

INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

1. INTRODUCTION

Project Overview

Purpose

2. LITERATURE SURVEY

Existing problem

References

Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas

Ideation & Brainstorming

Proposed Solution

Problem Solution fit

4. REQUIREMENT ANALYSIS

Functional Requirements

Non-Functional Requirements

5. PROJECT DESIGN

Data Flow Diagrams

Solution & Technical Architecture

User Stories

6. PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation

Sprint Delivery Schedule

7. CODING & SOLUTIONING

Feature 1

Feature 2

Database Schema (if Applicable)

8. TESTING

Test Cases

User Acceptance Testing

9. RESULTS

Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

ABSTRACT

The Purpose of this project is to highlight the versatility and functionality of flask along with MySQL by developing an inventory management system which efficiently handles the addition, updating and deletion of products and an interface for product sales. The in-depth explanation of all the technologies and its integration has been described below.

CHAPTER 1

INTRODUCTION

PROJECT OVERVIEW

Inventory, A structured, categorized and complete collection of items or objects that give clear insights about the qualitative and quantitative attributes of these objects. In this modern age where e-commerce websites are breaking the internet and consumption for most products have exponentially increased, it is of utmost importance to maintain a robust inventory system which is capable of maintaining record like products, quantities and various other transitive data needed to run a business smoothly. Hence, we have developed a robust Inventory management system which runs on flask and is connected to a MySQL database. The front end has been developed using bootstrap and connected to the back-end using flask. The system will be logon enabled. The managers will be able to add, update, delete, retrieve it and display it on the front-end.

PURPOSE

The aim of this project is to showcase the versatility & pragmatics of utilizing Flask along with Remote MySQL to deliver a robust inventory management system. The Scope of the project is to deliver a system that will successfully accept the data from the user to insert, update, delete data from the Database.

CHAPTER 2

LITERATURE SURVEY

EXISTING PROBLEM

Businesses are quickly realizing that inventory control is absolutely necessary to run an efficient business and make money in the process. Especially in today's competitive marketplace, business owners simply can't afford to have money go down the drain.

- High cost of inventory
- Consistent stock outs
- Low rate of inventory control
- High amount of obsolete inventory
- High amount of working capital
- High cost of storage
- Spreadsheet data-entry errors
- Lost customers

CHAPTER 3

PROPOSED SOLUTION

For this system we shall be using the following technologies:

FLASK:

Flask is a lightweight micro framework that is used to build web applications. 'Micro' does not mean that our whole web application must fit into a single Python file, nor does it mean that Flask is lacking in functionality. The 'micro' in micro framework means Flask aims to keep the core simple but extensible.[Flask, 2019] This means that there are no libraries or functionalities that are pre-bound or embedded with the flask core initially. We can choose the functionalities we want to integrate with our flask, that is it won't make many decisions for you, such as what database, security features, etc to use. Although, decisions that it does make are, what templating engine to be used, are easy to change. Everything else is up to you, so that Flask can be everything you need and nothing you don't. Flask, by default, does not include a database abstraction layer, form validation or anything else where different libraries already exist that can handle that. Instead, it supports extensions to add such functionalities to our application as if it were a core part of flask itself. There are various extensions that are provided for database integration, form validation, upload handling, various open authentication technologies, and more. Flask may be "micro", but it's ready for production use on a variety of needs.

IBM DB2:

Overview. This feature allows hosts (servers) to access **IBM DB2®** databases on your account. This is useful, for example, if you wish to allow shopping cart or guestbook applications on other servers to access your databases.

