documentation and MVC pattern, high-end authentication and authorization, blade template engine, query builder, unit testing, database migrations, eloquent ORM and various ready-to-use packages such as Jetstream.

### **MySQL**

MySQL is used as database management tool for developing and managing the databases for the system. It is chosen as it is easier to learn and to use, compared to other databases. Also, it is fast, open source and free database management tool.

### **XAMPP**

XAMPP server is used for performing server side or backend works of the system and for database connection as well using php my admin. It also allows programmers to view their own work on computers.

XAMPP is the most popular PHP development environment. XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. (ApacheFriends, 2021). Thus, it is used.

### **Visual Studio Code**

The overall coding part is written and performed using Visual Studio Code editor. The terminal part .i.e., running different Laravel commands for migration, creating controllers and models is also performed using VS Code Editor.

This code editor is chosen because it is free, open source and cross platform with features such as IntelliSense code completion, debugging, extensions marketplace, snippets, code refactoring, etc.

## Comparison of various methodologies

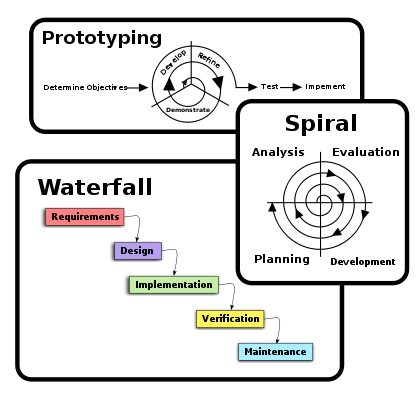


Figure 1: Showing various types of software methodologies (Wikimedia Foundation, Inc., 2021)

Above figure shows the 3 software development methodologies: waterfall, spiral and prototyping. Among these three development methodologies, I will be implementing one of the suitable method as my primary development method in my project. The detailed descriptions of these methodologies are provided below:

### **Waterfall Model**

Waterfall model is a linear-sequential life cycle model in which each phase must be completed before the next phase can begin. In waterfall model, the phases do not overlap and going to previous phase to handle any requirement changes is not possible. This model is divided into several phases. They are requirement analysis, system design, implementation, testing, integration deployment and maintenance. The outcome of one phase acts as the input for the next phase sequentially in this waterfall model. (SDLC - Waterfall Model, n.d.)

The advantages and disadvantages of waterfall model are shown below:

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Simple and easy to understand and use. * Suitable for smaller projects. * Each phase has specific deliverables and a review process. * Verification at each stage ensures early detection of errors and misunderstanding. | * High amount of risk and uncertainty in this model. * Very difficult to go back to any phase and make changes. * Not suitable for long projects and the projects where the requirements keeps changing. * Difficult to measure progress within stages. |

(SDLC - Waterfall Model, n.d.)

### **Spiral Model**

Spiral model is a combination of waterfall model and iterative model. It is a risk-driven software development process model. In spiral model, each phase begins with a design goal and ends with the client reviewing the progress. The development process starts with a small set of requirements and goes through each development phase for those set of requirements. The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase. The phases of spiral model include planning, risk analysis, engineering, development and evaluation. (Spiral Model: When to Use? Advantages & Disadvantages , 2021)

The advantages and disadvantages of spiral model are shown below:

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Additional functionality or changes can be done at a later stage. * Continuous or repeated development helps in risk management. * Development is fast and features are added in a systematic way. | * There is high risk of not meeting the schedule or time and budget. * Requires experts to analyze and evaluate the risks. * Not suitable for smaller projects. |

(Spiral Model: When to Use? Advantages & Disadvantages , 2021)

### **Prototyping Model**

A prototyping model is a software development model in which prototype is built, tested and reworked until an acceptable prototype is achieved. It is an iterative, trial and error method that takes place between developer and client. This type of model is best used when the requirements of the project are not known in detail. The several phases of prototyping model are requirement gathering and analysis, quick design, build prototype, user evaluation, refining prototype, implement product and maintain. (Prototyping Model in Software Engineering: Methodology, Process, Approach , 2021)

The advantages and disadvantages of prototyping model are shown below:

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Active involvement of user resulting in detection of error in the early stage. * Quicker user feedback helps in achieving better software development solutions. * Helps to identify the missing functionality in the system. | * Expensive, slow and time taking process model. * Cost of developing a prototype is a total waste as the prototype is thrown away in the end. * Sometimes customers may not be willing to participate in the iteration cycle for the longer time duration. |

(Prototyping Model in Software Engineering: Methodology, Process, Approach , 2021)

## Development Methodology to be implemented

Waterfall model is chosen as a development methodology for the development of the system as this model is simple to understand and use and is suitable for smaller projects. It is one of the oldest traditional model using which many software have been built till date. In waterfall model, each phase must be completed before the next phase can begin, the phases do not overlap. And it is very difficult to go back to previous phases to make any changes. The waterfall model splits the project life cycle into five phases. The phases include requirement analysis and specification, system design, implementation, testing and integration and maintenance.

Phases:

1. Requirement analysis and specification

In this phase, requirements will be collected and the collected requirements will be analyzed. The further planning for the project will also be conducted along with its scope study, study of aim and objectives and functionality of the project. Here, a document called System Requirement Specification (SRS) document will also be prepared that specifies the overall product description, its features and user details.

1. System Design

In system design phase, the requirements gathered in the SRS will be studied and designing of various diagrams such as use case diagram, entity relationship diagram, data flow diagram (context level DFD, Level 1 DFD and Level 2 DFD) will be performed. Likewise, the overall architecture of the system will also be defined in this phase.

1. Implementation

In implementation phase, the design from the system design phase will be transformed into source code. Laravel framework is used for coding purpose. Here, the works such as writing code, debugging code, documentation of code, preparing code for testing, feedback to designer and analyst, etc. will be performed and overall coding part will be completed in this phase.

1. Testing and integration

In testing phase, source code from implementation phase is tested. Various test cases will be performed in this phase to check for any errors and bugs. Test cases include checking whether the implementation matches the design or not, checking whether the implementations meets the requirements or not, validation testing and testing all the individual modules and then the complete system.

1. Maintenance

In maintenance phase, any errors that were not discovered during development and testing phase will be corrected and any changes in the requirements will be carried out. Likewise, enhancement of functionalities of the system will also be carried out in this phase.

# **SYSTEM REQUIREMENT SPECIFICATION**



## Introduction

Requirement analysis is the most important phase in the software development process. After the feasibility study is completed, this analysis phase begins. During this phase, various tasks like fact-finding, current system overview and research overview are carried out in order to determine the actual requirements for the proposed system. Based on the feasibility report and various tasks that are conducted in the analysis phase, SRS is documented for the development of the proposed system.

### **Purpose**

System Requirement Specification is a well-documented final report that must be prepared at the end of requirement analysis phase. The purpose of this document is to demonstrate a detailed description of the project. This document is generated for describing the system’s purpose, features, scope, different interfaces, functionality, operating environment and hardware and software requirements. This SRS document showcases the overall structure of the system and makes easier for the system designers and developers to understand the actual requirements.

## Intended Audience and Reading Suggestions

The purpose of this part of SRS is to provide explanation regarding the intended readers of SRS as well as reading suggestions for those intended readers.

### **Intended Audience**

The intended audience of this SRS document are as follows:

* Designer: The SRS document includes the overall description and overview of the system. According to this document, designers are responsible for developing the system. They extract the required specifications and convert them to actual development process using the SRS.

### **Reading Suggestions**

The reading suggestions for the readers of this document is mentioned in this section of the SRS document. The reading suggestions are as follows:

1. Though user can read the whole SRS document, they are suggested to read only certain sections of the SRS document such as Introduction, Overall Description and System Features, while other sections of the SRS are not required for user.
2. The developer are suggested to read the SRS in detail for understanding the actual requirements and functioning of the system.
3. The designer and coder must view the various diagrams in order to code the modules.
4. The tester must be familiar with the coding languages and visit through code section to check the required output.
5. The document writer must write a high-quality document which is easy to read and understand for everyone.

### **Project Scope**

The project ‘Inventory Management System’ is a web application which will help to manage the inventory of any small grocery stores. This project will provide an easy and convenient way for the management of inventory. The users of this system can keep track of the products, its availability, product stock and sales. The purpose of this project is to facilitate any small grocery stores who wants to manage their inventory in a proper and organized way.

## Overall Description

### **Product Perspective**

The system ‘Inventory Management System’ will demonstrate online inventory management process with features like managing product, stocks, staffs, category and sales. So, the perspective of the project will be to target those stores who want to keep their inventory records in an organized way and who want to maintain the product, stock, category, staffs and sales details appropriately.

Product Features

Questionnaire sample

### **User classes and characteristics**

Admin User

* Admin can add new category, new products, stocks and sales.
* Admin can update category, products, stocks and sales.
* Admin can remove category, products, stocks and sales.
* Admin can search the details of specific products.

Inventory Management

* View the products along with the available quantity or stock.
* Check the availability status of all products.
* Search and Check the availability status and sales details of specific products.

### **Operating Environment**

Operating environment for the website is listed below:

* Client/Server system
* Database: MySQL database
* Platform: Laravel

### **Design and Implementation Constraints**

The whole system will be developed using PHP Laravel framework i.e., both the front-end (user interface) and the back-end part for admin access. The authentication shall be done with username and password. The admin users must have valid username and password. They must be authenticated for managing the inventory. Only admin can log into the system.

### **Assumptions and Dependencies**

The following assumptions will be taken:

* Admin Users can change the contents of the system.
* Admin will have knowledge to perform data storing, editing and deleting.
* There must be internet connection for accessing the system.

## External Interface Requirements

### **User Interfaces**

This section is in pending and will be accomplished in the future.

### **Hardware Interfaces**

The system requires internet connection for performing different functions. Since the system runs over the internet, all the hardware that are needed to connect with the internet are the hardware interfaces.

### **Software Interfaces**

The languages that will be used for performing communications between the system are:

* PHP Laravel Framework
* MySQL

### **Communication Interfaces**

The system shall use HTTP protocol for communication over the internet. The system can be browsed using any web browser.

## System Features

This section provides information about the features of the proposed system.

### **Login Feature**

Admin user can login onto the system with valid username and password.

* **Product Details**

It shows the details of the products along with the category to which it belongs to, quantity, unit price and availability status .

* **Sales Details**

It shows the details of the sales of the product along with the quantity sold, total price and dates.

* **Add Stocks**

Admin user can add stocks of the products if not available or not enough stock along with the product name, quantity and date.

* **Search Inventory**

Anyone can search and view the availability and sales of specific products without login. Admin can search the details of specific products after logging into the system.

## Other Non-functional Requirements

### **Performance Requirements**

The system functionality and performance should be fluent and quick. It should not take long time to load. The user interface must be designed keeping user’s easiness in mind. The system must display proper messages in case of system error. In case of no internet connection, network error messages should be displayed.

### **Safety and Security Requirements**

The safety and security for admin user account will be taken seriously by implementing various safety and security measures. The passwords of the admin user will be encrypted before storing to the database. The information while performing POST request will be transmitted to the server without any modifications and interruption and with security.

### **Software Quality Attributes**

The system will provide reliability, accuracy and reliability for performing different functions for the user. The system will also provide user friendliness as the system will have simple and easy user interface through which user can easily use and navigate through different links at ease.

# **SYSTEM DESIGN**



## Introduction

Purpose

The purpose of this chapter ‘System Design’ is to show a visual representation of the proposed system with the help of various diagrams such as use case diagram, context diagram, data flow diagram, entity relationship (ER) diagram, etc. These diagrams are drawn to depict the design and the flow of the system. This design specification also helps the developers to visualize the flow of the entire system and develop the intended system.

Initial Assumption

* The inventory system is accessible without any username and password, but for making changes into the system, login is required.
* Admin user can change all the contents of the inventory system.

Normal flow

The admin user is in charge of creating and managing the entire proposed system Any changes or updates to the system can be done by the admin. Admin can make changes in category, products, stock, sales and on other databases. Admin can create, add, update and delete the details of these mentioned databases.

## Use Case Diagram

A use case diagram is a dynamic or behavior diagram in the Unified Modeling Language (UML) which uses actors and use cases in order to model the functionality of a system i.e., to visualize the functional requirements of a system. Here, the “actors” refer to people or entities operating under defined roles within the system and “use cases” refer to a set of actions, services and functions that the system needs to perform. (SmartDraw Software, LLC, n.d.)

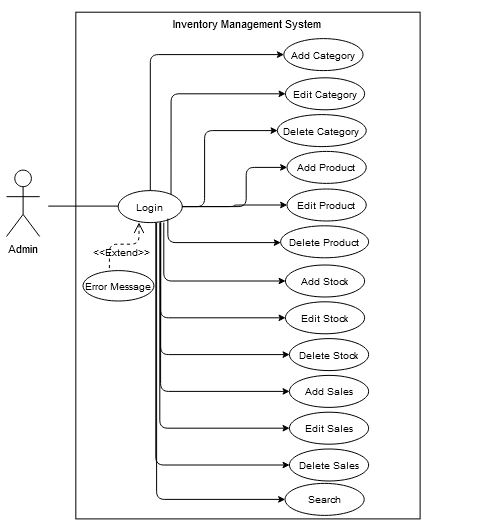


Figure 2: Use case diagram of Inventory Management System

### **Use Case Description for actor “Admin”**

|  |  |
| --- | --- |
| **Use Case** | **Use Case Description** |
| Login | Admin can login in the system |
| Add Category | Admin can add category in the system. |
| Edit Category | Admin can update category in the system. |
| Delete Category | Admin can delete category from the system. |
| Add Product | Admin can add product in the system. |
| Edit Product | Admin can update product in the system. |
| Delete Product | Admin can delete product from the system. |
| Add Stock | Admin can add stock in the system. |
| Edit Stock | Admin can update stock in the system. |
| Delete Stock | Admin can delete stock from the system. |
| Add Sales | Admin can add sales in the system. |
| Edit Sales | Admin can update sales in the system. |
| Delete Sales | Admin can delete sales from the system. |
| Search | Admin can search all inventory details of specific products. |

## Entity Relationship Diagram

An entity relationship diagram (ERD) is a type of diagram that shows the relationships of entity sets stored in a database where an entity is an object, a component of data which can have attributes and an entity set is a collection of similar entities. (SmartDraw Software, LLC, n.d.)

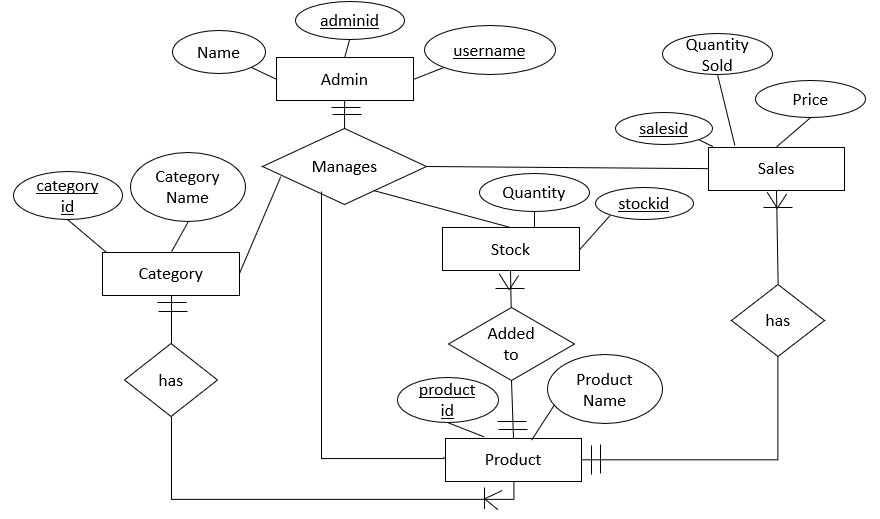


Figure 3: ER diagram of Inventory Management System

## Data Flow Diagram

A data flow diagram (DFD) is a visual representation of data or information flow within the system. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system. (What is Data Flow Diagram (DFD)? How to Draw DFD?, 2012) There are various hierarchy of DFD. The context level DFD, level 1 DFD and level 2 DFD are show below:

### **Context Level DFD**

### **Level 1 Data Flow Diagram**

### **Level 2 Data Flow Diagram**

## Architecture Diagram

Figure 4: Architecture of the system

The above diagram shows the system architecture of ‘Inventory Management System’. The system will be following client server-based architecture. If this site is hosted in any of the domain then the architecture will look like this. The inventory management users will view all the categories, products of that category, availability status and sales details of specific products but cannot modify the system. The admin will add, update and delete the contents of the system as required. All the data will be stored in the registered domain database center from which data store and data retrieve can be done. The server connected with MySQL database will be responsible for handling the request of both admin and inventory management users.

