Project Report

Inventory Management System for Retailers

Team ID - PNT2022TMID17815

Team Lead – M. Manojkumar	(Roll Number :	9517201904088)
Team Member 1 – T. Madhan	(Roll Number :	9517201904081)
Team Member 2 – M. Balasubramanian	(Roll Number :	9517201904028)
Team Member 3 – V. Bhuvanesh	(Roll Number :	9517201904030)

1. INTRODUCTION

1.1. Project Overview

Retail inventory management system needs to meet customer demand without running out of stock or carrying excess supply. Effective retail inventory management results in lower costs and a better understanding of sales patterns. Applications have been developed to help retailers track and manage stocks related to their own products. The System will ask retailers to create their accounts by providing essential details. 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. They can view details of the current inventory. The System will automatically send an email alert to the retailers if there is no stock found in their accounts. So that they can order new stock.

1.2. Purpose

- To develop Retail inventory management system that needs to maintain their own stock details.
- To create User (Retailer) accounts by providing essential details and logging into the application.
- To automatically send an email alert to the retailers if there is no stock found in their accounts.

2. LITERATURE SURVEY

2.1. Existing problem

In the retailers' inventory management process, they cannot not properly maintain their presence of goods, means that product availability.

2.2. References

- Xun Wang, Stephen M. Disney, Borja Ponte, On the stationary stochastic response of an order-constrained inventory system, European Journal of Operational Research, Volume 304, Issue 2, 2023, Pages 543-557, ISSN 0377-2217, doi : 10.1016/j.ejor.2022.04.020
- Mahmood Vahdani, Zeinab Sazvar, Coordinated inventory control and pricing policies for online retailers with perishable products in the presence of social learning, Computers & Industrial Engineering, Volume 168, 2022, 108093, ISSN 0360-8352, doi: 10.1016/j.cie.2022.108093.
- Rajesh Bose, Haraprasad Mondal, Indranil Sarkar, Sandip Roy, Design of smart inventory management system for construction sector based on IoT and cloud computing, e-Prime - Advances in Electrical Engineering, Electronics and Energy, Volume 2, 2022, 100051, ISSN 2772-6711, doi : 10.1016/j.prime.2022.100051.

- X. Qiao, Z. Wang and H. Chen, "Joint Ordering and Markdown Policy for Short Lifetime Products With Competitive Price- and Freshness-Based Demand," in IEEE Transactions on Automation Science and Engineering, vol. 18, no. 4, pp. 1956-1968, Oct. 2021, doi: 10.1109/TASE.2020.3027302.