CHAPTER 4

REQUIREMENT ANALYSIS

HARDWARE/SOFTWARE DESIGNING

Hardware Requirements:

- Minimum 1GB Ram
- Minimum Intel i3 Core Processor
- Minimum 10GB Hard disk space

Software Requirements:

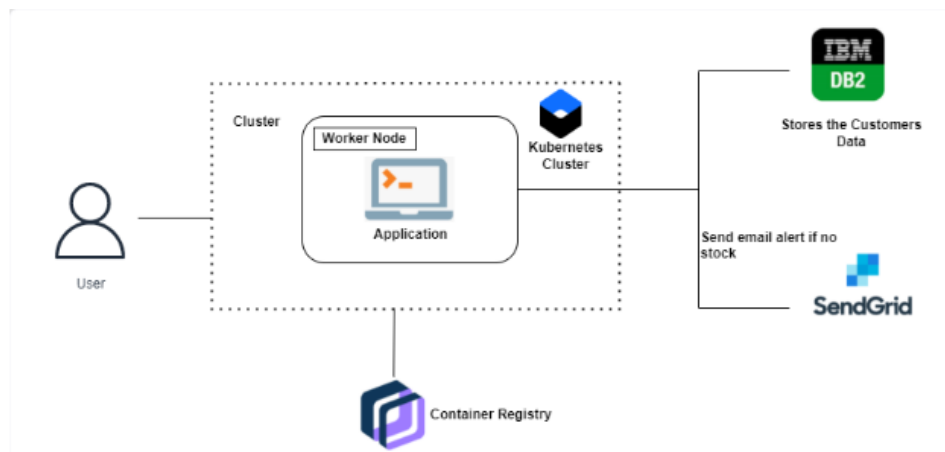
- Operating System: Windows 7 or later
- Browser: Chrome latest version
- Internet Connection: 1Mbps

CHAPTER 5

PROJECT DESIGN

THEORETICAL ANALYSIS

BLOCK DIAGRAM



EXPERIMENTAL INVESTIGATION

Modules included in web application as demanded by the proposed solution and adhering to the above block diagram are:

Registration: The user can register with their details in order to login in the web application.

Login: The users can Login in the web application using their details.

Add Product: Users/Retailers can add their products by entering Product name, Desc, and QTY.

Report: Users can Export the list of products and QTY by clicking the Export report button.

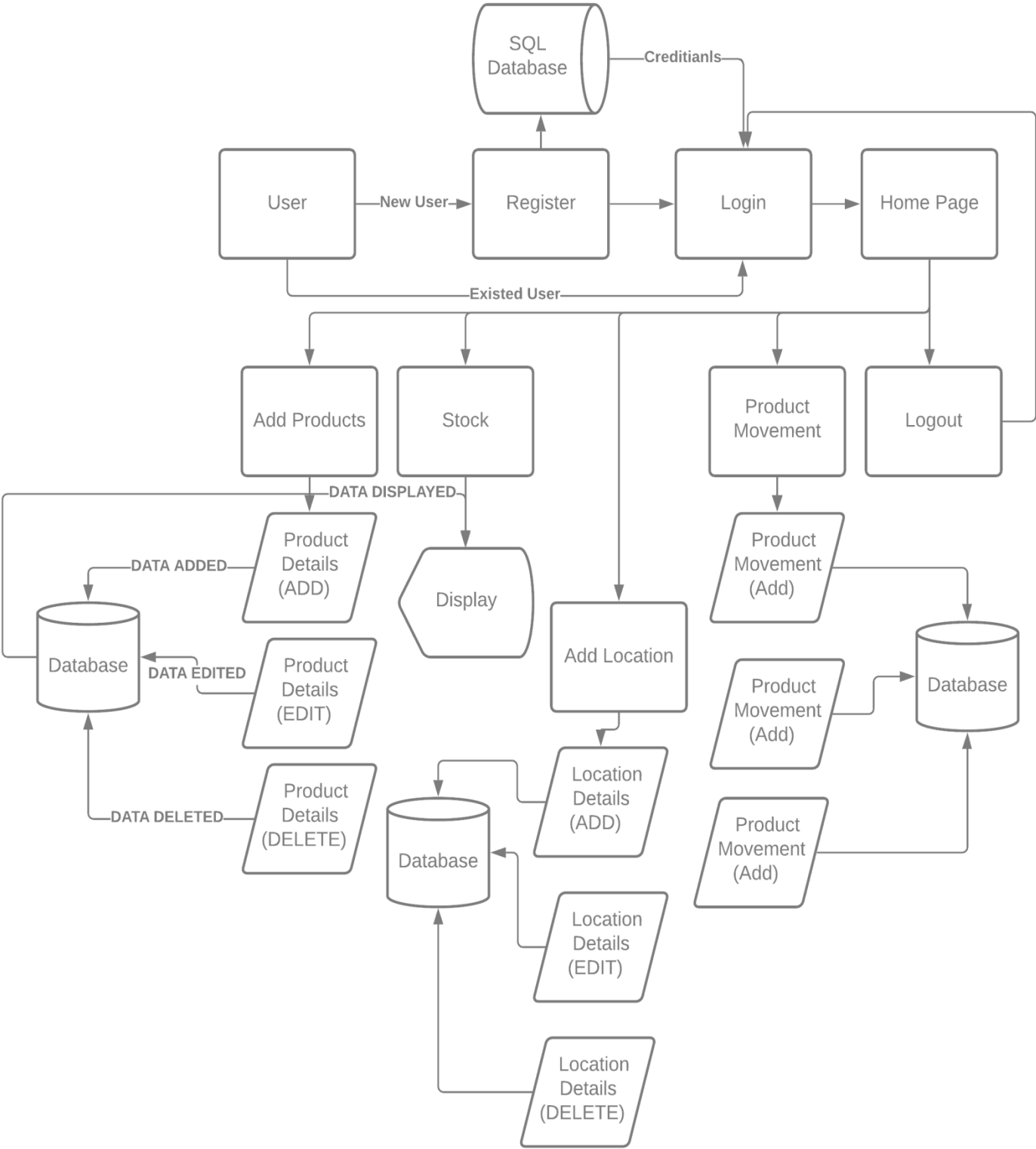
Stock: Users can view their available products in the stock.

