

**INVENTORY MANAGEMENT SYSTEM  
FOR RETAILERS**

**HX8001**

**Professional Readiness for Innovation,  
Employability and Entrepreneurship**

**PROJECT REPORT**

Submitted by

<b>R. AjayKrishnan</b>	<b>510619205001</b>
<b>V. Bose</b>	<b>510619205004</b>
<b>R. Deepak</b>	<b>510619205005</b>
<b>S. Karthikeyan</b>	<b>510619205012</b>
<b>V. Ram Kumar</b>	<b>510619205027</b>

*in partial fulfillment for the award of the*

*degree of*

**BACHELOR OF**

**TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

**C.ABDUL HAKEEM COLLEGE OF**

**ENGGINERING AND TECHNOLOGY**

**ANNA UNIVERSITY: CHENNAI 600 025**

**NOVEMBER 2022**

# TABLE OF CONTENT

1. **INTRODUCTION**
  - 1.1 Project Overview
  - 1.2 Purpose
2. **LITERATURE SURVEY**
  - 2.1 Existing problem
  - 2.2 References
  - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
  - 3.1 Empathy Map Canvas
  - 3.2 Ideation & Brainstorming
  - 3.3 Proposed Solution
  - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
  - 4.1 Functional requirement
  - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
  - 5.1 Data Flow Diagrams
  - 5.2 Solution & Technical Architecture
  - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
  - 6.1 Sprint Planning & Estimation
  - 6.2 Sprint Delivery Schedule
  - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
  - 7.1 Feature 1
  - 7.2 Feature 2
  - 7.3 Database Schema (if Applicable)
8. **TESTING**
  - 8.1 Test Cases
  - 8.2 User Acceptance Testing
9. **RESULTS**
  - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
  - Source Code
  - GitHub & Project Demo Link

## **1.INTRODUCTION**

Retail inventory management is the process of ensuring you carry merchandise that shoppers want, with neither too little nor too much on hand. By managing inventory, retailers meet customer demand without running out of stock or carrying excess supply.

### **1.1 PROJECT OVERVIEW**

In practice, effective retail inventory management results in lower costs and a better understanding of sales patterns. Retail inventory management tools and methods give retailers more information on which to run their businesses. Retailers can access their accounts by logging into the application.

Once retailers successfully log in to the application they can update their inventory details, also users will be able to add new stock by submitting essential details related to the stock.

### **1.2 PURPOSE**

Retail inventory management works by creating systems to log products, receive them into inventory, track changes when sales occur, manage the flow of goods from purchasing to final sale and check stock counts.

Inventory management helps companies identify which and how much stock to order at what time. It tracks inventory from purchase to the sale of goods. The practice identifies and responds to trends to ensure there's always enough stock to fulfill customer orders and proper warning of a shortage.

## **2.LITERATURE SURVEY**

### **1.1 RETAILER INVENTORY STRATEGY BASED ON SYSTEM DYNAMICS SIMULATION**

System dynamics (SD) was created during the mid-1950s by Professor Jay Forrester of the Massachusetts Institute of Technology.

The supply chain inventory management aims at meeting customers' demands, reducing inventory cost and increasing enterprise profit. We need place an order and replenish productions when the inventory is under safety stock quantity.

### **DISADVANTAGE:**

One of the disadvantages of this methodology is the development of the system dynamics model of complex systems since an extensive study has to be carried out, so that this system works correctly in the system dynamics model, and not on the contrary, affect the project where the dynamics of the system is being carried out.