# **SYSTEM DEVELOPMENT**

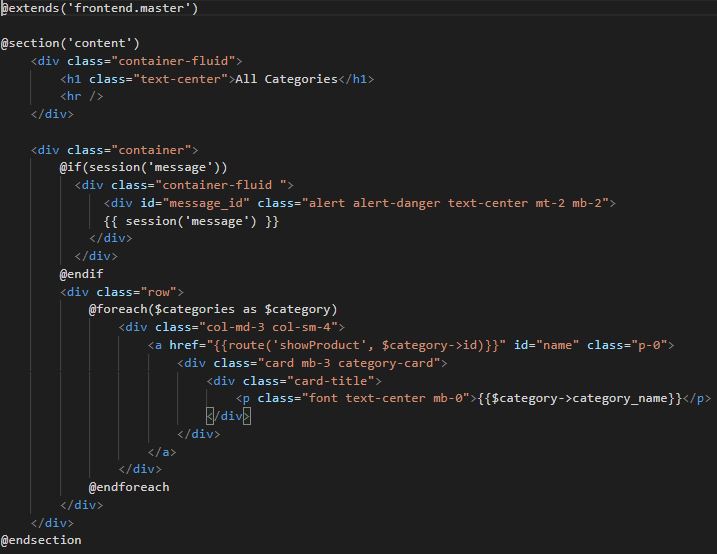
## Introduction

System development is a phase of software development where the coding part of the system is conducted. It is a phase where the actual development of the system begins. It is the longest phase. In this phase, code for developing the entire system is written in the chosen programming language.

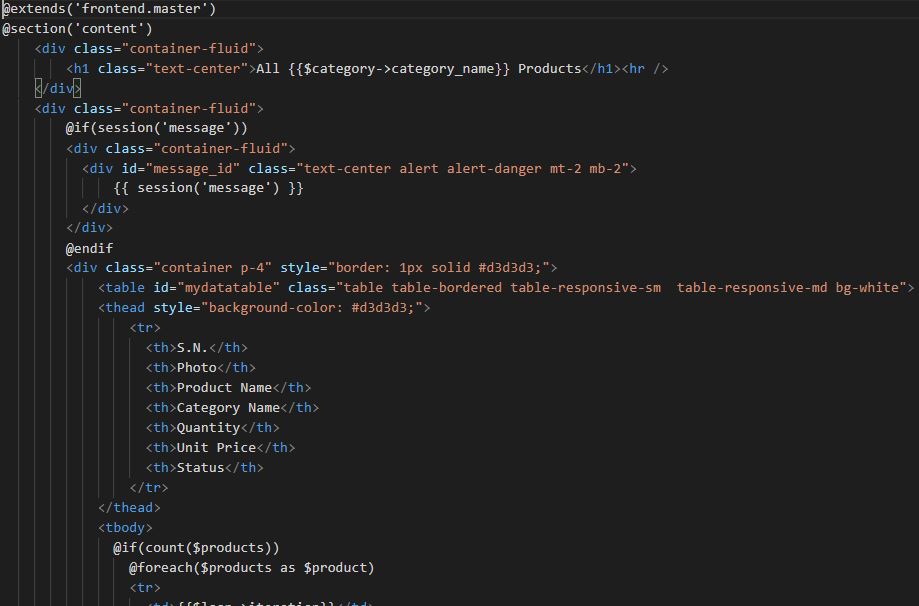
## Code Screenshots

### **Development of Front home page**

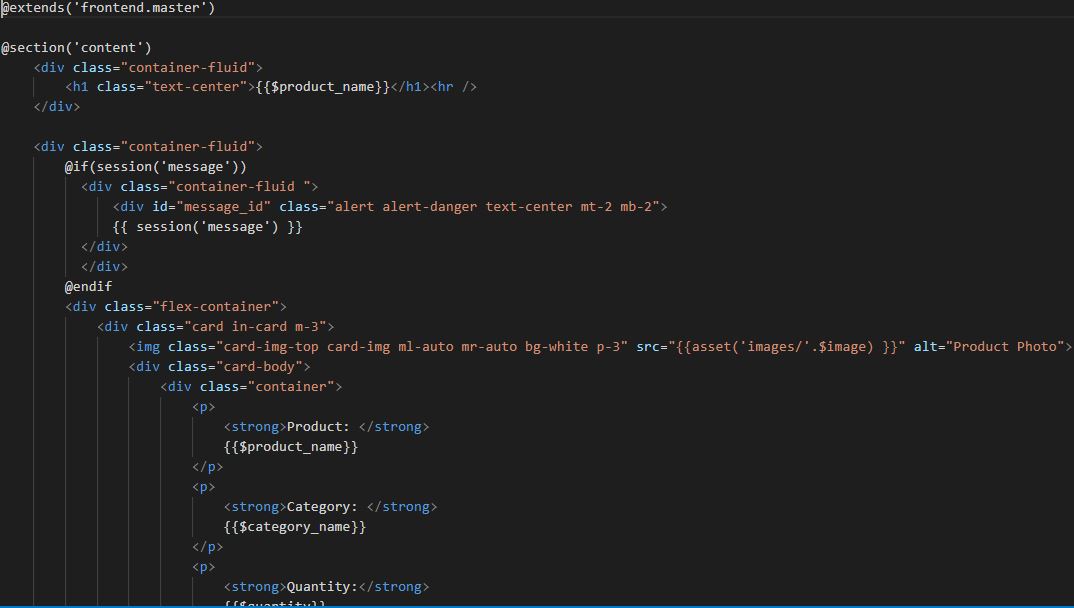
Code screenshot of category page



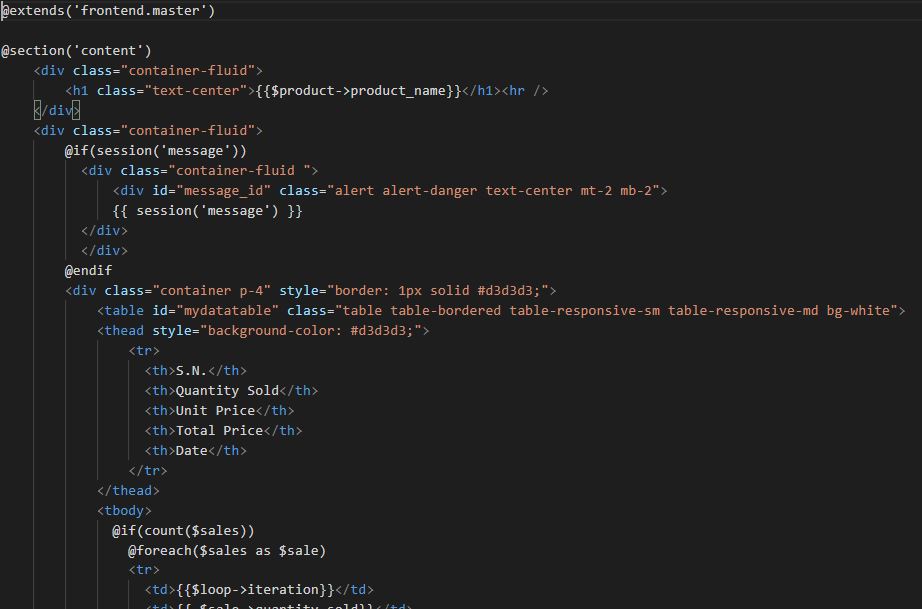
Code screenshot of products page



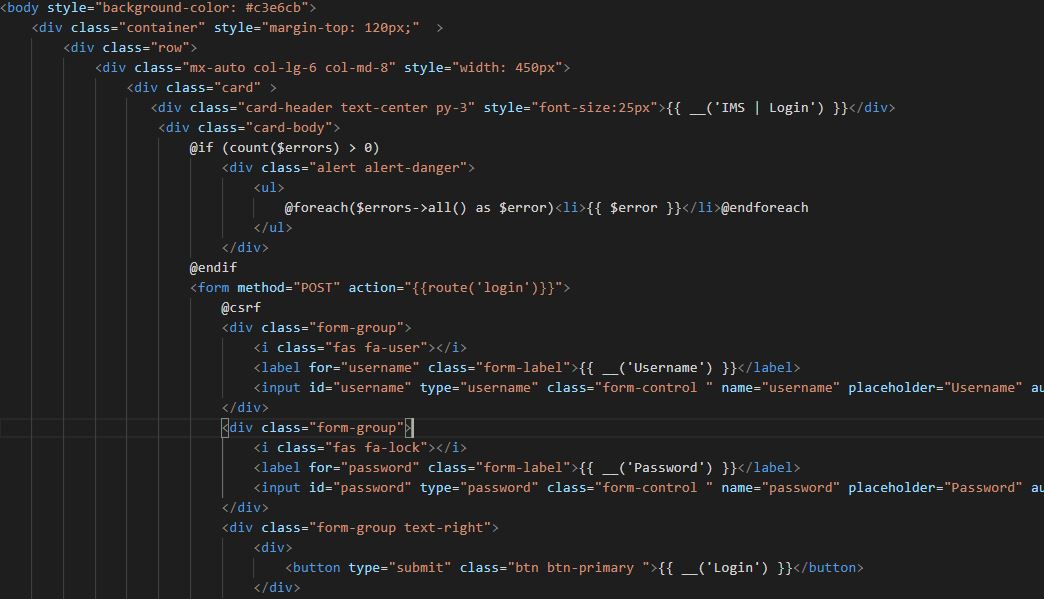
Code screenshot of viewing the availability of specific products



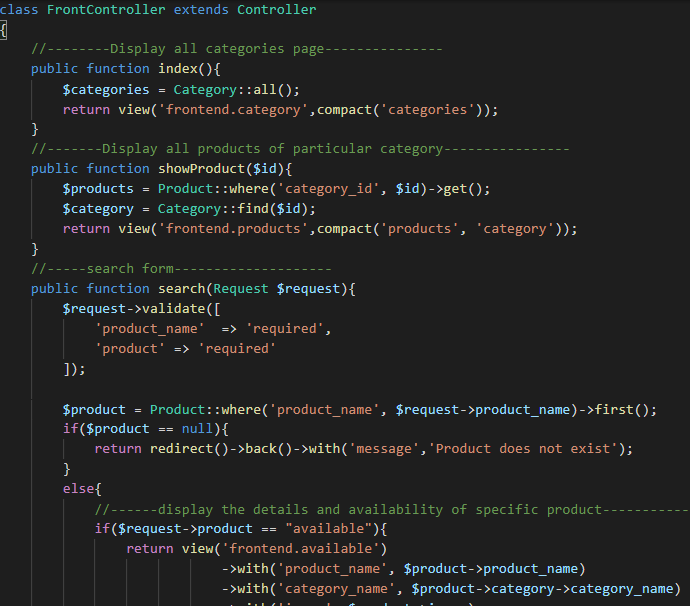
Code screenshot of viewing the sales details of specific products



Code screenshot of login page

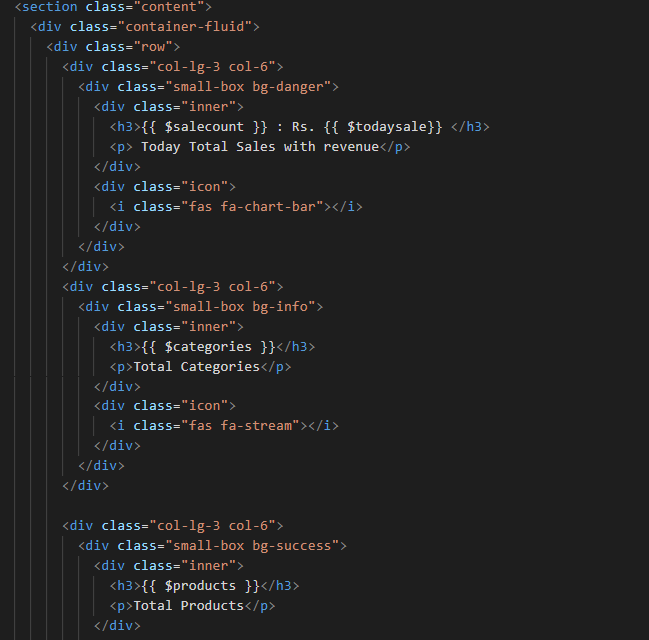


Code screenshot of front homepage controller (FrontController)

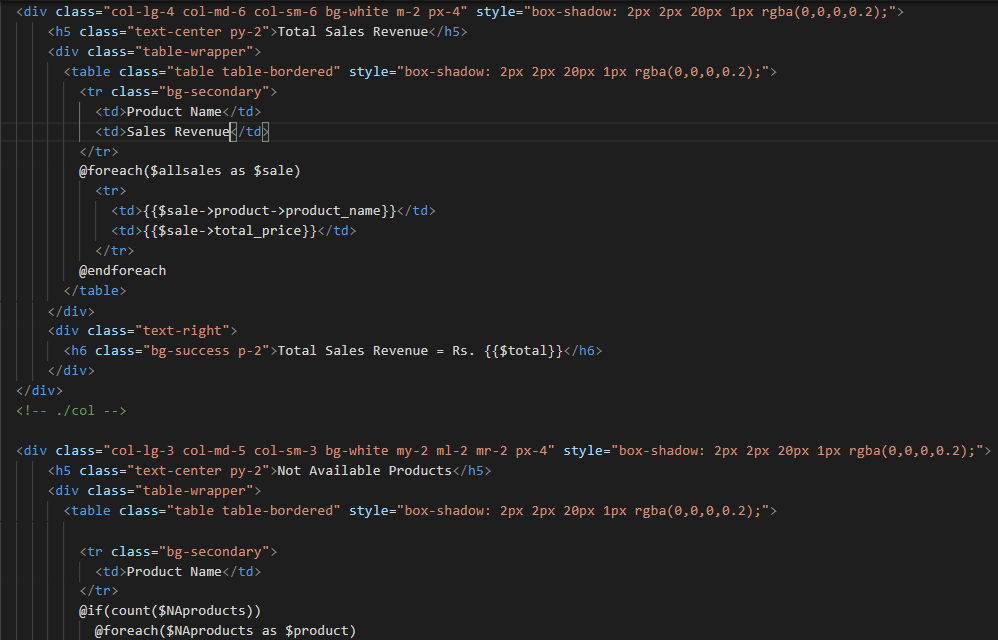


### **Development of Admin panel**

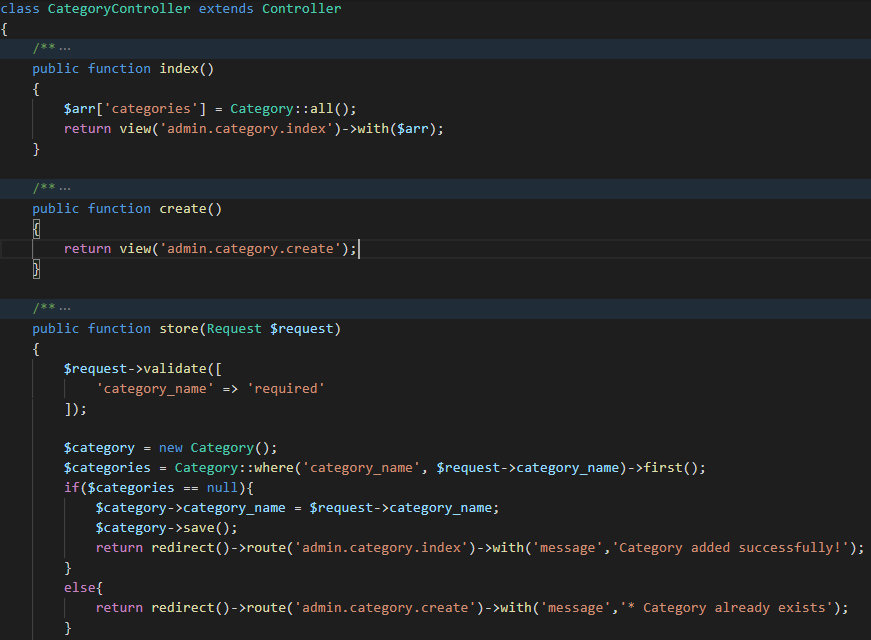
Code screenshot of dashboard page(1)