Location: Users can add their location.

Product Movement: User can manage their product movements

Logout: The user can logout from the web application

FLOWCHART



CHAPTER 6

PROJECT PLANNING AND SCHEDULING

SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement(Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password or register with username email and password	2	High	4
Sprint-2	Dashboard	USN-5	As a user, I can view the total stocks, threshold for alert instant add or remove stocks.	4	High	4
Sprint-2	Add items to inventory	USN-6	As a user, I can increase the stocks available with the details of the stock.	5	Medium	4
Sprint-3	Stock Update	USN-7	As a user, I can add products which are not available in the dashboard to the stock list.	5	Medium	4
Sprint-4	Request to Customer Care	USN-8	As an user, I can contact the Customer Care Executive and request any services I want from the customer care.	5	Low	4
Sprint-4	Contact Administrator	USN-9	I can be able to report any difficulties I experience and report	5	Medium	4

CHAPTER 7

CODING AND SOLUTIONING

FEATURE 1:

This particular feature contributes to the ultimatum as the home page.

CODING:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<link rel="stylesheet"
```

```
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
```

```
  integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBv  
  ApHHNl/vI1Bx" crossorigin="anonymous">
```

```
<link rel="stylesheet"
```

```
  href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css"
```

```
  integrity="sha384-xeJqLiuOvjUBq3iGOjvSQSIlwlrpqiSHXpduPd6rQpuiM3f5/ijby8pCsn  
  bu5S81n" crossorigin="anonymous">
```

```
<link rel="stylesheet" type="text/css" href="./homescreen.css">
```

```
<link rel="stylesheet" type="text/css"
```

```
  href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.min.css">
```

```
<title>
```

```
  homescreen
```

```
</title>
```

```
<script src="homescreen.js"></script>
```

```
</head>
```

```
<body>
```

```
<nav class="navbar navbar-expand-lg">
```

```
<a class="navbar-brand">Inventory Management</a>
```

```
<button class="navbar-toggler" type="button" data-toggle="collapse"
```

```
  data-target="#navbarNav"
```

```
  aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
```

```
<span class="navbar-toggler-icon"></span>
```

```
</button>
```

```
<div class="collapse navbar-collapse" id="navbarNav">
```

```
<ul class="navbar-nav">
```

```
<li class="nav-item active">
```

```
<a class="nav-link">Home</a>
```

```
</li>
```

```
<li class="nav-item active">
```

```
<a class="nav-link">Add</a>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
</nav>
```