### **REFERENCES:**

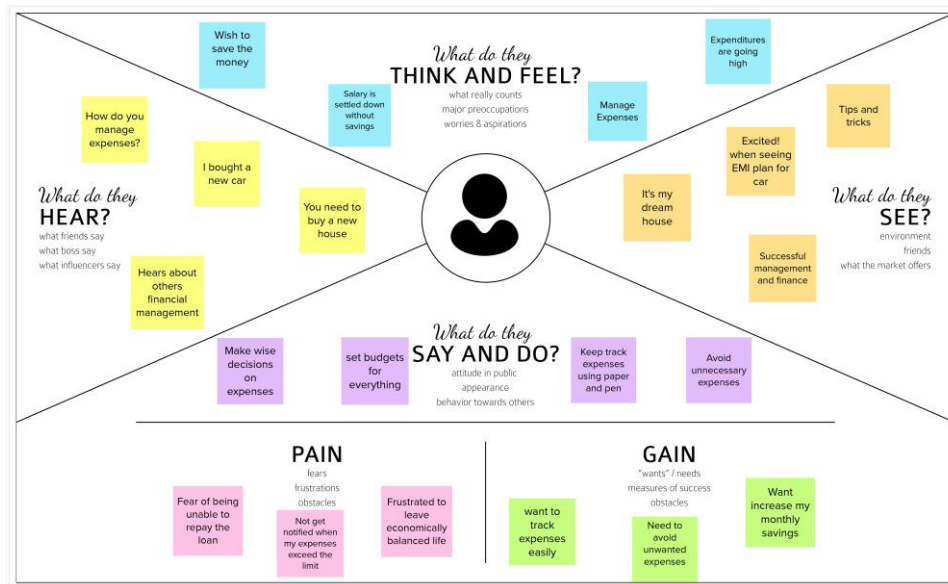
- [1] A. Sapietova and V. Dek 'y's, "Dynamic analysis of rotating machines in ~ msc. adams," Procedia Engineering, vol. 136, no. 143-149, p. 12, 2016.
- [2] T. Rebs, M. Brandenburg, and S. Seuring, "System dynamics modeling for sustainable supply chain management: A literature review and systems thinking approach," Journal of cleaner production, vol. 208, pp. 1265–1280, 2019.

## **PROBLEM STATEMENT:**

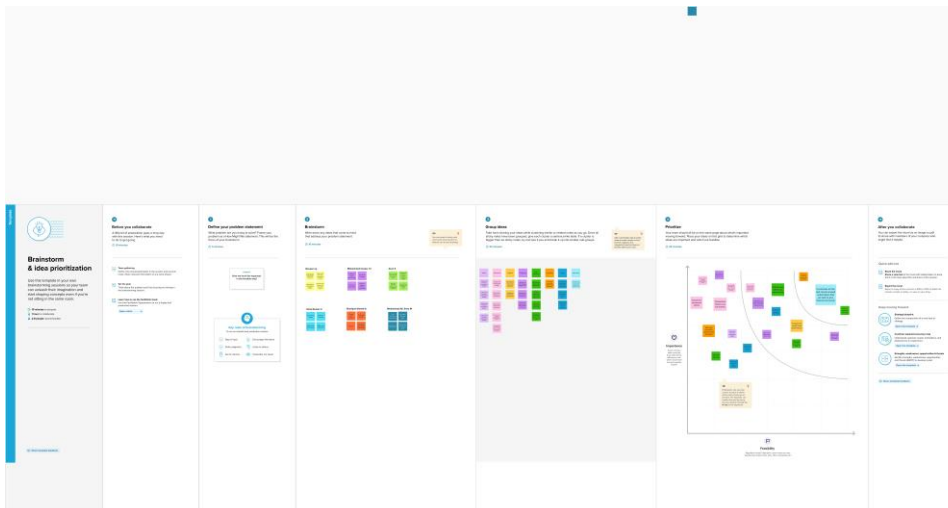
In inventory systems, demand is usually uncertain, and the lead-time can also vary. To avoid shortages, managers often maintain a safety stock. In such situations, it is not clear what order quantities and reorder points will minimize expected total inventory cost. Simulation models can address this question.

## 3.IDEATION & PROPOSED SOLUTION

### 3.1 EMPATHY MAP CANVAS:



### 3.2 BRAINSTORMING:



### **3.3 PROPOSED SOLUTION:**

Problem Statement (Problem to be solved) • The retailers generally facing issues in recording the stocks and its threshold limit available.