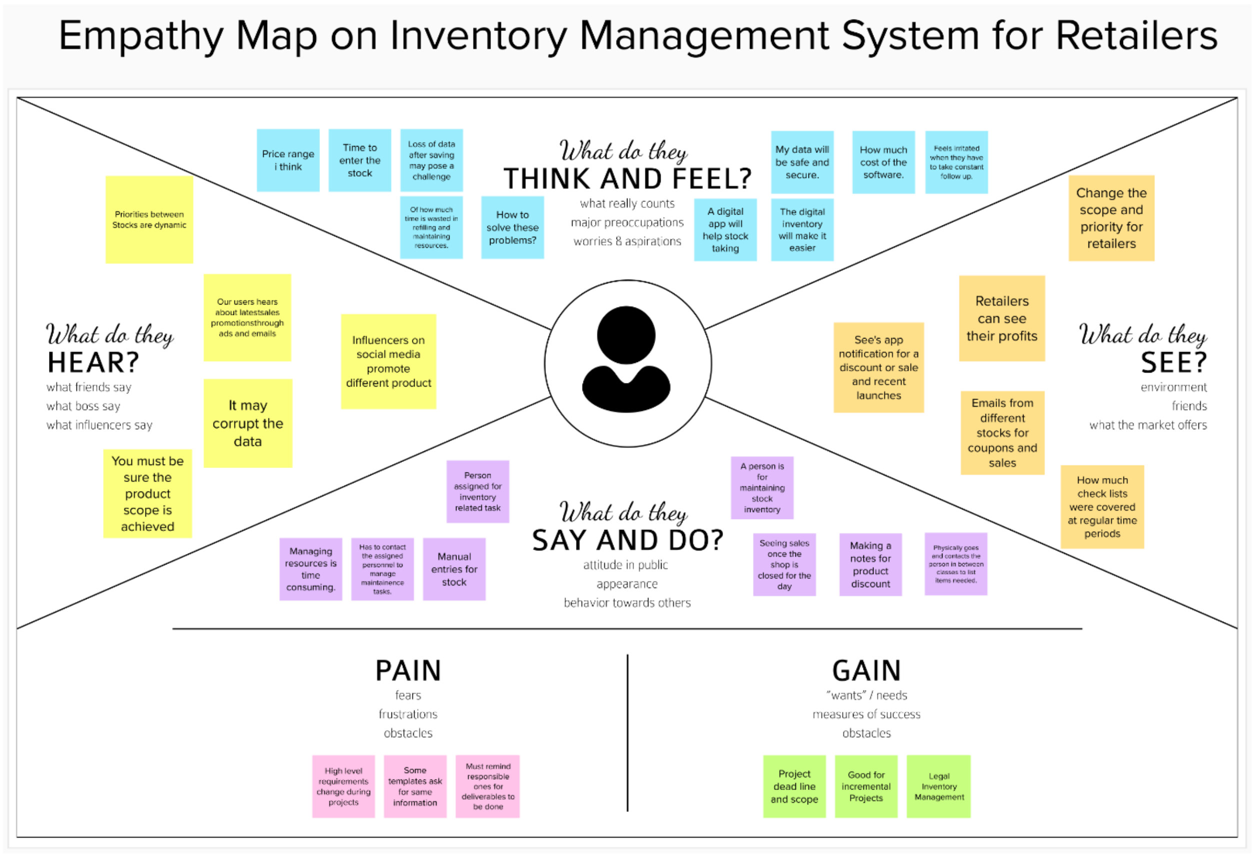
2.3. Problem Statement Definition

The retailers can maintain their own Inventories. And the retailer gets notification if their inventory goes under the ranged limits.

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



Empathy Map on Inventory Management System for Retailers
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Price range i think
Time to enter the stock
Loss of data after saving may pose a challenge
Of how much time is wasted in refilling and maintaining resources.
How to solve these problems?
A digital app will help stock taking
My data will be safe and secure.
The digital inventory will make it easier
How much cost of the software.
Feels irritated when they have to take constant follow up.
What do they HEAR?
what friends say
what boss say
what influencers say
Priorities between Stocks are dynamic
Our users hears about latestsales promotionsthrough ads and emails
Influencers on social media promote different product
It may corrupt the data
You must be sure the product scope is achieved
What do they SEE?
environment
friends
what the market offers
Change the scope and priority for retailers
Retailers can see their profits
See's app notification for a discount or sale and recent launches
Emails from different stocks for coupons and sales
How much check lists were covered at regular time periods
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Person assigned for inventory related task
Managing resources is time consuming.
Has to contact the assigned personnel to manage maintainence tasks.
Manual entries for stock
A person is for maintaining stock inventory
Seeing sales once the shop is closed for the day
Making a notes for product discount
Physically goes and contacts the person in between classes to list items needed.
PAIN
fears
frustrations
obstacles
High level requirements change during projects
Some templates ask for same information
Must remind responsible ones for deliverables to be done
GAIN
“wants” / needs
measures of success
obstacles
Project dead line and scope
Good for incremental Projects
Legal Inventory Management