```
<div class="bodywrapper">
```

```
<div class="bodyclass">
```

```
<div class="container" id="main">
```

```
<h4>
```

```
<center>Your Inventories</center>
```

```
</h4>
```

```
<div class="stock">
```

```
<table>
  <tr>
    <td>
      Inventory Name :
    </td>
    <td>
      Halzer
    </td>
  </tr>
  <tr>
    <td>Threshold :</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Total Stocks : </td>
    <td>150</td>
  </tr>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
<div class="inventory 1" id="">
```

```
</div>
```

```
</div>
```

```
<div class="bodyclass">
```

```
<div class="container" id="main">
```

```
<div class="ConfigTitle">
```

```
<h4>Inventory Config</h4>
```

</div>

<div class="configData">

<form>

<div class="formContent">

<label for="threshold">Set Threshold for alert</label>

<input type="text" id="threshold" name="threshold"/>

</div>

<div class="formContent">

<label for="threshold">Change Inventory Location</label>

<input type="text" id="threshold" name="threshold"/>

</div>

<div class="formContent">

<label for="threshold">Add Stock</label>

<input type="text" id="threshold" name="threshold"/>

</div>

<div class="formContent">

<label for="threshold">Remove Stock</label>

<input type="text" id="threshold" name="threshold"/>

</div>

<div class="formContent">

<input class="formButton" type="button" value="Modify"
onclick="doAlert('Modified!')"/>

<input class="formButton" type="button" value="Clear"
onclick="doAlert('Cleared!')"/>

<input class="formButton" type="button" value="Freeze Stock"
onclick="doAlert('Freeze Stock!')"/>

</div>

</form>

</div>

</div>


```
        <div class="inventory 1" id="">
        </div>
    </div>
</div>

</body>

</html>
```

STYLE:

```
.navbar{
    background-color: #ff416c;
    padding: 20px;
}
.navbar-brand{
    color : white;
}

.nav-link{
    color: rgb(184, 175, 175);
}
body{
    background: #f6f5f7;
}
.bodywrapper{
    height: 90vh;
```

```
    justify-content: center;
    align-items: center;
    display: flex;
}
.bodyclass{
    font-family: 'monserrat', sans-serif;
    margin-top: 3%;
    margin-bottom: 3%;
    float: left;
    width: 600px;
}
.container{
    position: relative;
    width: 90%;
    max-width: 100%;
    min-height: 480px;
    margin-left: 2%;
    background: #fff;
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0 14px 28px rgba(0,0,0,0.25),
                0 10px 10px rgba(0,0,0,0.22)
}
.container h4{
    margin-top : 5%;
}

.stock{
    padding: 30%;
}
```

```
/* left */
```

```
.ConfigTitle{  
    height: 10vh;  
    padding: 20px;  
    border: solid 0.1px;  
    border-radius: 5px;  
    margin-top: 5%;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}
```

```
.configData{  
    height: fit-content;  
    padding: 20px;  
    border: solid 0.1px;  
    border-radius: 5px;  
    margin-top: 5%;  
}
```

```
.formContent{  
    display: flex;  
    flex-direction: row;  
    justify-self: center;  
    justify-content: space-between;
```