❖ Idea / Solution description

❖ This proposed system will have a daily update system whenever a product is sold or it is renewed.

❖ The product availability is tracked daily and an alert system is again kept on to indicate those products which falls below the threshold limit.

❖ The application allows the customers to know all the present time available stocks and also when the new stock will be available on the store for them to buy.

❖ Novelty / Uniqueness • Certain machine learning algorithms are used to predict the seasonal high selling products which can be made available during that time.

❖ Prediction of the good selling brand of all certain products based on their popularity, price and customer trust and satisfaction will be implemented.

❖ Notifications will be sent to the retailers if any product that the customers have been looking for is not available.

❖ Social Impact / Customer Satisfaction • The customers will be highly satisfied since the wastage of time while searching for an unavailable product is reduced.

### **3.4 PROBLEM SOLUTION FIT:**

1. CUSTOMER SEGMENTION(S) - ➤ It is an important marketing tool. ➤ Customer segmentation helps the enterprises increase profit and improve customer service level.
2. CONSTRAINTS - ➤ Constraints include limits on ➤ Raw Materials ➤ Machine Capacity  
➤ Work force capacity ➤ Inventory Investment ➤ Storage Space
3. CHALLENGES - ➤ Inconsistent Tracking ➤ Warehouse Efficiency ➤ Inaccurate Data ➤ Changing Demand ➤ Limited Visibility ➤ Manual Documentation ➤ Problem Stock ➤ Insufficient Order Management
4. PROBLEM ROOT CAUSE - ➤ Visibility Problems ➤ Lack of real-time reporting  
Inefficiency ➤ Under Stocking ➤ Over Stocking ➤ Lack of trend forecasting
5. YOUR SOLUTION - • Developing a database in that database if the product is in out of stock sending a SMS alerting message • provide a option for graphical view of sales
6. CHANNELS of BEHAVIOUR -1. ONLINE • Alerting the particular person about the stocks limits, either full or empty or even about the reach of a particular limit  
2. OFFLINE • Manual Checking • Stock Distribution among the Inventory



## **4. REQUIREMENT ANALYSIS**

### **4.1 FUNCTIONAL REQUIREMENTS:**

FR-1 Account Creation Created through Email Creation through Github Creation through LinkedIn Creation via Google

FR-2 User Confirmation

Confirmation via Email FR-3

Successful Log in Notification

through Email

FR-4 Update inventory details

Notification through Email FR-5

Add new stock Notification through

Email

FR-6 Unavailability of stock Alert notification through Email

## **4.2 NON - FUNCTIONAL RREQUIREMENTS:**

NFR-1 Usability - When the account is created in the application , all the details about the stock has to be filled. And when the stocks are no more left the application will get to know this and does the process.

NFR-2 Security - The Application has a very good security , because not everyone can access the application. Only the employees who have the access can only access the application. The Application requires Authentication. Because of this no one can misuse this application.

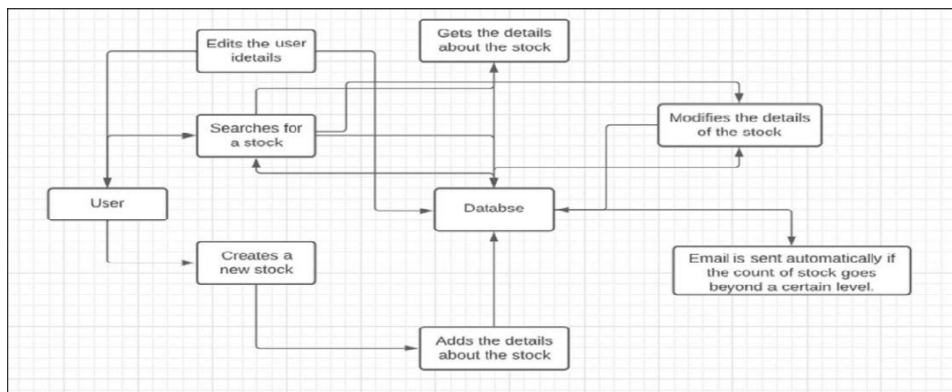
NFR-3 Performance - The products can be tracked by the users. The tracking can enhance the timed deliveries. With the help of this the man power can also be reduced. Emails will be delivered automatically when there is shortage of products.