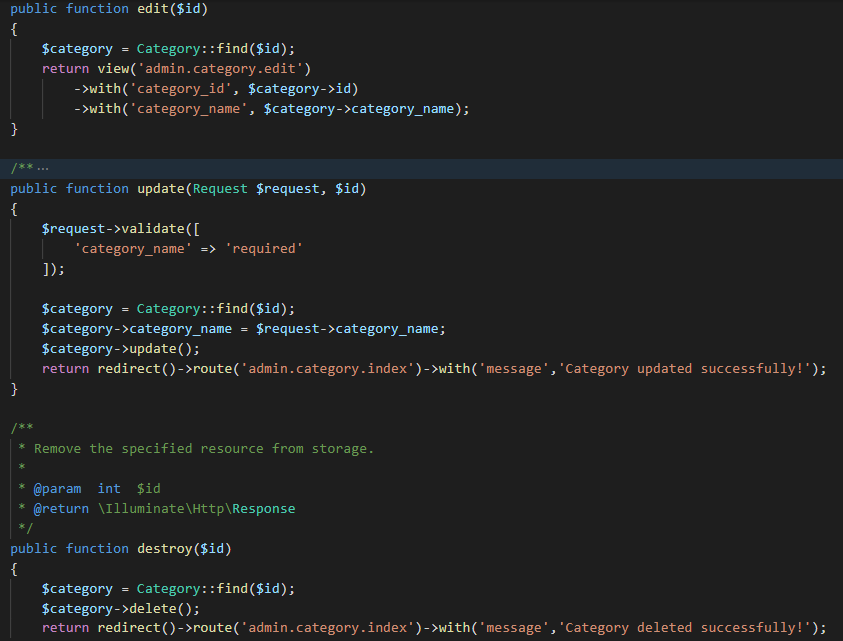
Code screenshot of dashboard page(2)



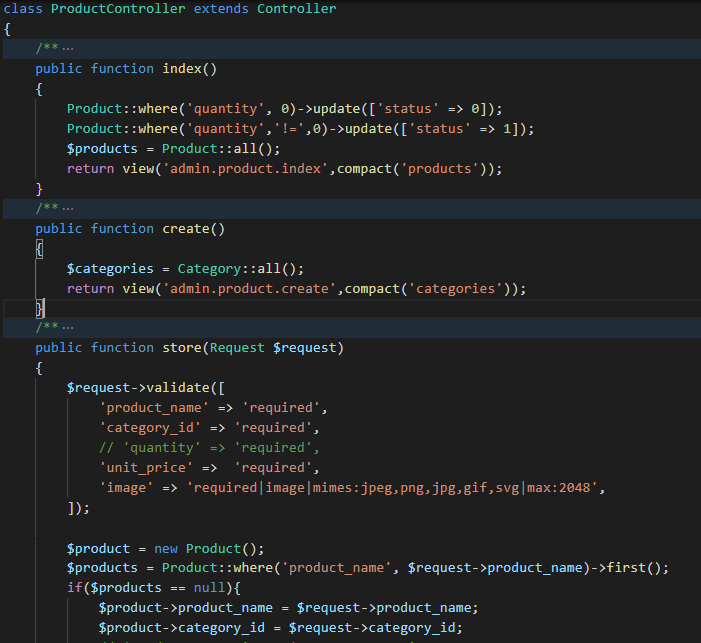
Code screenshot of CategoryController(1)



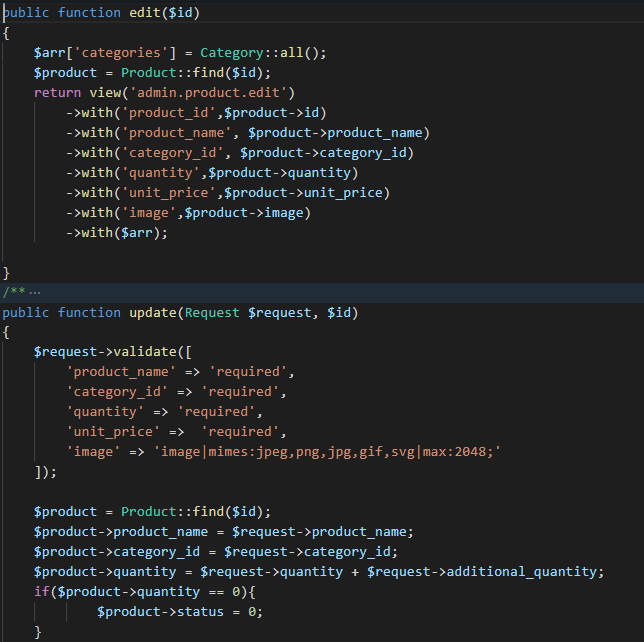
Code screenshot of CategoryController(2)



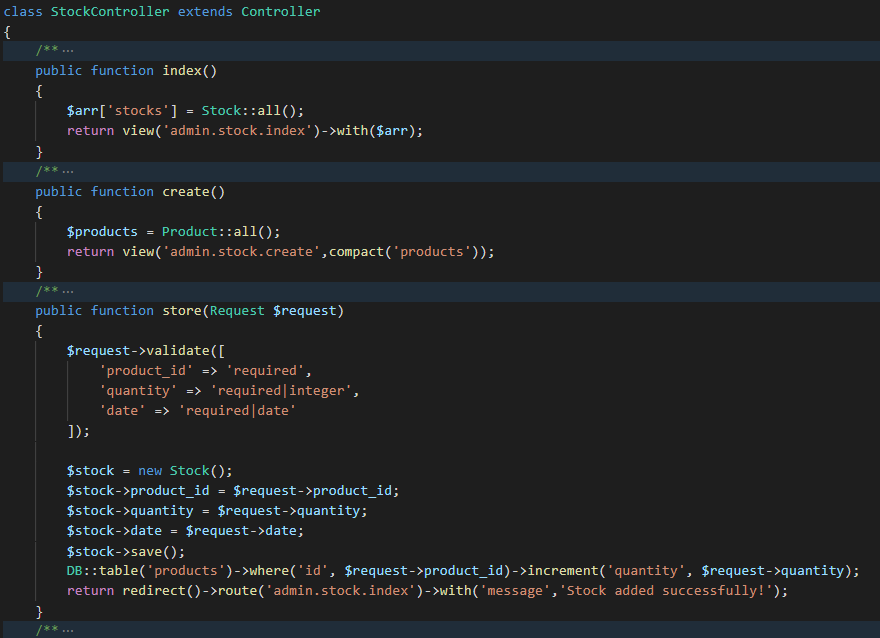
Code screenshot of ProductController(1)



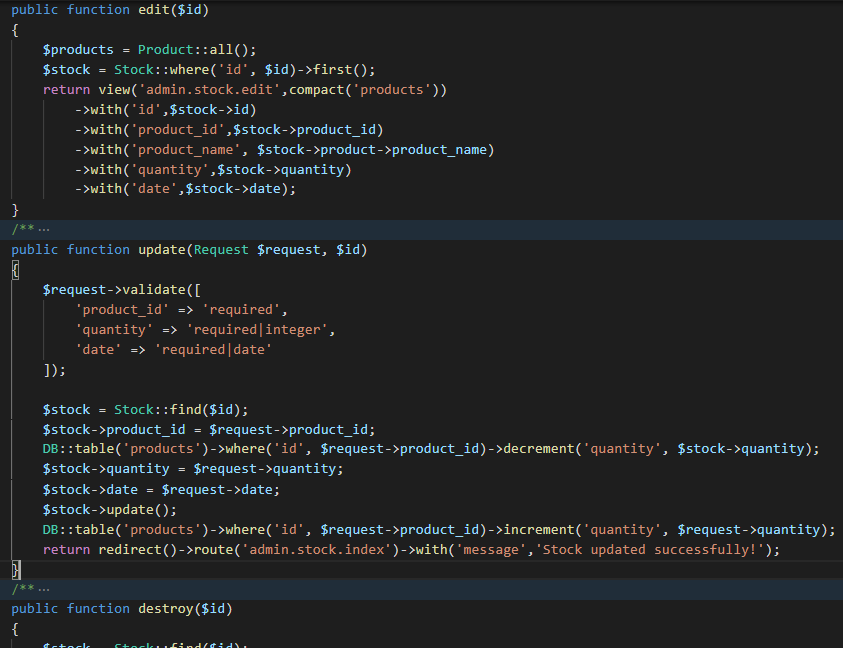
Code screenshot of ProductController(2)



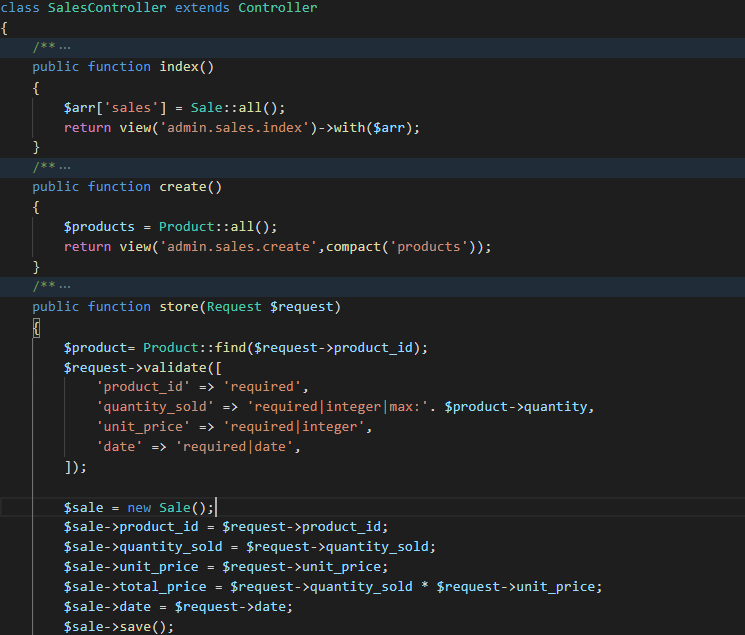
Code screenshot of StockController(1)



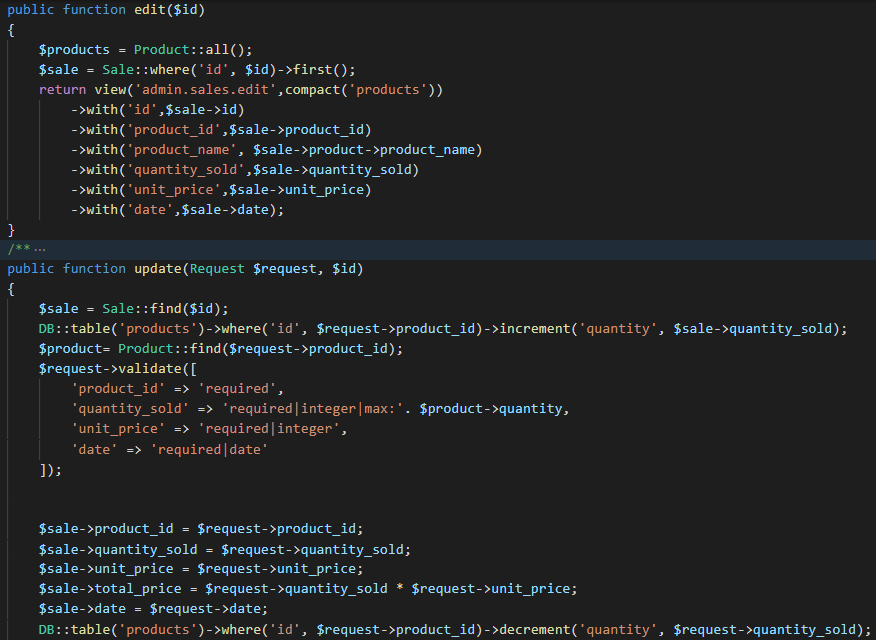
Code screenshot of StockController(2)



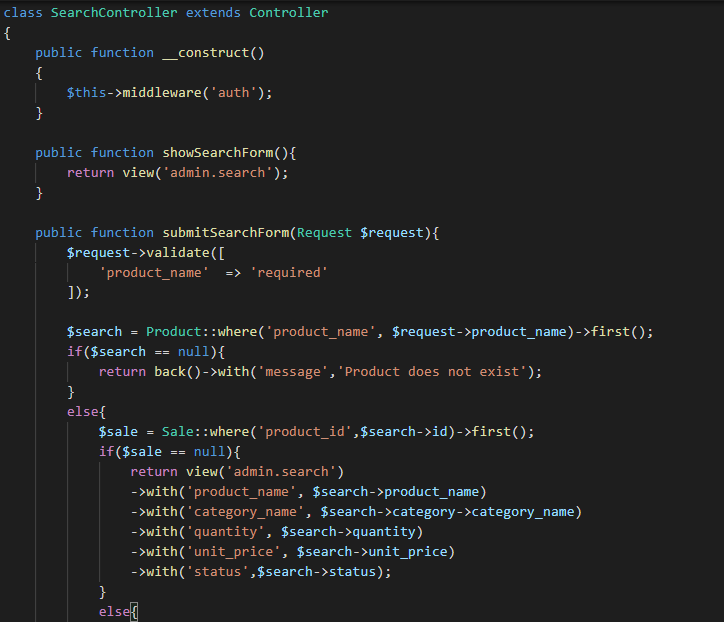
Code screenshot of SalesController(1)



Code screenshot of SalesController(2)



Code screenshot of SearchController



# **SYSTEM TESTING**



## Purpose of System Testing

System testing is a phase of software development which is conducted to check whether the system is working accurately and efficiently as per the requirements before deployment, to check for any missing requirements and to check for any errors and bugs in the system. Various test activities are performed in this phase. Therefore, the purpose of system testing is to make the system free of bugs and errors, to ensure the system works accurately and for the validation and verification of the system.

## Objectives of System Testing

The objectives of system testing are:

* One of the main objectives is to find any bugs and errors in the system.
* To check whether the system works accurately and efficiently.
* A successful test case will have high chance of finding an undiscovered error.