```
align-items: center;
margin: 20px 0;
}
```

```
.formButton{
margin: 5px 0;
background-color: transparent;
}
```

FEATURE 2:

This page entirely contributes by being the add inventory page.

CODING:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
```

```
integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHH
NI/vI1Bx" crossorigin="anonymous">
```

```
<link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css"
```

```
integrity="sha384-xeJqLiuOvjUBq3iGOjvSqsIlwprqjSHXpduPd6rQpuiM3f5/ijby8pCsnbu5S81
```

```
n" crossorigin="anonymous">
```

```
<link rel="stylesheet" type="text/css" href="./addscreen.css">
```

```
<link rel="stylesheet" type="text/css"
```

```
  href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.min.css">
```

```
<title>
```

```
  homescreen
```

```
</title>
```

```
<script>
```

```
  function doAlert(message){
```

```
    alert(message)
```

```
  }
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<nav class="navbar navbar-expand-lg">
```

```
<a class="navbar-brand">Inventory Management</a>
```

```
<button class="navbar-toggler" type="button" data-toggle="collapse"
```

```
data-target="#navbarNav"
```

```
  aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
```

```
<span class="navbar-toggler-icon"></span>
```

```
</button>
```

```
<div class="collapse navbar-collapse" id="navbarNav">
```

```
<ul class="navbar-nav">
```

```
<li class="nav-item active">
```

```
<a class="nav-link">Home</a>
```

```
</li>
```

```
<li class="nav-item active">
```

```
        <a class="nav-link">Add</a>
    </li>
</ul>
</div>
</nav>
```

```
<div class="stockWrapper">
    <div class="stockArea">
        <div class="ConfigTitle">
            <h4>Add Stock</h4>
        </div>
        <div class="configData">
            <form>
                <div class="formContent">
                    <label for="sname">S Name</label>
                    <input type="text" id="sname" name="sname"/>
                </div>
                <div class="formContent">
                    <label for="quantity">Quantity</label>
                    <input type="text" id="quantity" name="quantity"/>
                </div>
                <div class="formContent">
                    <label for="date">Date</label>
                    <input type="date" id="date" name="date"/>
                </div>
                <div class="formContent">
                    <label for="boughtFrom">Bought From</label>
                    <select id="boughtFrom" name="boughtFrom">
                        <option value="1">Option 1</option>
                        <option value="2">Option 2</option>
```

```
        <option value="3">Option 3</option>
    </select>
</div>
</form>
</div>
</div>
</div>

</body>

</html>
```

STYLE:

```
.navbar{
    background-color: #ff416c;
    padding: 20px;
}
```

```
.navbar-brand{
    color : white;
}
```

```
.nav-link{
    color: rgb(184, 175, 175);
}
```

```
body{
    background: #f6f5f7;
```

```
}
```

```
.stockWrapper{  
  height: 90vh;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
}
```

```
.stockArea{  
  position: relative;  
  width: 60%;  
  max-width: 100%;  
  min-height: 70%;  
  background: #fff;  
  border-radius: 10px;  
  overflow: hidden;  
  box-shadow: 0 14px 28px rgba(0,0,0,0.25),  
             0 10px 10px rgba(0,0,0,0.22)  
}
```

```
.ConfigTitle{  
  height: 10vh;  
  padding: 20px;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```



```
.configData{  
    height: fit-content;  
    padding: 20px;  
}
```

```
.formContent{  
    display: flex;  
    flex-direction: row;  
    justify-self: center;  
    justify-content: space-between;  
    align-items: center;  
    margin: 20px 0;  
}
```

```
.formButton{  
    margin: 5px 0;  
    background-color: transparent;  
}
```