NFR-4 Availability - Inventory management systems are designed to monitor product availability, determine purchasing schedules for better customer interaction.

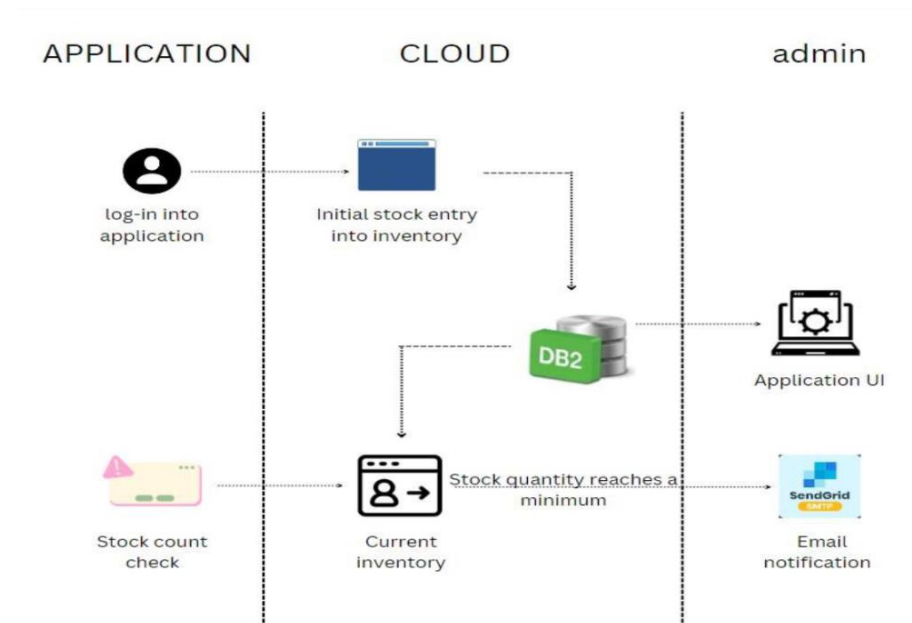
NFR-5 Scalability - Scalability is an aspect or rather a functional quality of a system, software or solution. This proposed system for inventory management system can accommodate expansion without restricting the existing workflow and ensure an increase in the output or efficiency of the process.

## 5. PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAM:



### 5.2 SOLUTION & TECHNICAL ARCHITECTURE:



## **5.3USER STORIES:**

### ***Customer (Mobile user) Registration***

**USN-1** As a user, I can register for the application by entering my email, password, and confirming my password. I can access my account / dashboard High Sprint-1 **USN-2** As a user, I will receive confirmation email once I have registered for the application I can receive confirmation email & click confirm High Sprint-1

**USN-3** As a user, I can register for the application through Facebook I can register & access the dashboard with Facebook Login Low Sprint-2

**USN-4** As a user, I can register for the application through Gmail I can receive confirmation email & click confirm Medium Sprint-1 Login

**USN-5** As a user, I can log into the application by entering email & password I can enter into my account High Sprint-1 Dashboard

**USN-6** As a user, it displays all top brands and offers of product I can avail the recent offer in sale High Sprint-2 Customer (Web user) Application

**USN-7** As a user, I can register, login and shop the products easily I can access my shopping very soon High Sprint-3 Customer Care Executive Update Inventory details

**USN-8** To keep track of order and availability of stock in inventory I can control the inventory stock correctly High Sprint-4 Administrator Add new stock