## Importance of System Testing

The importance of system testing are:

* Any kind of bugs and errors in the system can be identified early and solved before its deployment.
* Properly tested system ensures accuracy, efficiency, reliability, security and high performance.
* This results in time saving, cost effectiveness and customer satisfaction. (What is Software Testing? Definition, Basics & Types in Software Engineering , 2021)

## Overall Test Cases

|  |  |
| --- | --- |
| **Test Cases** | **Objective** |
| 1 | To check whether admin is logged in when correct username and password are provided. |
| 2 | To check whether message shows up if Admin tries to login with empty fields in username and password or with other wrong Authentication process. |
| 3 | To check whether validation messages shows up while entering wrong username and password. |
| 4 | To check whether error message shows up if new password does not match with confirm new password while changing password. |
| 5 | To check whether error message displays when minimum password of at least 8 characters is not set in the new password field change password form page. |
| 6 | To check whether password changes or not after clicking Update button. |
| 7 | To check whether the photos of the products are displayed if photos are inserted in admin panel. |
| 8 | To check if a new stock of the product added in stock page increases the quantity of the product in products page. |
| 9 | To check if a new sale of the product added in sales page decreases the quantity of the product in products page. |
| 10 | To test whether the quantity of the product in products page changes (either increases or decreases ) according to the update made in the product stock quantity in stock after clicking edit icon. |
| 11 | To test whether the quantity of the product in products page changes (either increases or decreases ) according to the update made in the sold product quantity in sales after clicking edit icon. |
| 12 | To test whether result displays in search inventory page after entering product name in admin panel. |
| 13 | To check whether delete confirmation modal pops up after clicking delete icon. |
| 14 | To test whether admin logouts after clicking logout button. |
| 15 | To test whether back button of the browser redirects to login page after logout. (for preventing access to admin panel after logout) |
| 16 | To test whether all the details along with availability status of specific product displays after entering product name and search by ‘available option’ in front homepage. |
| 17 | To test whether all the sales details of specific product displays after searching by entering product name and search by ‘sales option’ in front homepage. |

|  |  |
| --- | --- |
| Test Case | 1 |
| Objective | To check whether admin in logged in when correct username and password are provided. |
| Test Data | Username and password |
| Expected Test Result | Admin should be logged in. |
| Actual Test Result | Admin was logged in. |
| Conclusion | Successful |

**Test Result**

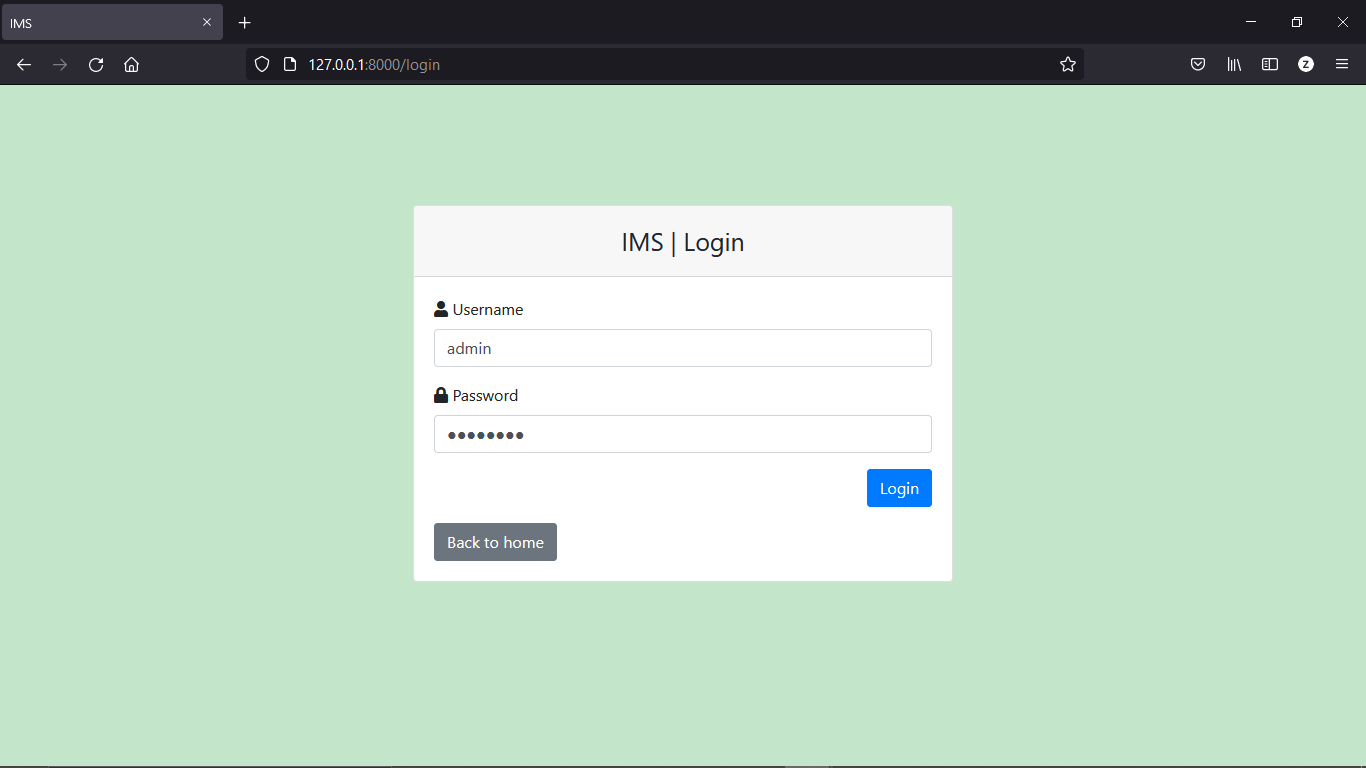


Figure 5: Login form page with correct username and password

After submitting correct username and password, admin was logged in. It redirected to admin dashboard page.

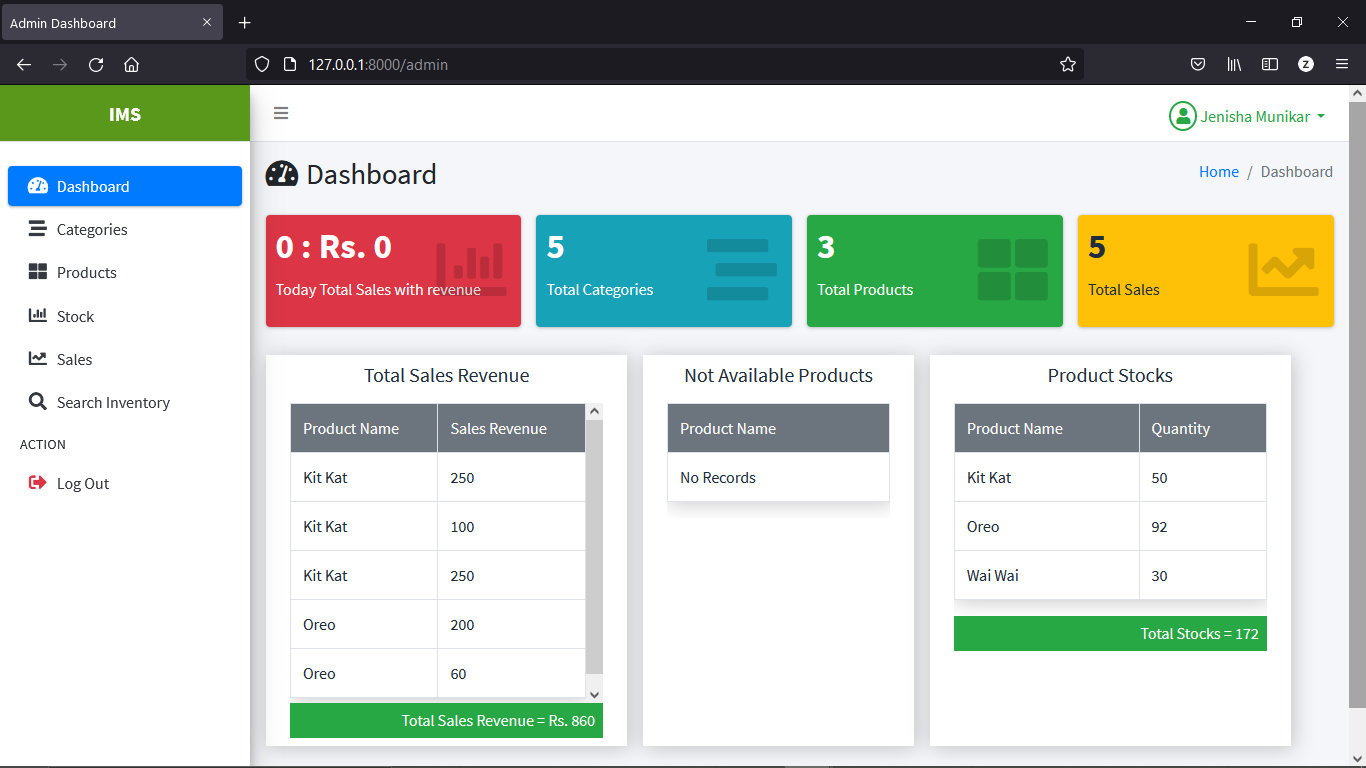


Figure 6: Test result for logging with correct admin username and password

|  |  |
| --- | --- |
| Test Case | 2 |
| Objective | To check whether message shows up if Admin tries to login with empty fields in username and password or with other wrong Authentication process. |
| Test Data | Username and password |
| Expected Test Result | Message should appear |
| Actual Test Result | Message appears |
| Conclusion | Successful |

**Test Result**

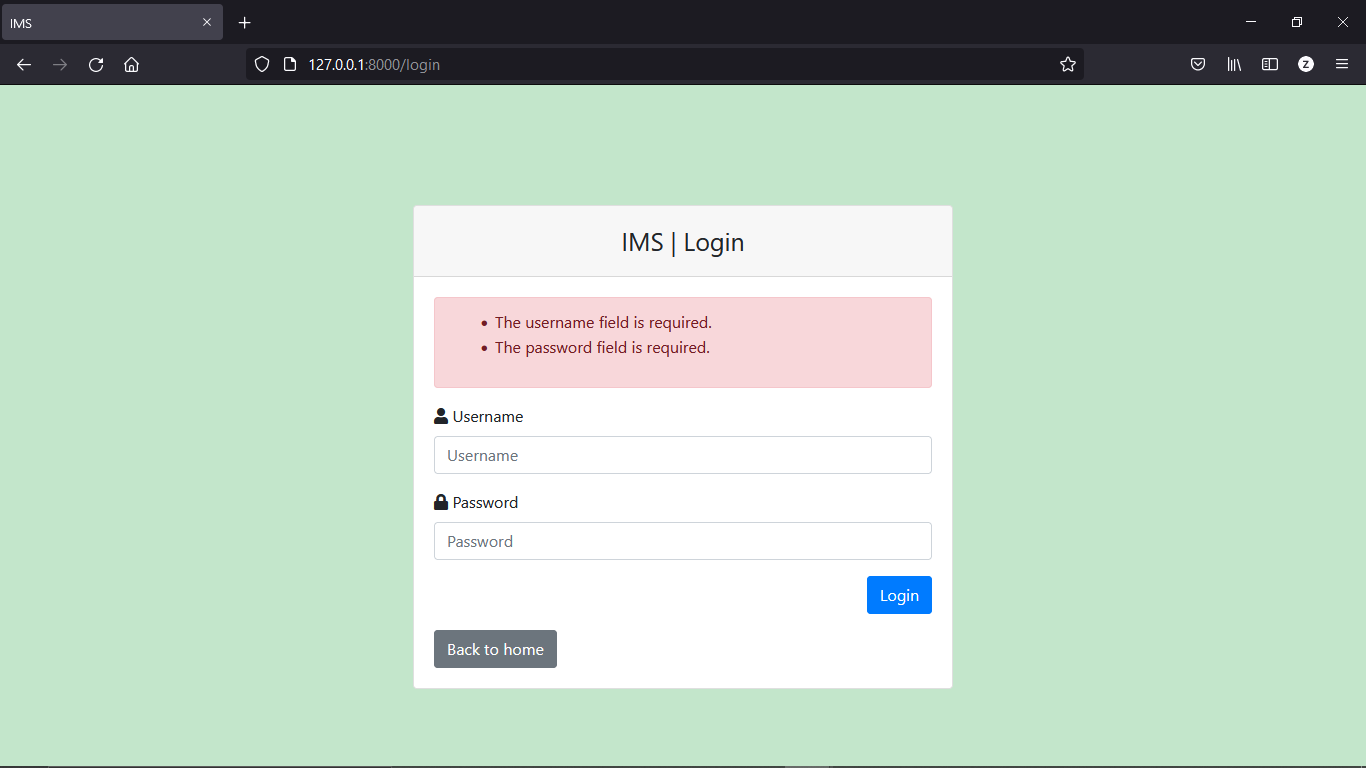


Figure 7: Test result for leaving the fields empty while log in

|  |  |
| --- | --- |
| Test Case | 3 |
| Objective | To check whether validation messages shows up while entering wrong username and password. |
| Test Data | Username and password |
| Expected Test Result | Message should appear |
| Actual Test Result | Message appears |
| Conclusion | Successful |

**Test Result**

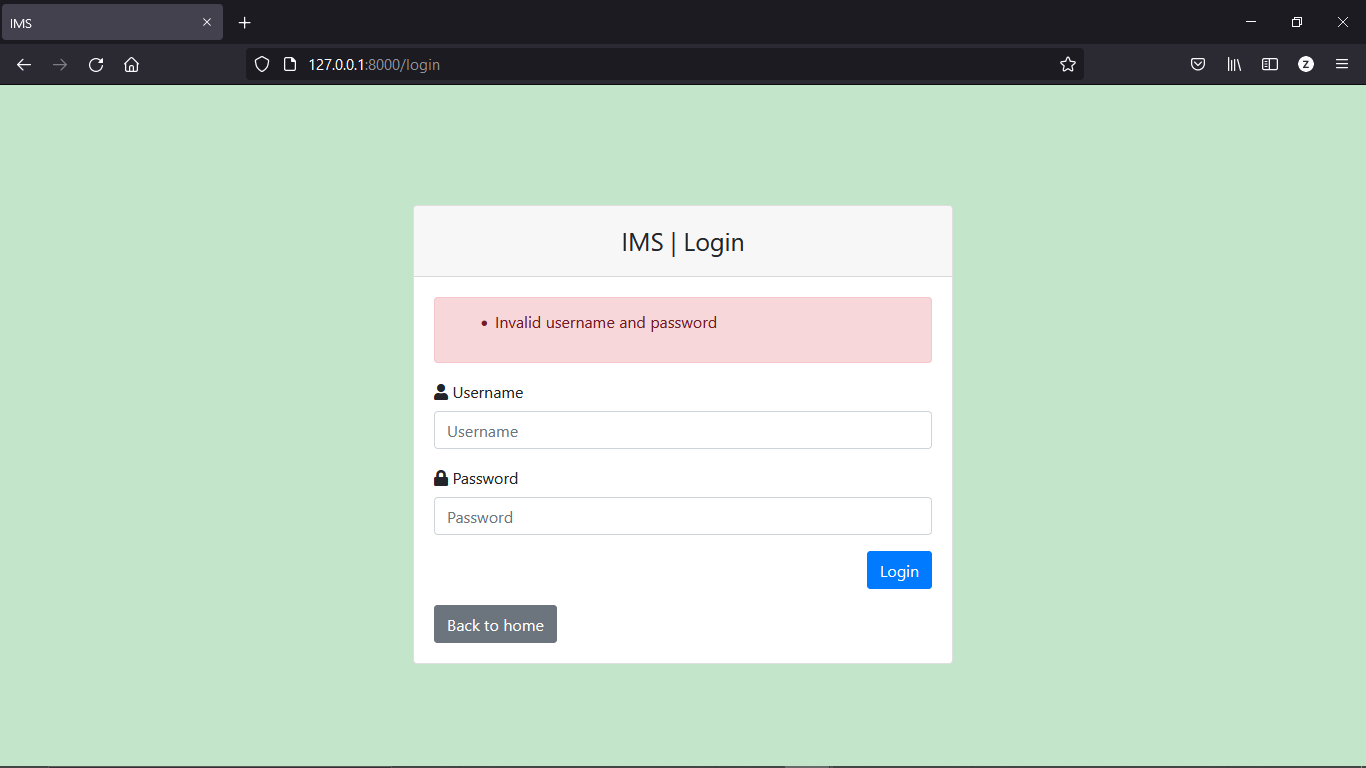


Figure 8: Test result for entering wrong email and password

|  |  |
| --- | --- |
| Test Case | 4 |
| Objective | To check whether error message shows up if new password does not match with confirm new password while changing password. |
| Test Data | Checks at database. |
| Expected Test Result | Message should appear. |
| Actual Test Result | Message appear. |
| Conclusion | Successful |