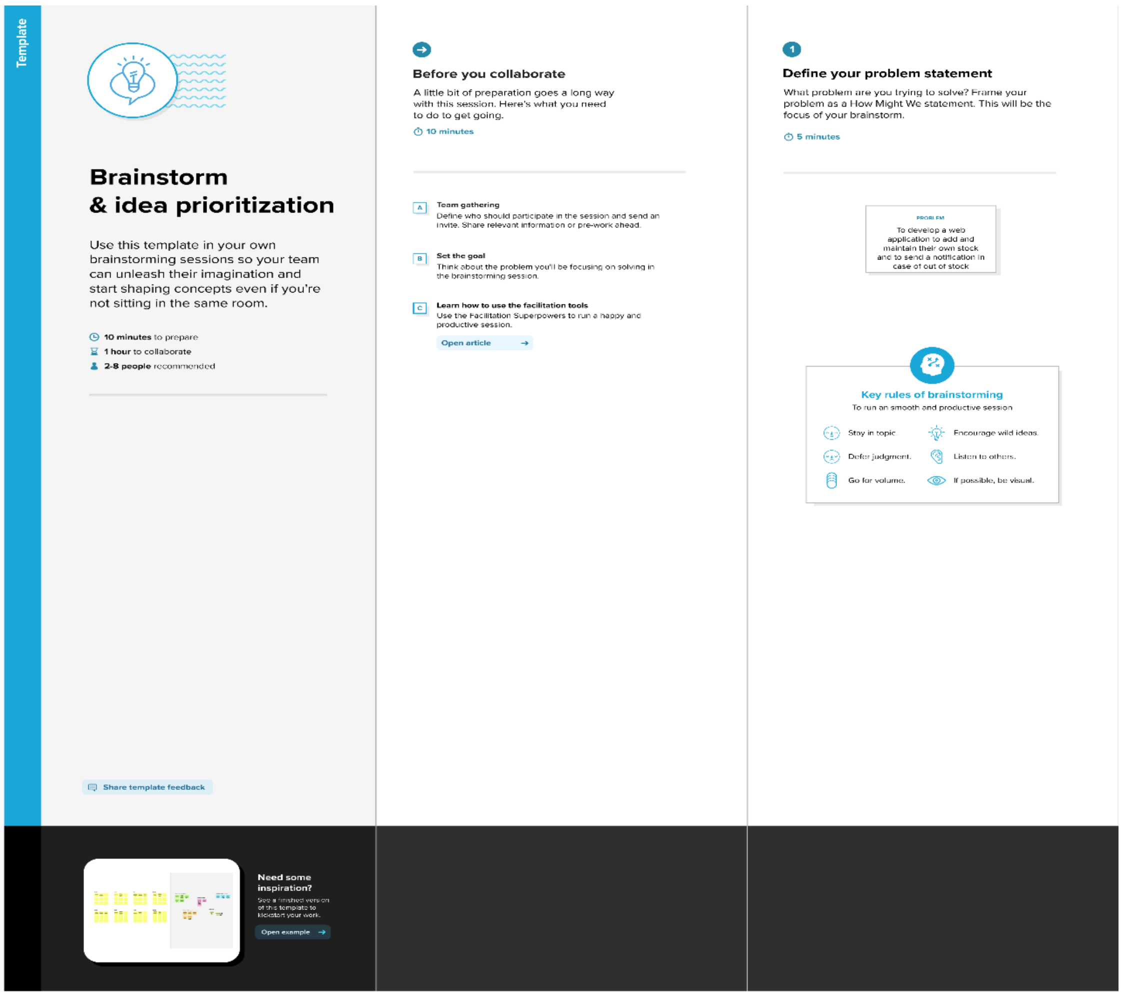
3.2. Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: Brainstorm & Idea Prioritization

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Need some inspiration?
See a finished version of this template to kickstart your work.
Open example

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.
Open article

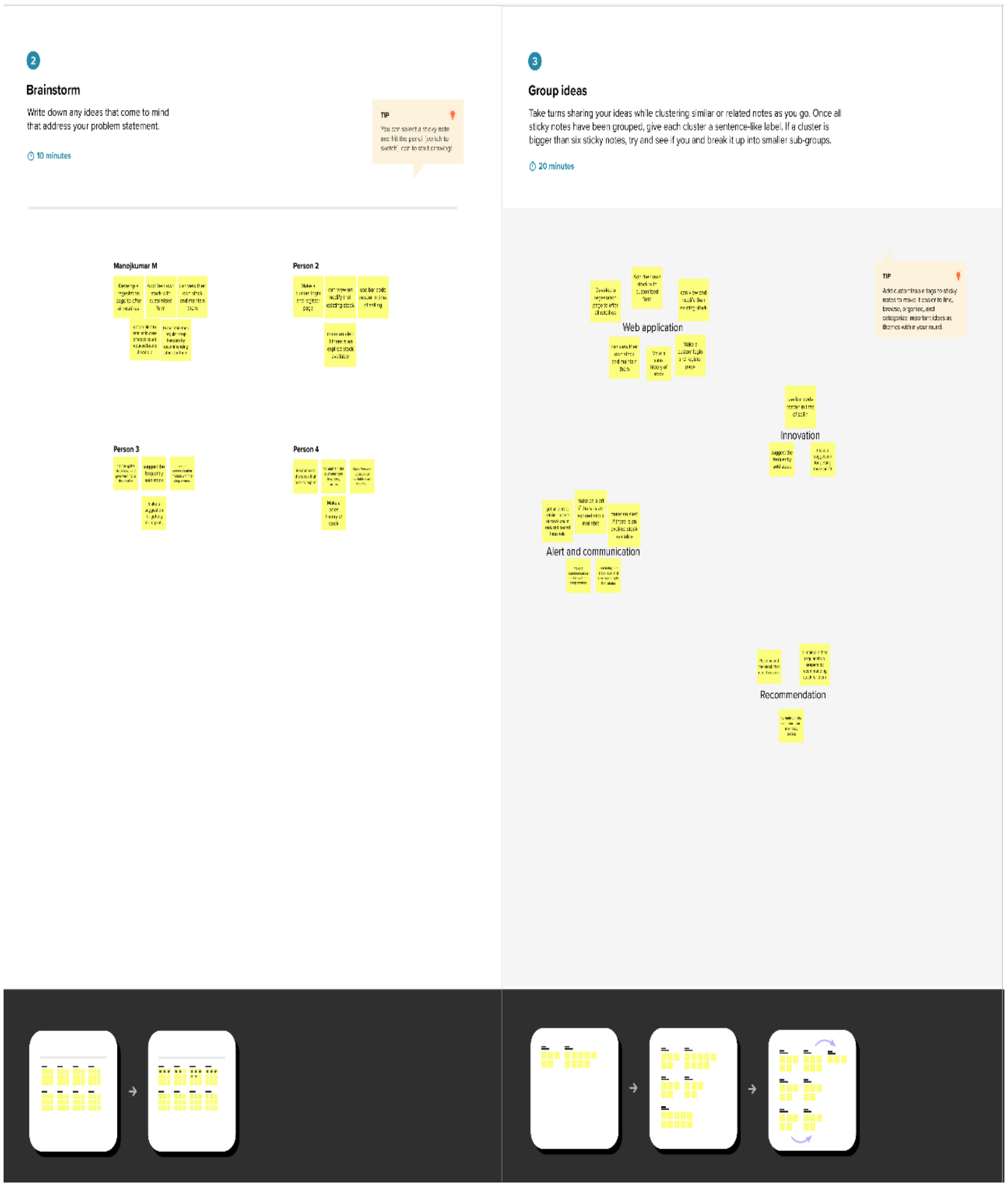
1 **Define your problem statement**
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes

PROBLEM
To develop a web application to add and maintain their own stock and to send a notification in case of out of stock

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer judgment.	Listen to others.
Go for volume.	If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

3

Group ideas

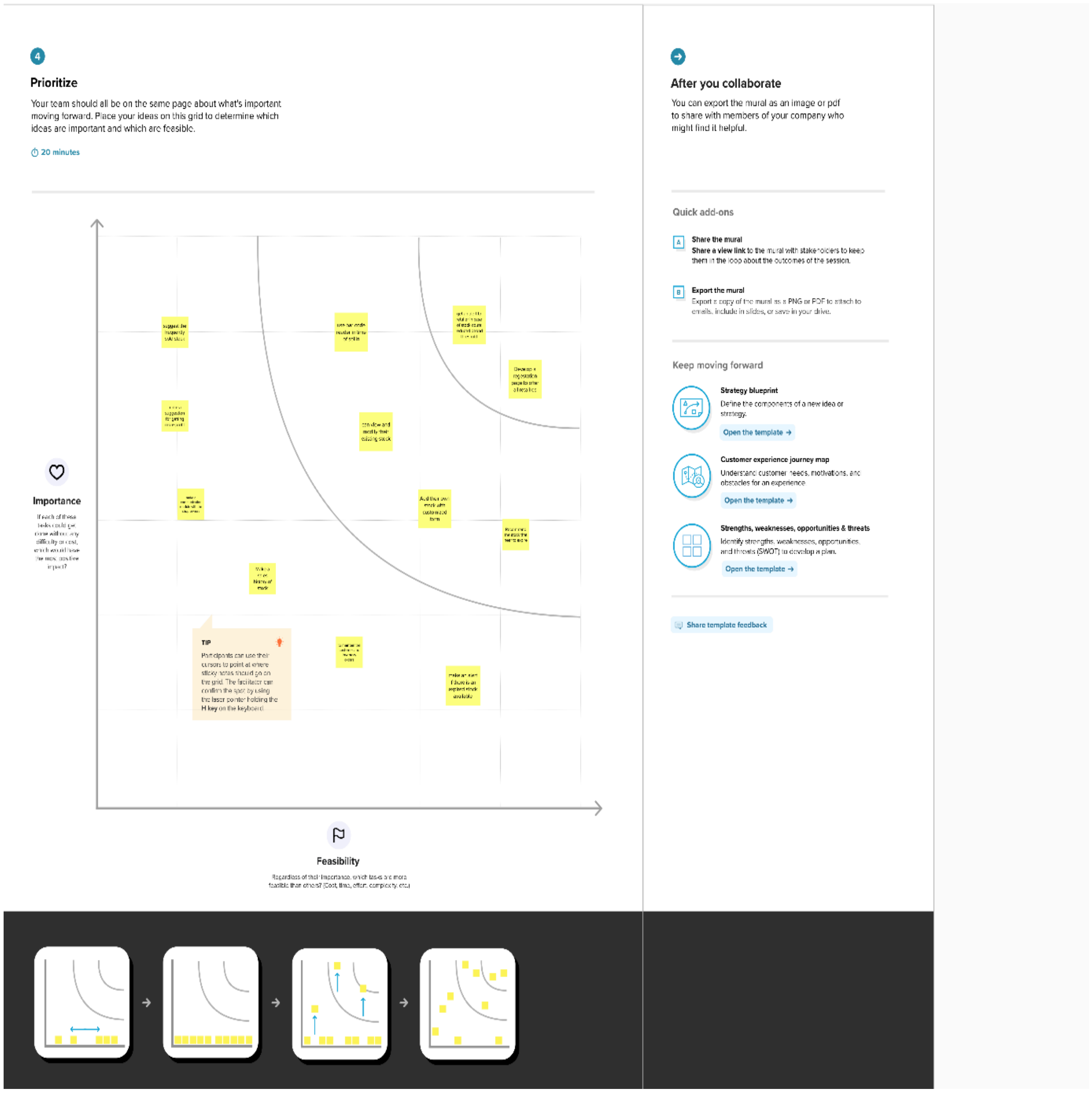
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Retail inventory management is the process managing inventory, retailers meet customer demand without running out of stock or carrying excess supply. The System will automatically send an email alert to the retailers if there is no stock found in their accounts
2.	Idea / Solution description	It enables retailers to access their inventory data from virtually any location & avoids risk of human error, saves you endless hours, and ensures you don’t make mistakes.
3.	Novelty / Uniqueness	The System will automatically send an email alert to the retailers if there is no stock found in their accounts. So that they can order new stock.
4.	Social Impact / Customer Satisfaction	The enhanced processes can help e-Commerce and online retailer build a strong repertoire with consumers.
5.	Business Model (Revenue Model)	Detailed inventory management mitigates these issues, allowing warehouse managers to refresh inventory only when needed. It’s both space and cost-effective.
6.	Scalability of the Solution	This proposed model is a cloud-based web application which can be accessed through browser. This might include expanding into new geographies, building an online presence, expanding new products and partners, or creating stronger alignments with suppliers and customers