**USN-9** To add new products into the application I can provide new trend products in an application for customer needs High Sprint-3 Customer Care Executive Verify customer feedback

**USN-10** To design application that meets user's desires I can satisfy the customer expectations High Sprint-4 Customer Care Executive Inventory Control

**USN-11** To refill the unavailability of stock in inventory I can alert notification through email Medium Sprint-2 Administrator Performance of Application

## 6.PROJECT PLANNING & SCHEDULING

### 6.1 SPRINT PLANNING & ESTIMATION

Project Tracker, Velocity & Burndown Chart: (4 Marks)

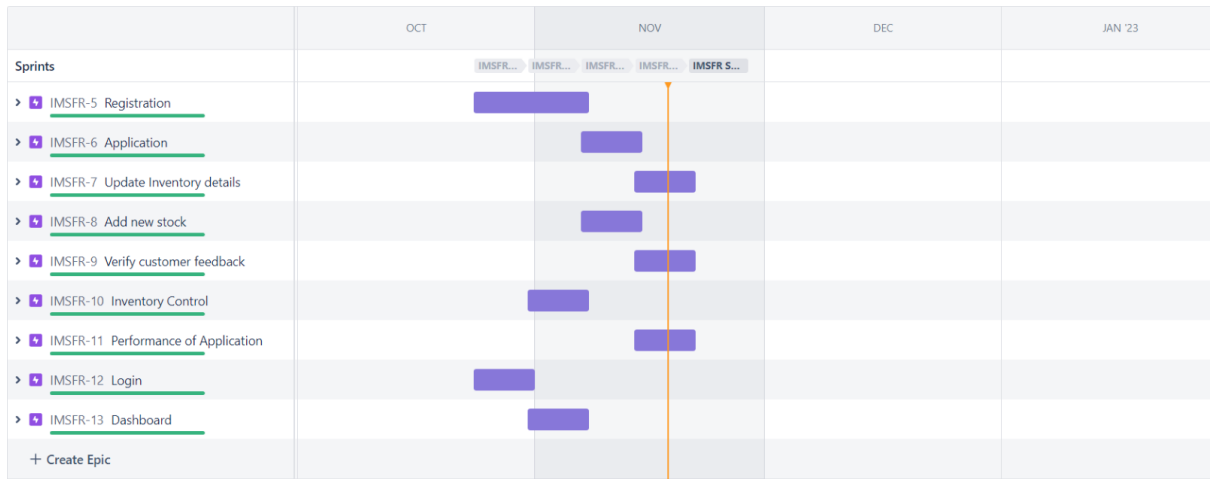
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

**Velocity:**

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

## 6.2 REPORTS FROM JIRA:



## **7. CODING & SOLUTIONING:**

### **7.1 FEATURE 1:**

#### **OPTIMIZING YOUR INVENTORY**

You want to maintain the right amount of inventory required to meet demand, keep logistics costs low, and avoid common inventory issues such as stock outs, overstocking, and backorders.

- **Inventory Forecasting**

A company has a 30-day forecast for cookies. If they sold 43 units over the previous 30 days, the base demand would be 43.

It gives a starting point to show you how you can increase accuracy. You can gather trends like these from your data that may influence demand.

- **Demand Planning**

A grocery store wants to prepare its inventory levels for next year at the end of Thanksgiving. They look at sales, competition, and growth. They can now plan and launch new deals to position themselves as the go-to Thanksgiving destination.

Your inventory may need different optimization and planning. For example, you ordered the exact amount of raw materials. Later on, with variation in supply, you wish you had extra safety stock.

## 7.2 FEATURE 2:

## SECURITY AND BACKUP

You have to make sure that your inventory is safe to be in control of stored goods. You need security for warehouse management as safety and efficiency lead to good performance. To help keep operations running smoothly, ensure you address supply chain and inventory control strategies and have backup plans in place before there is an issue.