```
input, select{  
    width: 80%;  
    padding: 15px;  
}
```

CHAPTER 8

TESTING

TEST CASES

1. Login button click with wrong credentials entered.
2. Signup with already registered mail ID.
3. Signup with wrong form data entered.
4. Entering home page with logged out session.
5. Clicking homepage buttons with logged out sessions.
6. Invalid data entered in the change password page and requested for change in password.

USER ACCEPTANCE TESTING

S.NO	TEST CASE	REQUIRED OUTPUT	RESULT OUTPUT	STATUS
1	Login button click with wrong credentials	wrong credentials entered notification	wrong credentials entered notification	Accepted
2	Sign up with already registered email ID	Email already registered notification	Email already registered notification	Accepted
3	Sign up with wrong form data entered	wrong credentials entered notification	wrong credentials entered notification	Accepted

4	Entering home page with logged out session	Take user to login page	Take user to login page	Accepted
5	Clicking home page with logged out session	Take user to login page	Take user to login page	Accepted
6	Invalid data entered in change password page and requested for change in password	Wrong form data entered notification	Wrong form data entered notification	Accepted

CHAPTER 9

RESULTS

Sign-Up Page:

Inventory management system

Welcome Back!

To keep connected with us please login with your personal info

SIGN IN

Create Account

Sign Up

Sign-In Page:

Inventroy Management System

Sign in

Sign In

Hello, Friend!

Enter your personal details and start managing your stocks

SIGN UP

Stocks Page:

INVENTORY

Rehman

Administrator

Online

General

Home

Products

Stock

Location

Product Movement

Logout

Product Name

Product QTY

Washing Machine

10

Computer

10

Printer

10

LG Fridge

10

Air Conditioner

10

Television

10

Oven

10

Cooler

10

Fan

10

Sofa

10

Location Page:

INVENTORY

Rehman

Administrator

Online

General

Home

Products

Stock

Location

Product Movement

Logout

Location Information

ADD Location

Location id

Location name

Edit

Delete

1

Hyderabad

Edit

Delete

2

Delhi

Edit

Delete

3

Mumbai

Edit

Delete

4

Kolkata

Edit

Delete

5


Banglore

Edit

Delete

Product Movement Page:

INVENTORY



Rehman
Administrator
Online

General

Home

Products

Stock

Location

Product Movement

Logout

Product Movement

ADD PRODUCT MOVEMENT

Movement id	Product name	Date_Time	From Location	To Location	Qty	Edit	Delete
1	LG Fridge	25/04/2021 : 4:52	Hyderabad	Banglore	5	Edit	Delete
2	Printer	4/28/2021 : 3:23	Delhi	Hyderabad	5	Edit	Delete

CHAPTER 10

ADVANTAGES AND DIS-ADVANTAGES

ADVANTAGES:

1. Better Inventory Accuracy.
2. Reduced risk of overselling.
3. Cost Savings.
4. Avoiding stockouts and Excess stock.
5. More Productivity.
6. Increased Profits.
7. Better Customer Experience

DIS-ADVANTAGES:

1. Expensive for small businesses.
2. Risk of system crashes.
3. Malicious Hacks.
4. Works for only a single company.

8.Applications

1. Wholesale.
2. Distributors.
3. Ecommerce.
4. Retailers.
5. Manufacturing companies.
6. Warehouse

CHAPTER 11

CONCLUSION

In Conclusion, the system categorically stores, deletes, updates and maintains a structured collection of product details and records. We were successfully able to add, delete, modify any product and its attributes. The system was extremely secure due to the hashing technology as well as the security key feature from the library which ensures that any data being sent from the front end will always be encrypted and forwarded to the back end. Hence, there will be no loss of data of any kind.

CHAPTER 12

FUTURE SCOPE

Future work can be done to add a sales page and payments page to manage stocks and product sales simultaneously. The customers can buy products from the sales page with a secured payment gateway in order to avoid fraudulent and scams.

CHAPTER 13

APPENDIX

SOURCE CODE:

<https://github.com/IBM-EPBL/IBM-Project-3352-1658554918/tree/main/Project%20Development%20Phase>

Bibilography

1. [Flask, *flask*.] <https://flask.palletsprojects.com/en/1.1.x/foreword/>
2. [Flask install]
<https://www.twilio.com/docs/usage/tutorials/how-to-set-up-your-python-and-flask-development-environment>
3. [My SQL database creation] <https://remotemysql.com/>
4. [Creating a Flask Project]
<https://code.tutsplus.com/tutorials/creating-a-web-app-from-scratch-using-python-flask-and-mysql--cms-22972>

5. [Creating REST API]
~~<https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask>~~
6. [Designing the UI] <https://spring.io/guides/tutorials/react-and-spring-data-rest/>
7. [Flask and MYSQL Database Connection]
<https://www.askpython.com/python-modules/flask/flask-mysql-database>
8. [Docker Image creation] <https://www.geeksforgeeks.org/dockerize-your-flask-app/>
9. [Depolying APP to TAS]
<https://drive.google.com/file/d/1Y7V9XBm1iCsC37Fjv2UO6fnyGPZJcE6r/view?usp=sharing>
10. [installing cli foundary] <https://docs.cloudfoundry.org/cf-cli/install-go-cli.html>