**Test Result**

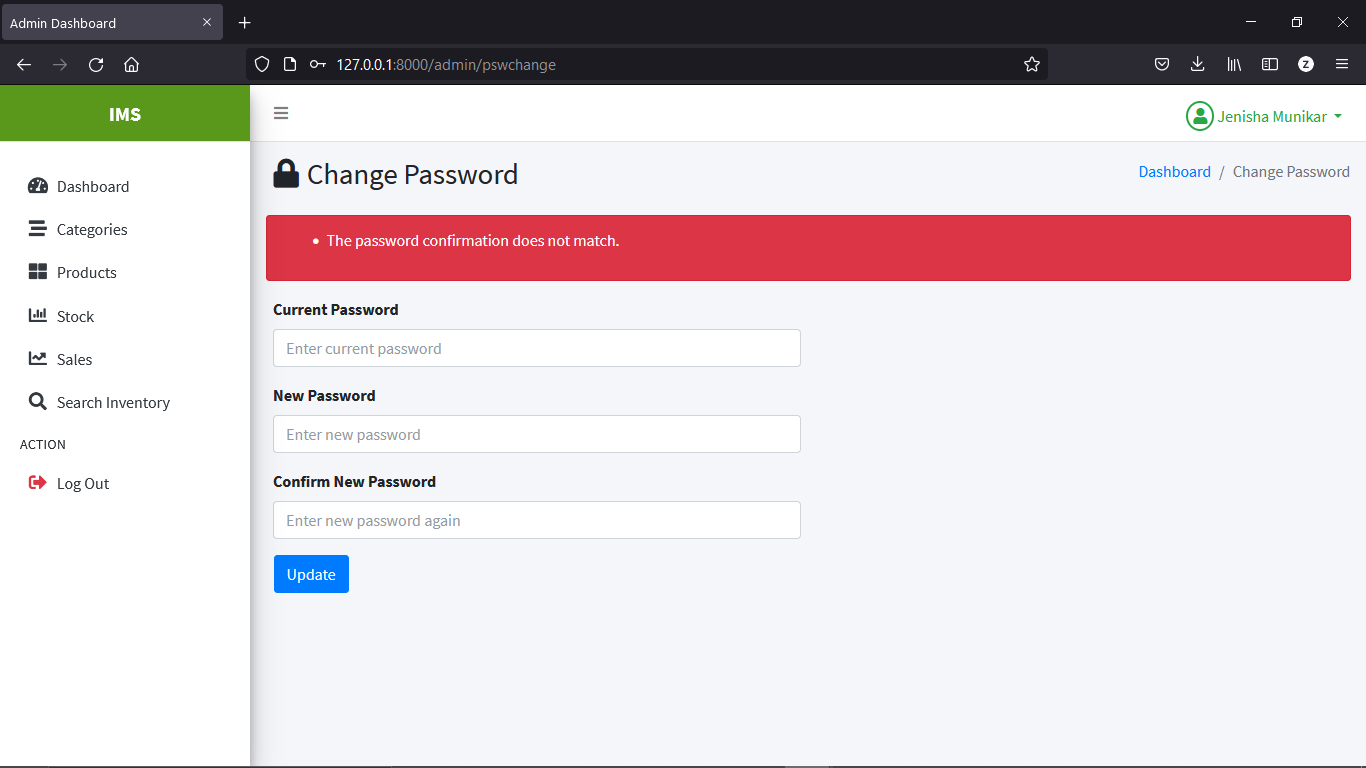


Figure 9: Test result for checking whether error message displays when new password and confirm new password does not match

|  |  |
| --- | --- |
| Test Case | 5 |
| Objective | To check whether error message displays when minimum password of at least 8 characters is not set in the new password field at change password form page. |
| Test Data | New password field data |
| Expected Test Result | The minimum password of at least 8 characters should be set. |
| Actual Test Result | The minimum password of at least 8 characters was set. |
| Conclusion | Successful |

**Test Result**

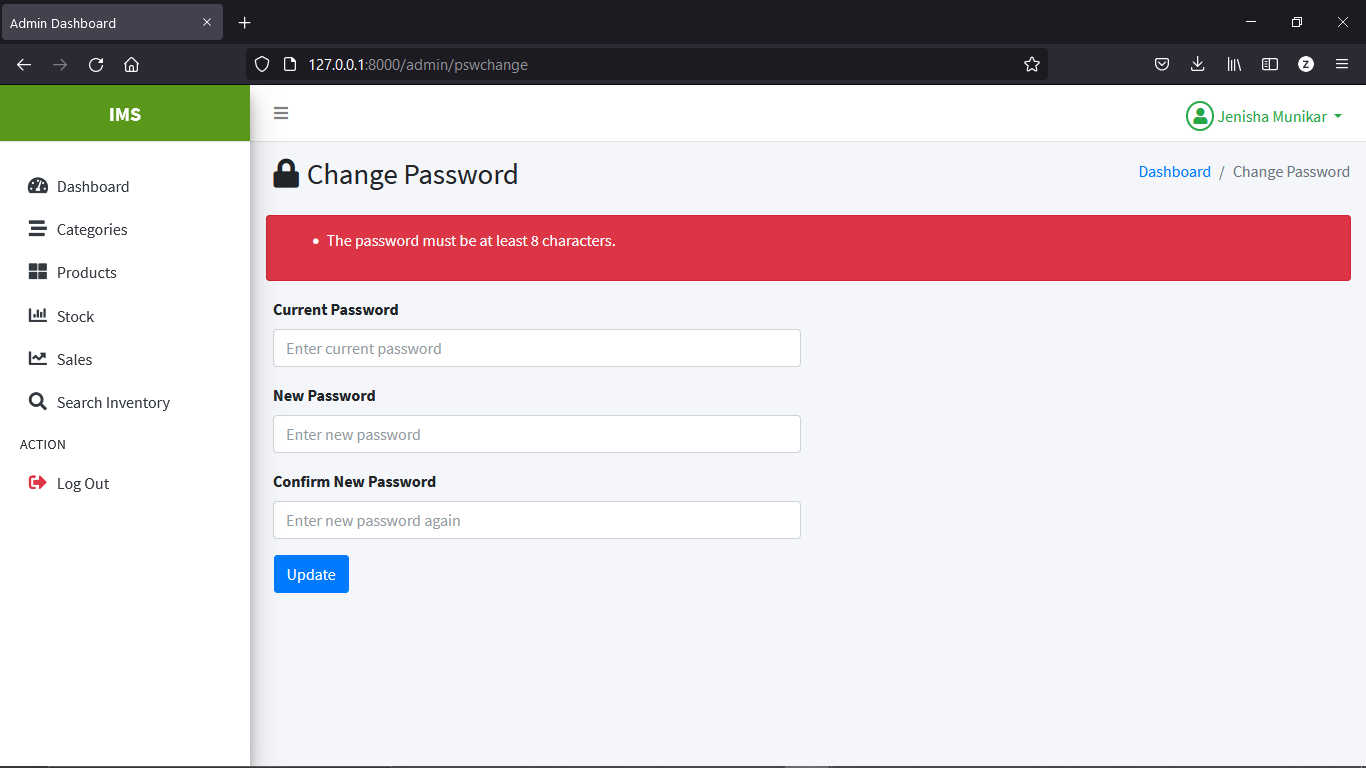


Figure 10: Test result for checking whether error message displays when minimum password of at least 8 characters is not set in new password field

|  |  |
| --- | --- |
| Test Case | 6 |
| Objective | To check whether password changes or not after clicking Update button. |
| Test Data | Checks admin password |
| Expected Test Result | The password should be changed. |
| Actual Test Result | Password changed. |
| Conclusion | Successful |

**Test Result**

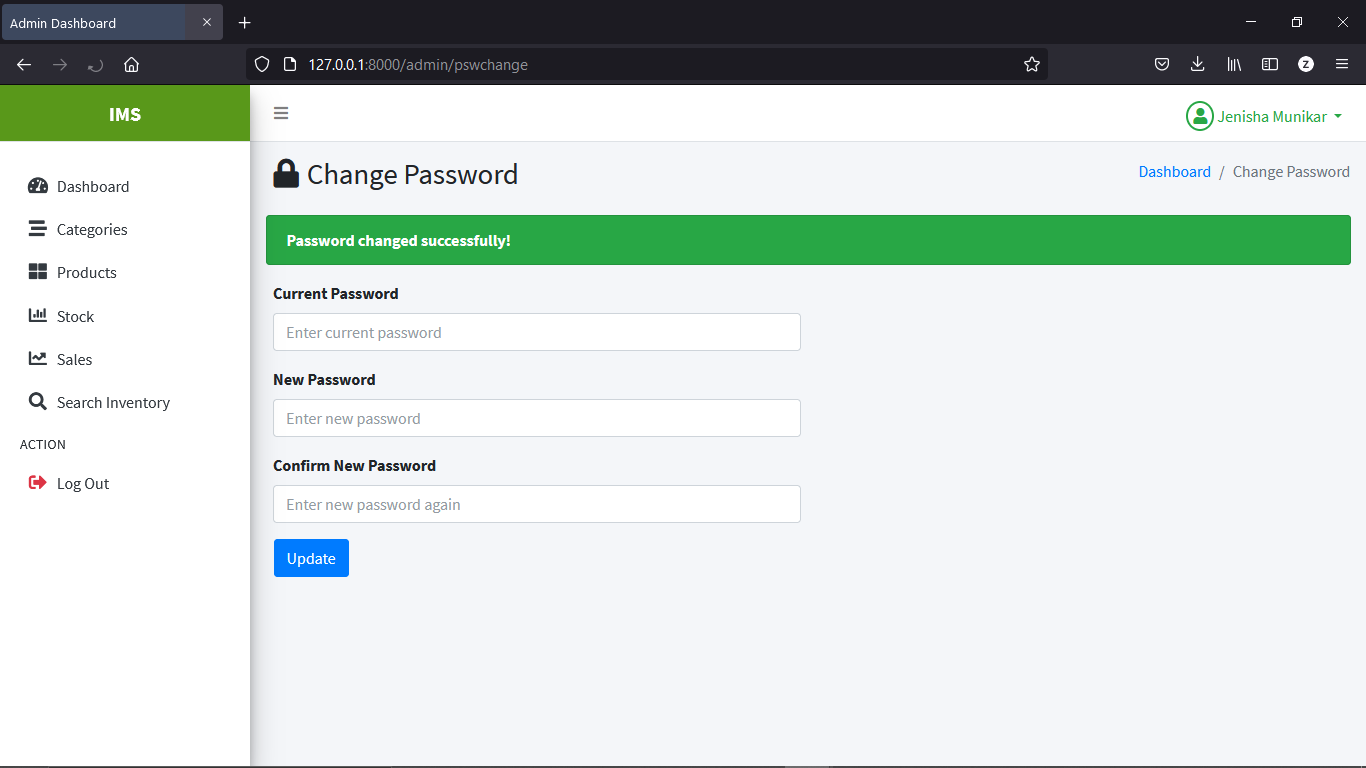


Figure 11: Test result for checking whether password changes or not after clicking Update button.

|  |  |
| --- | --- |
| Test Case | 7 |
| Objective | To check whether the photos of the products are displayed if photos are inserted in admin panel. |
| Test Data | Product photos |
| Expected Test Result | The photo of the products should be displayed. |
| Actual Test Result | The photo of the products are displayed |
| Conclusion | Successful |

**Test Result**

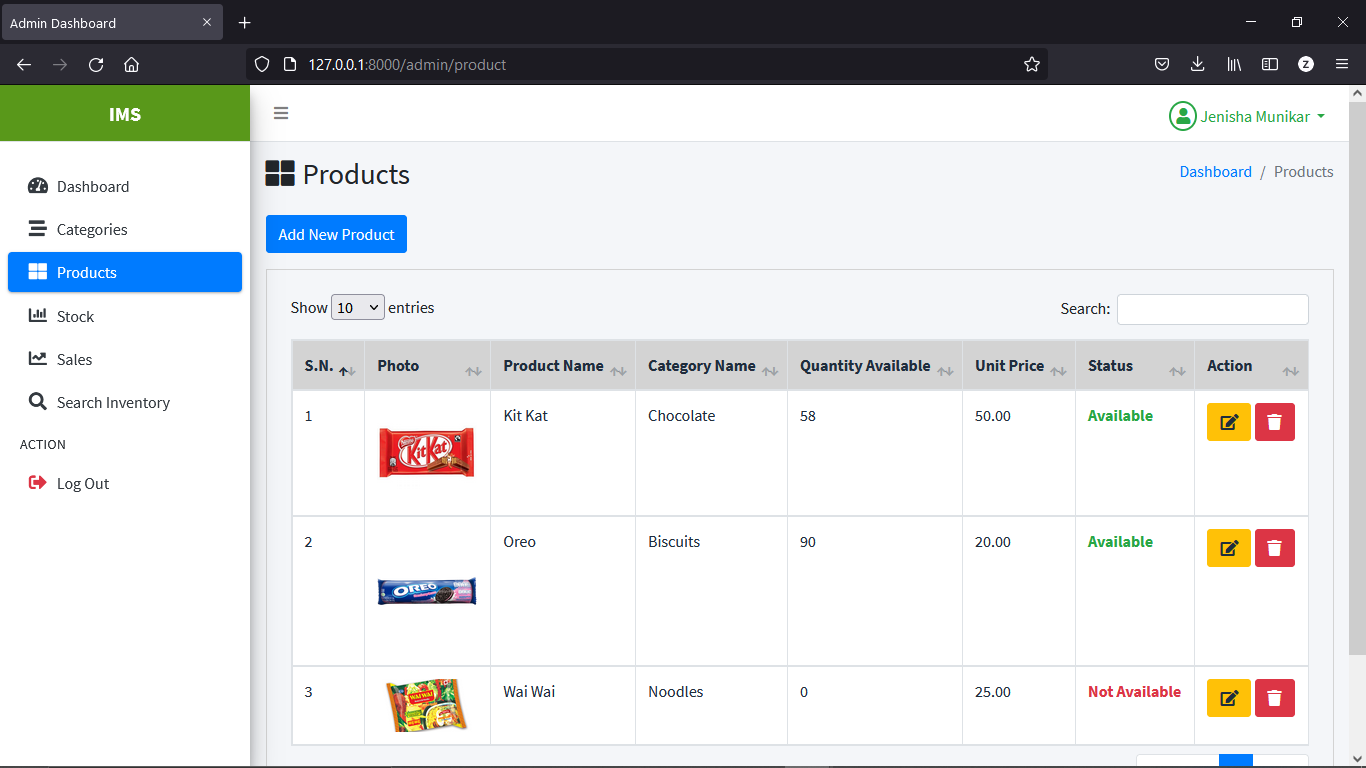
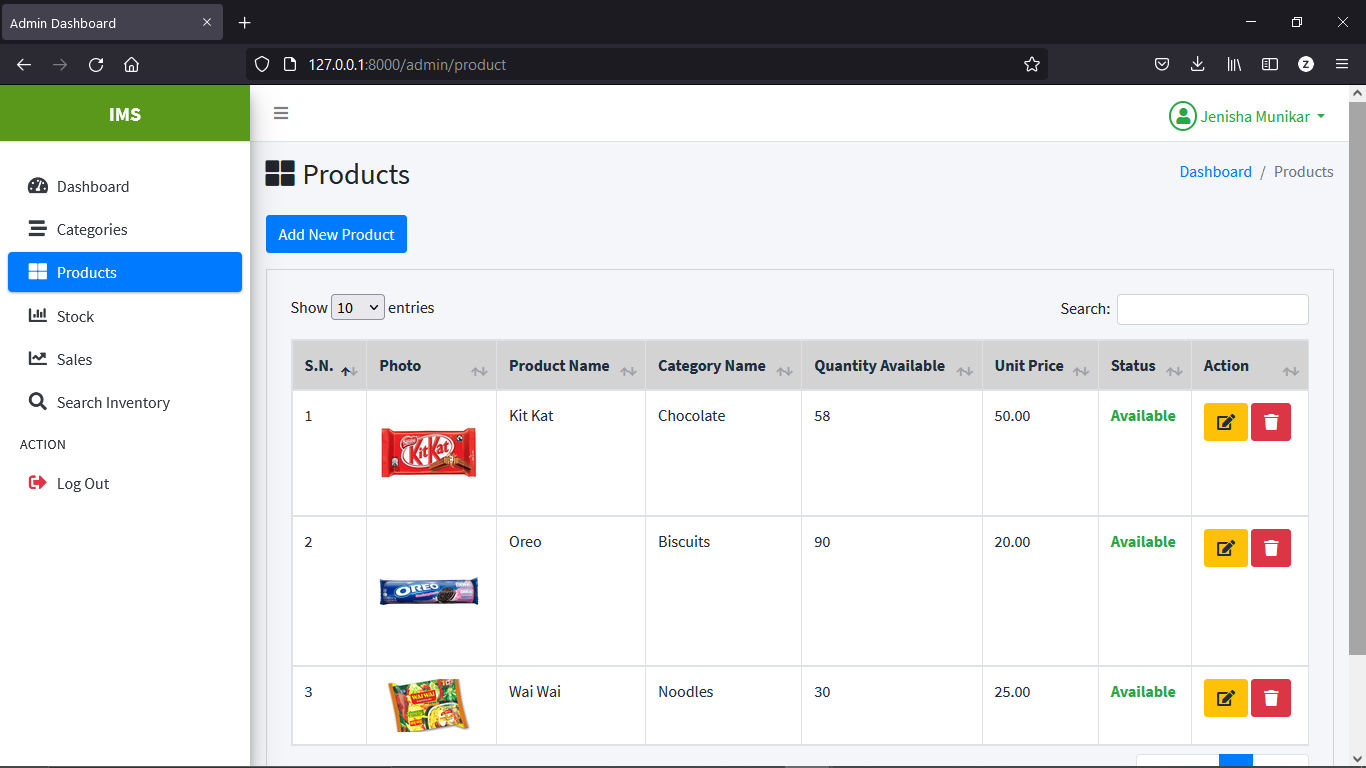