## 8. TESTING





## 8.1 Test Cases:

## 8.2 USER ACCEPTANCE TESTING

### 1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Inventory Management System for Retailers project at the time of the release to User Acceptance Testing (UAT).

### 2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	4	5	2	18
Duplicate	1	0	6	0	7
External	2	4	0	1	7
Fixed	12	2	5	20	39
Not Reproduced	0	0	2	0	2
Skipped	0	0	2	1	3
Won't Fix	0	5	2	1	8
Totals	22	15	22	25	84

## 9.RESULTS:

### Inventroy Management System

#### Sign in

Enter Your email

Enter Your username

Enter Your Password

Sign In

#### Hello, Friend!

Enter your personal details and start managing your stocks

SIGN UP

## Dashboard Welcome Deepak

### Bangalore

Product	Warehouse	Qty
P1	Bangalore	0
P2	Bangalore	30

### Mumbai

Product	Warehouse	Qty
P1	Mumbai	50

... ..

# Products

Add Product

Product ID	Product Cost	Product Quantity		
P1			Edit	Delete
P2			Edit	Delete
P3			Edit	Delete
P4			Edit	Delete

# Locations

Add Location

Location ID		
Bangalore	Edit	Delete
Mumbai	Edit	Delete
Nasik	Edit	Delete

## Product Movements

Add Product Movements

Movement ID	Time	From Location	To Location	Product ID	Quantity	
40	2019-02-01 05:09:50	Bangalore	--	P1	100	Delete
41	2019-02-01 05:10:13	Bangalore	Mumbai	P1	100	Delete
42	2019-02-01 05:10:48	Mumbai	Pune	P1	50	Delete
43	2019-02-01 05:11:40	Bangalore	--	P2	30	Delete

```
Command Prompt
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\deepa>d:
D:\>CD D:\Final Code\Final Code
D:\Final Code\Final Code>FLASK RUN
Usage: flask [OPTIONS] COMMAND [ARGS]...
Try 'flask --help' for help.

Error: No such command 'RUN'.

D:\Final Code\Final Code>flask run
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [19/Nov/2022 11:41:41] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 11:41:43] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [19/Nov/2022 11:41:47] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 11:41:47] "GET /style.css HTTP/1.1" 404 -
127.0.0.1 - - [19/Nov/2022 11:44:16] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [19/Nov/2022 11:44:17] "GET /dashboard HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 11:45:38] "GET /products HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 11:47:40] "GET /locations HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 11:48:25] "GET /product_movements HTTP/1.1" 200 -

D:\Final Code\Final Code>
```

Thus the project has been successfully implemented and executed.

## **10.ADVANTAGES & DISADVANTAGES:**

### **ADVANTAGES:**

**Saves Time** - Paper-based retail inventory management can take a lot of time and effort.

**Eliminates Errors** - Traditional retail inventory processes can be vulnerable to errors.

**Improves Transparency** - In the retail industry, the visibility of the real-time status of the various items in the inventory is very critical.

**Cost-Effective** - Manual inventory control would increase your labor and process costs.

**Efficient Stock Counting** - If done manually, stock counting is a tedious and error-prone process.

## DISADVANTAGES:

**Production problem:** even though inventory management can reveal to you the amount of stock you have at hand and the amount that you have sold off, it can also hide production problems that could lead to customer service disasters.

**Complexity:** some methods and strategies of inventory management can be relatively complex and difficult to understand on the part of the staff.

**High implementation costs:** some inventory management systems can come at a high price because the business needs to install specialized systems and software in order to use them.

**Bureaucracy:** even though inventory management allows employees at every level of the company to read and manipulate company stock and product inventory, the infrastructure required to build such a system adds a layer of bureaucracy to the whole process and the business in general.

The control of inventory is complex because of the many functions it performs. It should thus be viewed as a shared responsibility.

## **10. CONCLUSION**

The project inventory management for retailers has been successfully implemented by using python, flask, html, css , Java script and the database created by using IBM db2 and also successfully executed and implemented.

## **11. FUTURE SCOPE**

The scope of an inventory system can cover many needs, including valuing the inventory, measuring the change in inventory and planning for future inventory levels. The value of the inventory at the end of each period provides a basis for financial reporting on the balance sheet. Measuring the change in inventory allows the company to determine the cost of inventory sold during the period. This allows the company to plan for future inventory needs.

## 12. APPENDIX

### Source code

```
<!DOCTYPE html>

<html lang="en" >

<head>

  <meta charset="UTF-8">

  <title>Inventory Management System</title>

  <link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.min.css'><link rel="stylesheet" href="./style.css">

  <!--internal css-->

  <style>

    @import url('https://fonts.googleapis.com/css?family=Montserrat:400,800');

* {

  box-sizing: border-box;

}

body {

  background: #f6f5f7;

  display: flex;

  justify-content: center;

  align-items: center;

  flex-direction: column;

  font-family: 'Montserrat', sans-serif;

  height: 100vh;

  margin: -20px 0 50px;

}
```



```
h1 {  
    font-weight: bold;  
    margin: 0;  
}
```

```
h2 {  
    text-align: center;  
}
```

```
p {  
    font-size: 14px;  
    font-weight: 100;  
    line-height: 20px;  
    letter-spacing: 0.5px;  
    margin: 20px 0 30px;  
}
```

```
span {  
    font-size: 12px;  
}
```

```
a {  
    color: #333;  
    font-size: 14px;  
    text-decoration: none;  
    margin: 15px 0;  
}
```

```
button {  
    border-radius: 20px;  
    border: 1px solid #FF4B2B;  
    background-color: #FF4B2B;  
    color: #FFFFFF;  
    font-size: 12px;  
    font-weight: bold;  
    padding: 12px 45px;  
    letter-spacing: 1px;  
    text-transform: uppercase;  
    transition: transform 80ms ease-in;  
}
```

```
button:active {  
    transform: scale(0.95);  
}
```

```
button:focus {  
    outline: none;  
}
```

```
button.ghost {  
    background-color: transparent;  
    border-color: #FFFFFF;  
}
```

```
form {  
    background-color: #FFFFFF;  
    display: flex;
```

```
    align-items: center;
    justify-content: center;
    flex-direction: column;
    padding: 0 50px;
    height: 100%;
    text-align: center;
}
```

```
input {
    background-color: #eee;
    border: none;
    padding: 12px 15px;
    margin: 8px 0;
    width: 100%;
}
```

```
.container {
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 14px 28px rgba(0,0,0,0.25),
                0 10px 10px rgba(0,0,0,0.22);
    position: relative;
    overflow: hidden;
    width: 768px;
    max-width: 100%;
    min-height: 480px;
}
```

```
.form-container {
```

```
    position: absolute;
    top: 0;
    height: 100%;
    transition: all 0.6s ease-in-out;
}
```

```
.sign-in-container {
    left: 0;
    width: 50%;
    z-index: 2;
}
```

```
.container.right-panel-active .sign-in-container {
    transform: translateX(100%);
}
```

```
.sign-up-container {
    left: 0;
    width: 50%;
    opacity: 0;
    z-index: 1;
}
```

```
.container.right-panel-active .sign-up-container {
    transform: translateX(100%);
    opacity: 1;
    z-index: 5;
    animation: show 0.6s;
}
```

```
@keyframes show {  
  0%, 49.99% {  
    opacity: 0;  
    z-index: 1;  
  }  
  
  50%, 100% {  
    opacity: 1;  
    z-index: 5;  
  }  
}
```

```
.overlay-container {  
  position: absolute;  
  top: 0;  
  left: 50%;  
  width: 50%;  
  height: 100%;  
  overflow: hidden;  
  transition: transform 0.6s ease-in-out;  
  z-index: 100;  
}
```

```
.container.right-panel-active .overlay-container{  
  transform: translateX(-100%);  
}
```

```
.overlay {
```

```
background: #FF416C;
background: -webkit-linear-gradient(to right, #FF4B2B, #FF416C);