Figure 12: Test result for checking whether the photos of the products are displayed if photos are inserted

|  |  |
| --- | --- |
| Test Case | 8 |
| Objective | To check if a new stock of the product added in stock page increases the quantity of the product in products page. |
| Test Data | Quantity of the product in products page |
| Expected Test Result | The quantity of the product should be increased in products page. |
| Actual Test Result | The quantity of the product is increased. |
| Conclusion | Successful |

**Test Result**

Before adding new stock, the quantity of the product ‘Oreo ’ was 90.



After adding new stock with stock quantity 10, the quantity of the product ‘Oreo’ is increased to 100.

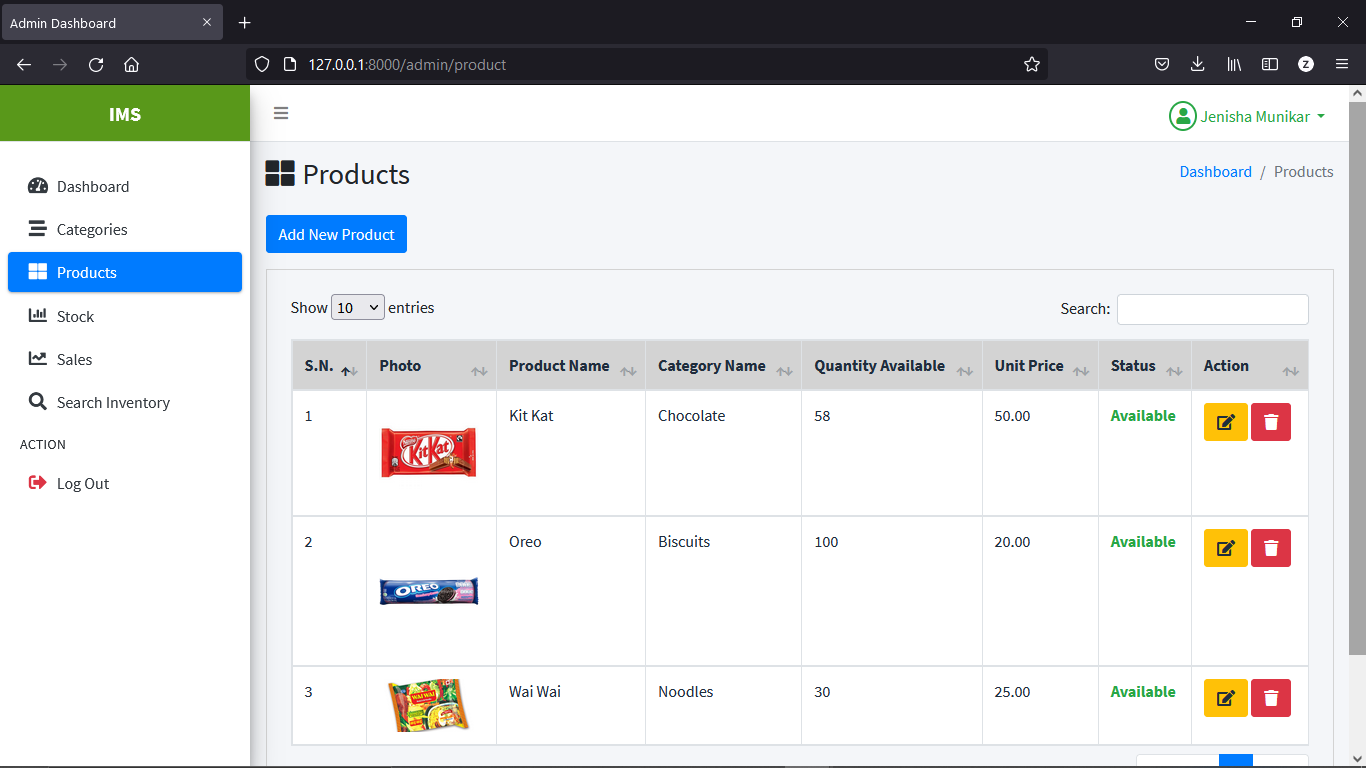


Figure 13: Test result for checking if a new stock of the product added in stock page increases the quantity of the product in products page.

|  |  |
| --- | --- |
| Test Case | 9 |
| Objective | To test whether the quantity of the product in products page changes (either increases or decreases ) according to the update made in the product stock quantity in stock. |
| Test Data | Quantity of the product in products page |
| Expected Test Result | The quantity of the product should be changed in products page. |
| Actual Test Result | The quantity of the product is changed. |
| Conclusion | Successful |

**Test Result**

After editing the stock quantity of product ‘Oreo’ from 10 to 5, the quantity of the product is decreased from 100 to 95.

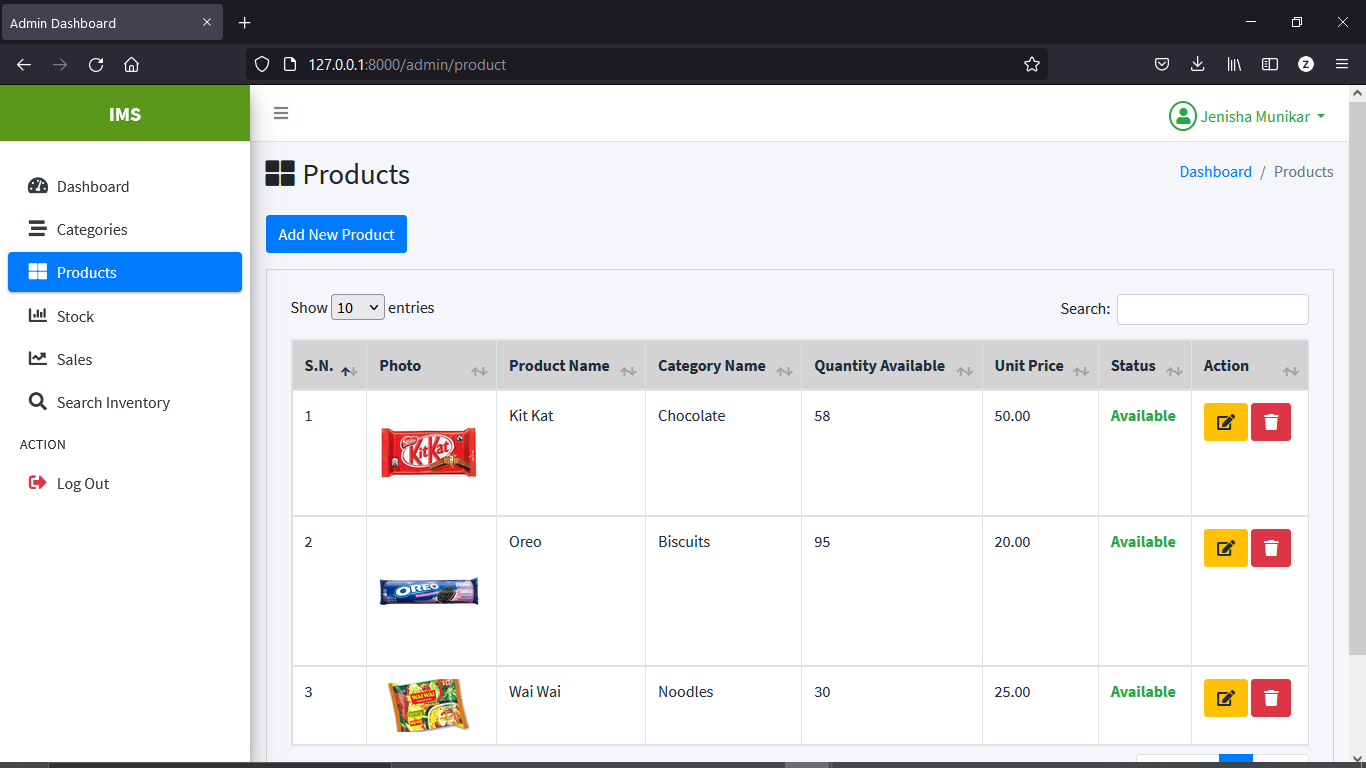
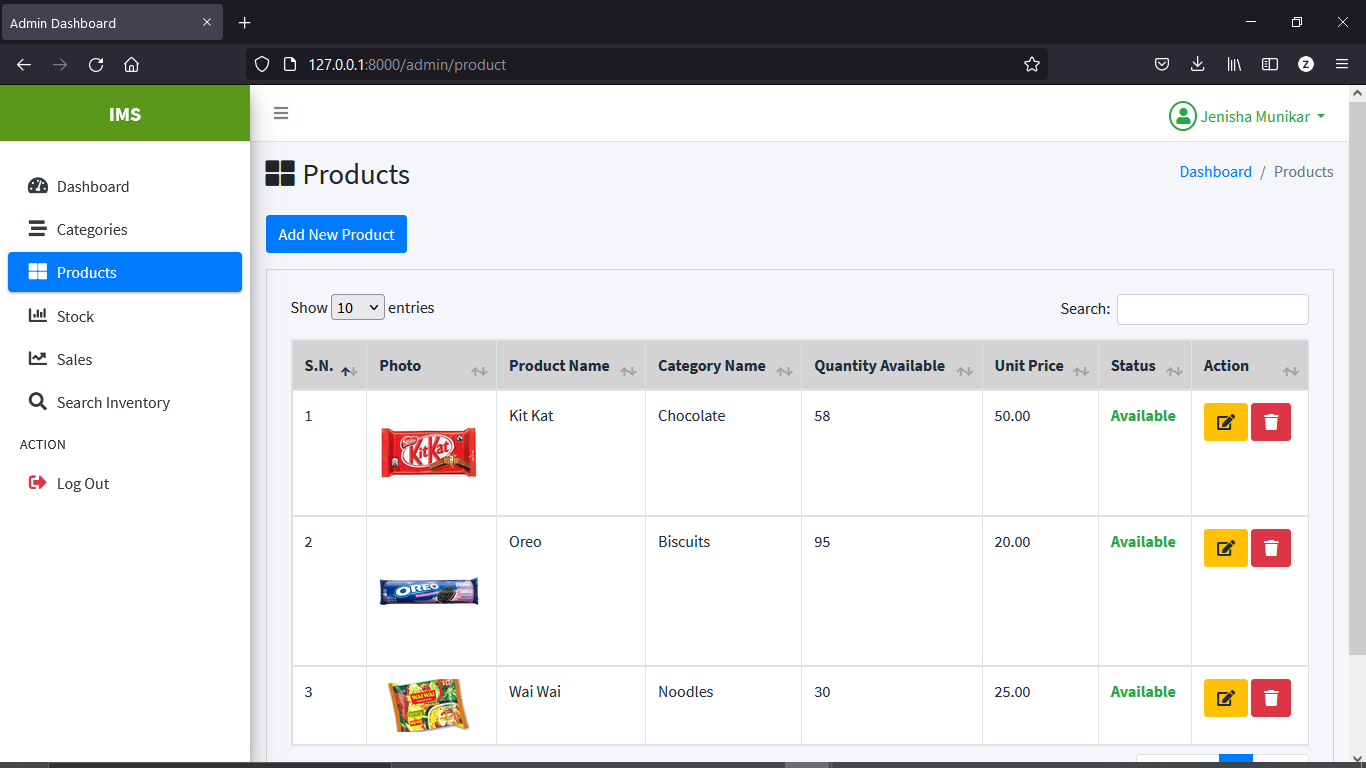


Figure 14: Test result for checking whether the quantity of the product in products page changes according to the update made in the product stock quantity in stock

|  |  |
| --- | --- |
| Test Case | 10 |
| Objective | To check if a new sales of the product added in sales page decreases the quantity of the product in products page. |
| Test Data | Quantity of the product in products page |
| Expected Test Result | The quantity of the product should be decreased in products page. |
| Actual Test Result | The quantity of the product is decreased. |
| Conclusion | Successful |

**Test Result**

Before adding new sales of product ‘Oreo’, the quantity of the product ‘Oreo ’ was 95.



After adding new sales with sold quantity 5, the quantity of the product ‘Oreo’ is decreased to 90.

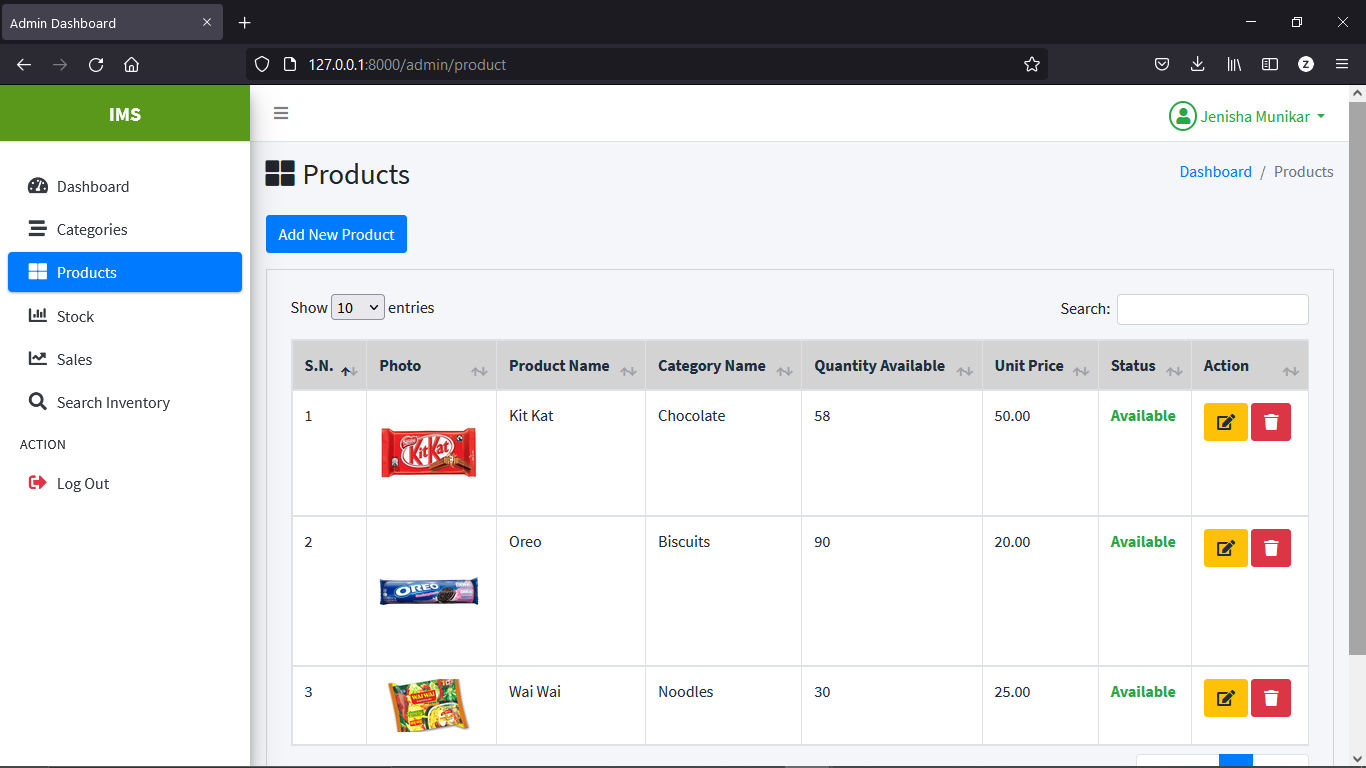


Figure 15: Test result for checking if a new sales of the product added in sales page decreases the quantity of the product in products page

|  |  |
| --- | --- |
| Test Case | 11 |
| Objective | To test whether the quantity of the product in products page changes (either increases or decreases ) according to the update made in the sold product quantity in sales after clicking edit icon. |
| Test Data | Quantity of the product in products page |
| Expected Test Result | The quantity of the product should be changed in products page. |
| Actual Test Result | The quantity of the product is changed. |
| Conclusion | Successful |

**Test Result**

After editing the sold quantity of product ‘Oreo’ from 5 to 3, the quantity of the product is increased from 90 to 92.

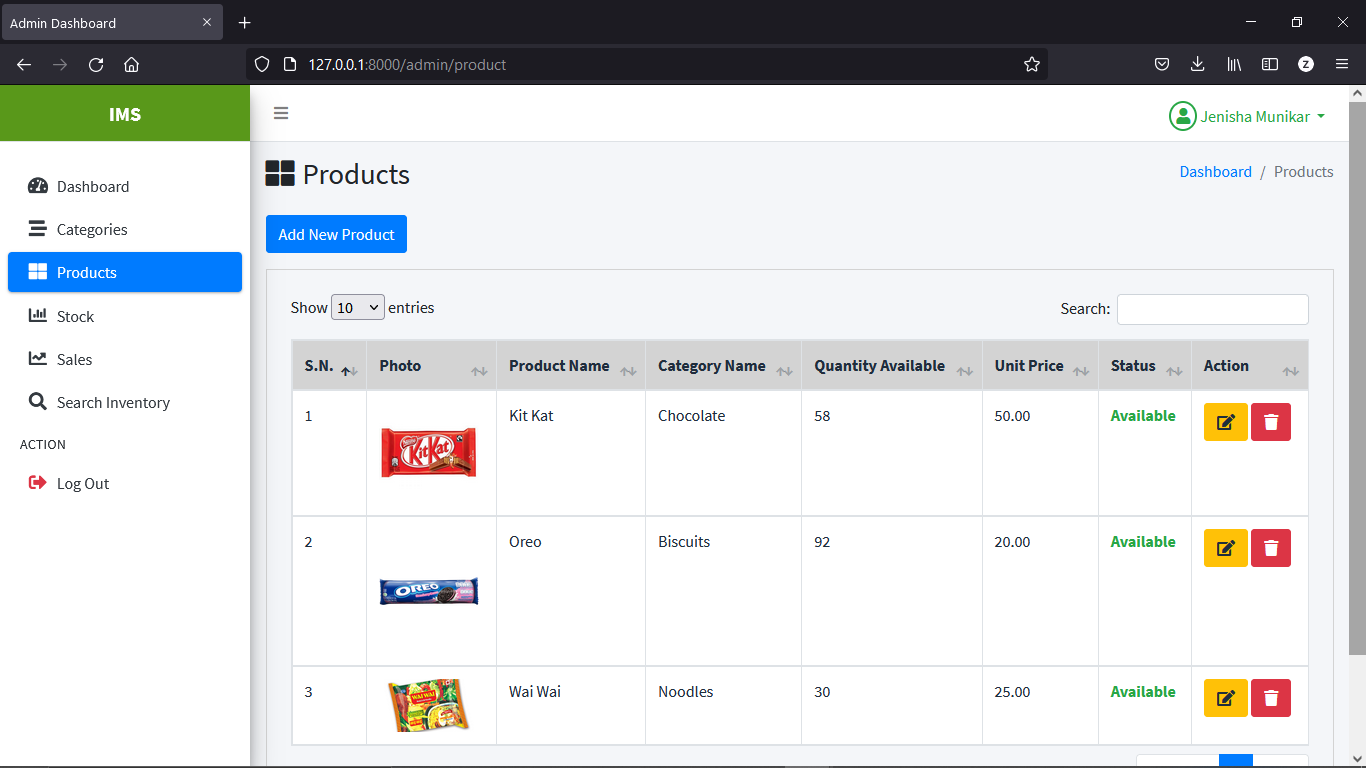


Figure 16: Test result for checking whether the quantity of the product in products page changes according to the update made in the sold product quantity in sales

|  |  |
| --- | --- |
| Test Case | 12 |
| Objective | To test whether result displays in search inventory page of admin panel after entering product name. |
| Test Data | Product name |
| Expected Test Result | The result should display all the details of test data. |
| Actual Test Result | The result is displayed. |
| Conclusion | Successful |

**Test Result**

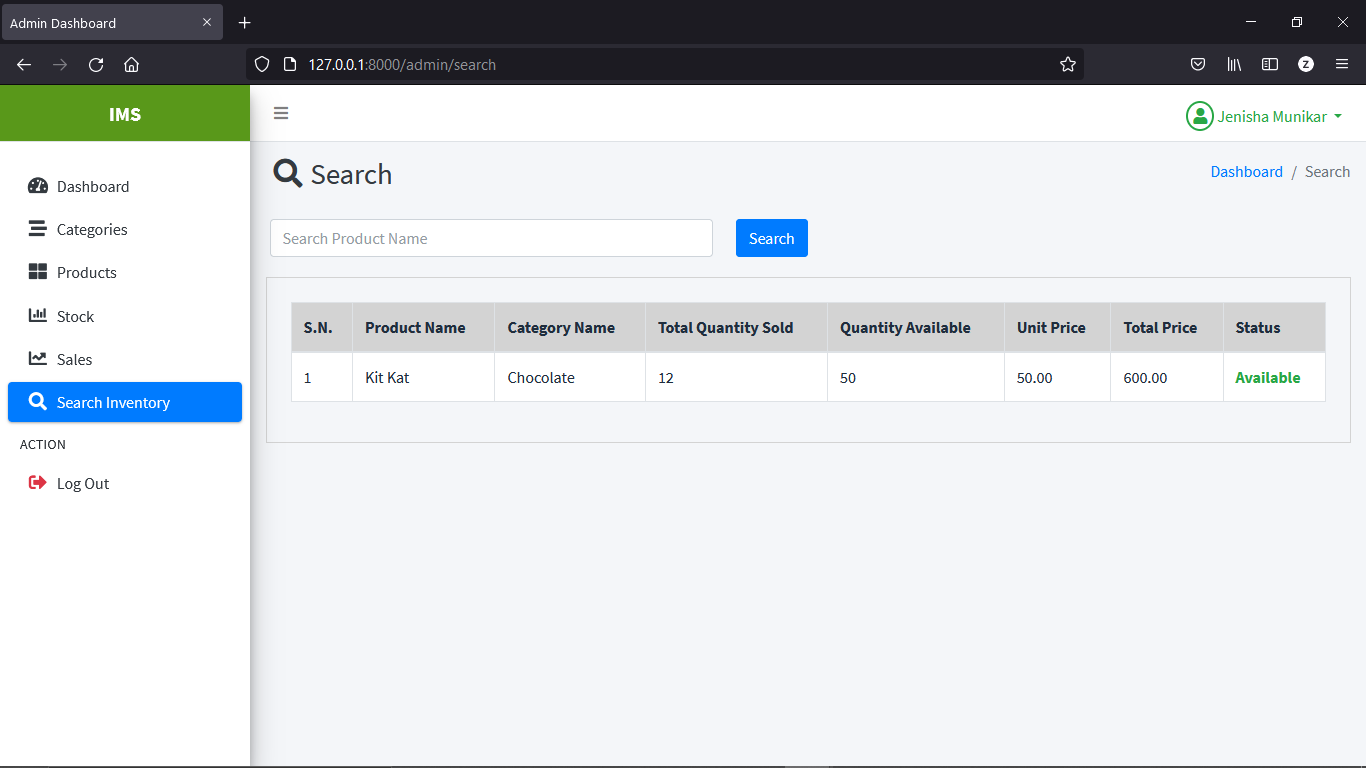


Figure 17: Test result for checking whether result displays in search inventory page of admin panel after entering product name

|  |  |
| --- | --- |
| Test Case | 13 |
| Objective | To check whether delete confirmation modal pops up after clicking delete icon. |
| Test Data | Delete icon |
| Expected Test Result | The delete confirmation modal should pop up. |
| Actual Test Result | The delete confirmation modal pops up. |
| Conclusion | Successful |

**Test Result**

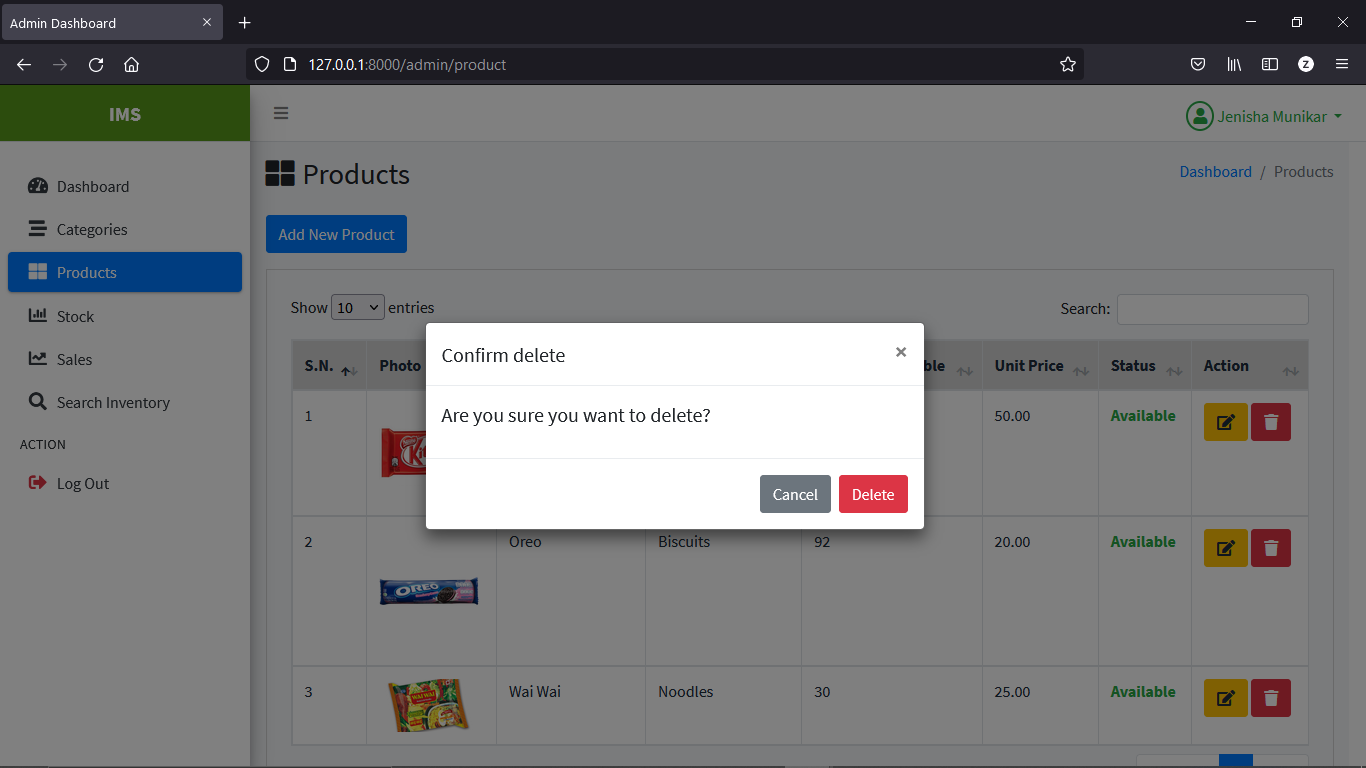


Figure 18: Test result for checking whether delete confirmation modal pops up after clicking delete icon

|  |  |
| --- | --- |
| Test Case | 14 |
| Objective | To test whether admin logouts after clicking logout button. |
| Test Data | Logout button |
| Expected Test Result | Admin should be logged out. |
| Actual Test Result | Admin was logged out. |
| Conclusion | Successful |