background: linear-gradient(to right, #FF4B2B, #FF416C);
background-repeat: no-repeat;
background-size: cover;
background-position: 0 0;
color: #FFFFFF;
position: relative;
left: -100%;
height: 100%;
width: 200%;
transform: translateX(0);
transition: transform 0.6s ease-in-out;
}
```

```
.container.right-panel-active .overlay {
    transform: translateX(50%);
}
```

```
.overlay-panel {
    position: absolute;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    padding: 0 40px;
    text-align: center;
    top: 0;
    height: 100%;
    width: 50%;
}
```

```
    transform: translateX(0);
    transition: transform 0.6s ease-in-out;
}
```

```
.overlay-left {
    transform: translateX(-20%);
}
```

```
.container.right-panel-active .overlay-left {
    transform: translateX(0);
}
```

```
.overlay-right {
    right: 0;
    transform: translateX(0);
}
```

```
.container.right-panel-active .overlay-right {
    transform: translateX(20%);
}
```

```
.social-container {
    margin: 20px 0;
}
```

```
.social-container a {
    border: 1px solid #DDDDDD;
}
```

```
border-radius: 50%;  
display: inline-flex;  
justify-content: center;  
align-items: center;  
margin: 0 5px;  
height: 40px;  
width: 40px;  
}
```

```
footer {  
    background-color: #222;  
    color: #fff;  
    font-size: 14px;  
    bottom: 0;  
    position: fixed;  
    left: 0;  
    right: 0;  
    text-align: center;  
    z-index: 999;  
}
```

```
footer p {  
    margin: 10px 0;  
}
```

```
footer i {  
    color: red;  
}
```



```

    footer a {
        color: #3c97bf;
        text-decoration: none;
    }
</style>

</head>
<body>
<!-- partial:index.partial.html -->
<h2>Inventory Management System</h2>
    <div class="container" id="container">
        <div class="form-container sign-in-container">
            <form action="{{ url_for('login') }}" method="post">
                <h1>Sign in</h1>
                {{ msg }}
                <input id="email" name="email" type="email" placeholder="Enter Your
email"/>
                <input id="username" name="username" type="text" placeholder="Enter
Your username"/>
                <input id="password" name="password" type="password"
placeholder="Enter Your Password"/>
                <input type="submit" class="btn" value="Sign In">
            </form>
        </div>
        <div class="overlay-container">
            <div class="overlay">
                <div class="overlay-panel overlay-left">
                    <h1>Welcome Back!</h1>
                    <p>To keep connected with us please login with your personal
info</p>
                    <button class="ghost" id="signIn">Sign In</button>
                </div>

```

```

        <div class="overlay-panel overlay-right">
            <h1>Hello, Friend!</h1>
            <p>Enter your personal details and start managing your
stocks</p>
            <button class="ghost" id="signUp">Sign Up</button>
        </div>
    </div>
</div>
</div>
</div>

<footer>
    <p>
        Created with <i class="fa fa-heart"></i> by
        <a>Deepak and Team</a>
    </p>
</footer>
<!-- partial -->
<script>
    const signUpButton = document.getElementById('signUp');
const signInButton = document.getElementById('signIn');
const container = document.getElementById('container');

signUpButton.addEventListener('click', () => {
    location.href = "/";
});

signInButton.addEventListener('click', () => {
    container.classList.remove("right-panel-active");
});
</script>

```

</body>

</html>

Git    hub    Link:    <https://github.com/IBM-EPBL/IBM-Project-18483-1659686013>

Demo	Video	Link:
<a href="https://drive.google.com/file/d/1ikob2lK8HMa2UcX5PhTaEPA-qLzvqnqo/view?usp=share_link">https://drive.google.com/file/d/1ikob2lK8HMa2UcX5PhTaEPA-qLzvqnqo/view?usp=share_link</a>		