**Test Result**

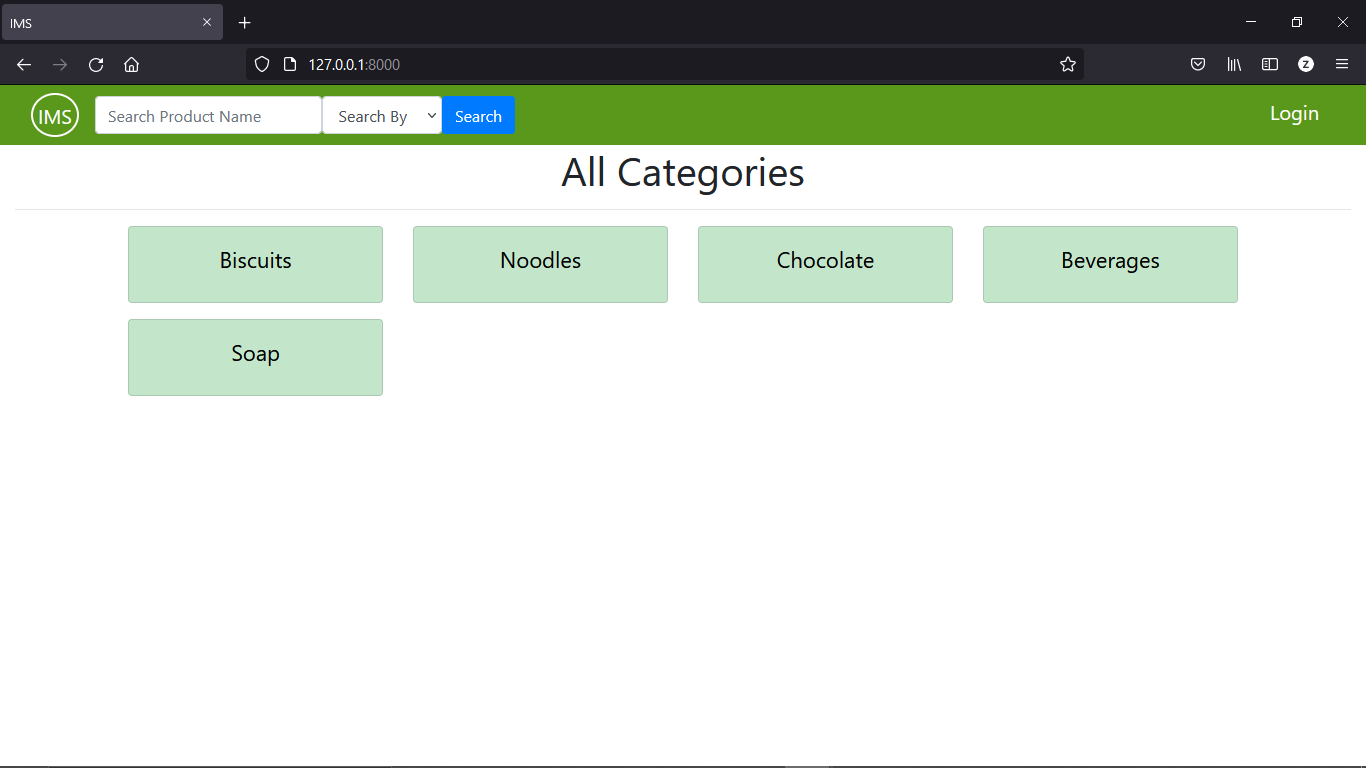


Figure 19: Test result for checking whether admin logouts after clicking logout button\

|  |  |
| --- | --- |
| Test Case | 15 |
| Objective | To test whether back button of the browser redirects to login page after logout. (for preventing access to admin panel after logout) |
| Test Data | Back button of browser |
| Expected Test Result | Back button of browser should redirect to login page after logout. |
| Actual Test Result | Back button of browser redirects to login page. |
| Conclusion | Successful |

**Test Result**

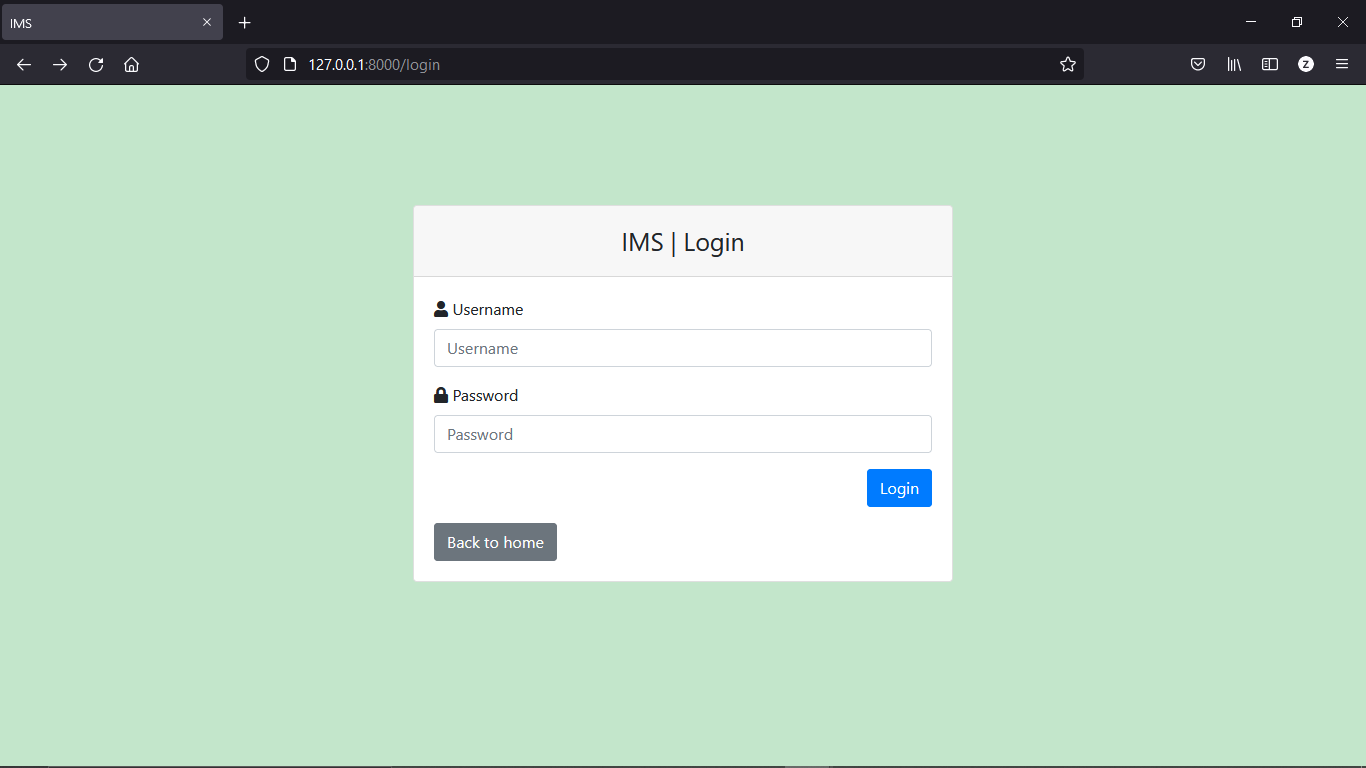


Figure 20: Test result for checking whether back button of the browser redirects to login page after logout. (for preventing access to admin panel after logout)

|  |  |
| --- | --- |
| Test Case | 16 |
| Objective | To test whether all the details along with availability status of specific product displays after searching by entering product name and search by ‘available option’ in front homepage. |
| Test Data | Product name and available option |
| Expected Test Result | All the details along with availability status of the searched product should be displayed. |
| Actual Test Result | All the details along with availability status of the searched product is displayed. |
| Conclusion | Successful |

**Test Result**

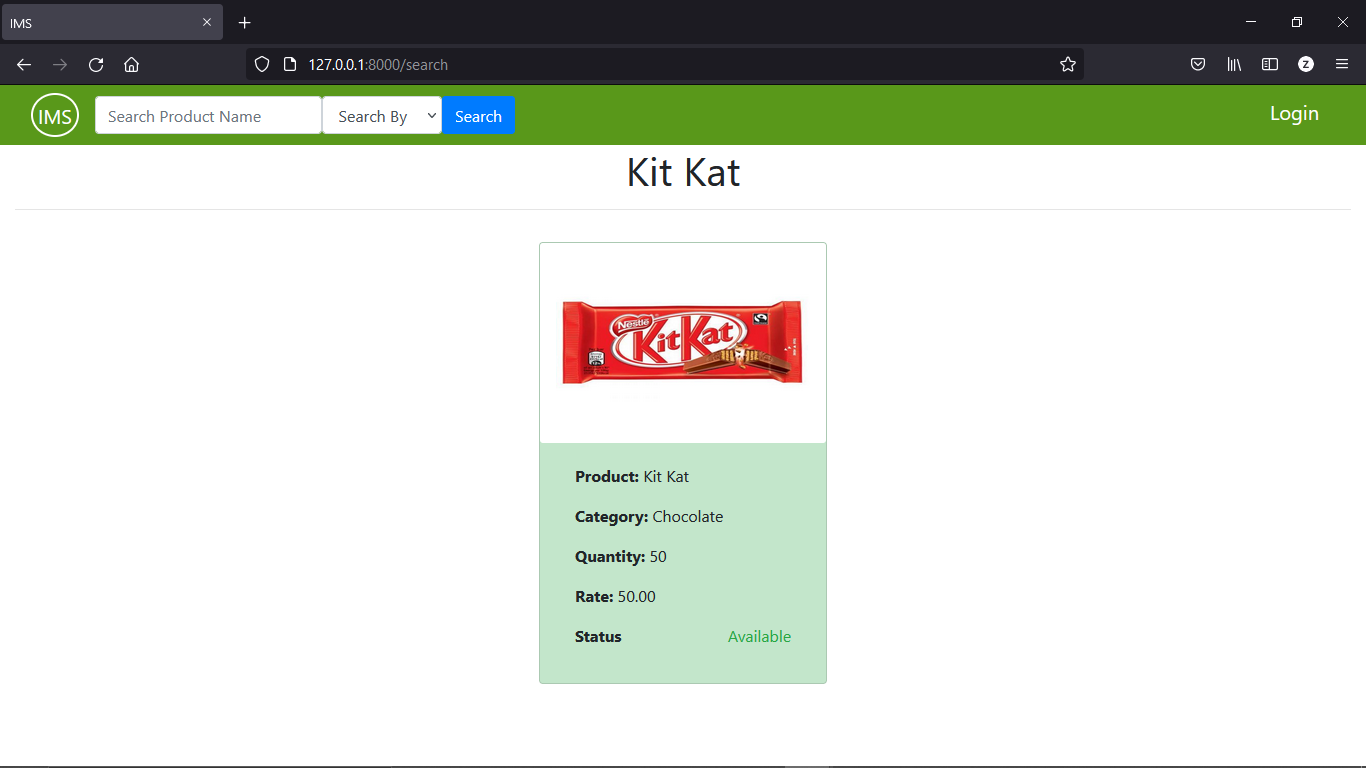


Figure 21: Test result for checking whether all the details along with availability status of specific product displays after searching by entering product name and search by ‘available option’ in front homepage.

|  |  |
| --- | --- |
| Test Case | 17 |
| Objective | To test whether all the sales details of specific product displays after searching by entering product name and search by ‘sales option’ in front homepage. |
| Test Data | Product name and sales option |
| Expected Test Result | All the sales details of the searched product should be displayed. |
| Actual Test Result | All the sales details of the searched product is displayed. |
| Conclusion | Successful |

**Test Result**

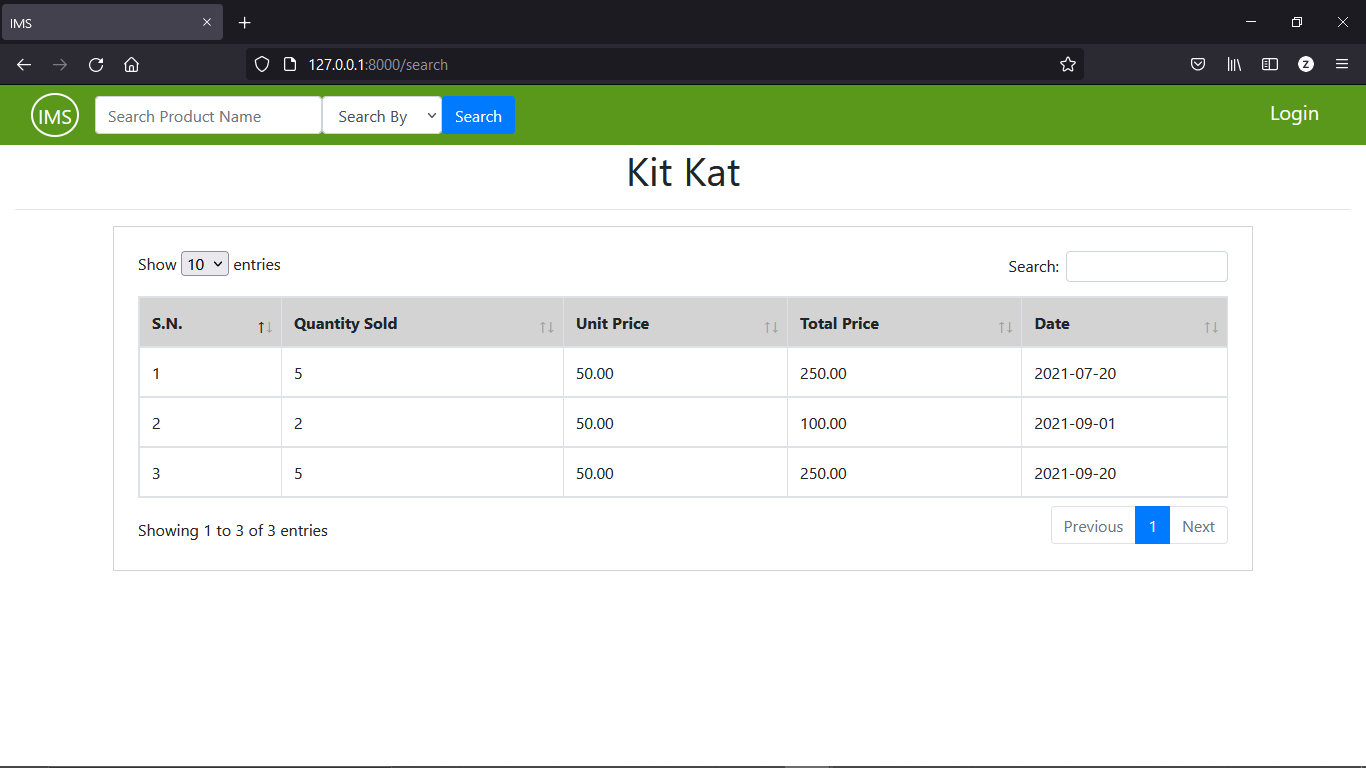


Figure 22: Test result for checking whether all the sales details of specific product displays after searching by entering product name and search by ‘sales option’ in front homepage.

# **System Evaluation**

## Introduction

## Evaluation Overview

## Evaluation of Project Deliverables

## System Evaluation

## Criteria

## Evaluation of Project Practices

## Conclusion

# **Project Conclusion**

## Introduction

## Project Achievement

## Future Escalation

## Critical Appraisal

## Personal Achievement

## Summary

# **Conclusion**

# **References**

ApacheFriends. (2021). *XAMPP Apache + MariaDB + PHP + Perl*. Retrieved from ApacheFriends: https://www.apachefriends.org/index.html

Laravel. (2021). *Installation*. Retrieved from Laravel: https://laravel.com/docs/8.x#meet-laravel

*Prototyping Model in Software Engineering: Methodology, Process, Approach* . (2021). Retrieved from Guru99: https://www.guru99.com/software-engineering-prototyping-model.html#1

*SDLC - Waterfall Model*. (n.d.). Retrieved from TutorialsPoint: https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm

SmartDraw Software, LLC. (n.d.). *Entity Relationship Diagram*. Retrieved from SmartDraw: https://www.smartdraw.com/entity-relationship-diagram/#whatIsERD

SmartDraw Software, LLC. (n.d.). *Use Case Diagram*. Retrieved from SmartDraw: https://www.smartdraw.com/use-case-diagram/

*Spiral Model: When to Use? Advantages & Disadvantages* . (2021). Retrieved from Guru99: https://www.guru99.com/what-is-spiral-model-when-to-use-advantages-disadvantages.html

*What is Data Flow Diagram (DFD)? How to Draw DFD?* (2012, January 27). Retrieved from Visual Paradigm: https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp

*What is Software Testing? Definition, Basics & Types in Software Engineering* . (2021). Retrieved from Guru99: https://www.guru99.com/software-testing-introduction-importance.html#2