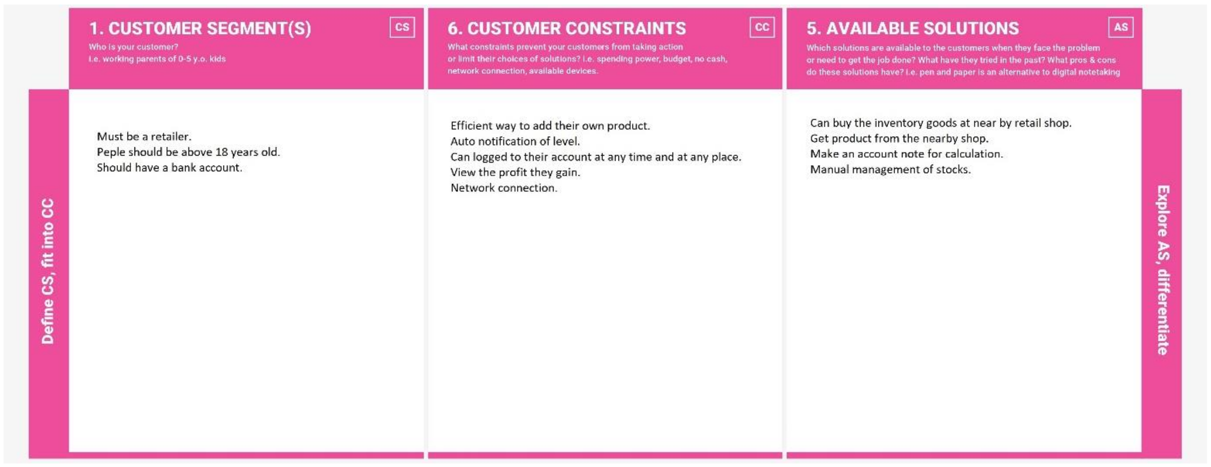
3.4. Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

Purpose:

- ❑ Solve complex problems in a way that fits the state of your customers.
- ❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- ❑ Sharpen your communication and marketing strategy with the right triggers and messaging.
- ❑ Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- ❑ **Understand the existing situation in order to improve it for your target group.**

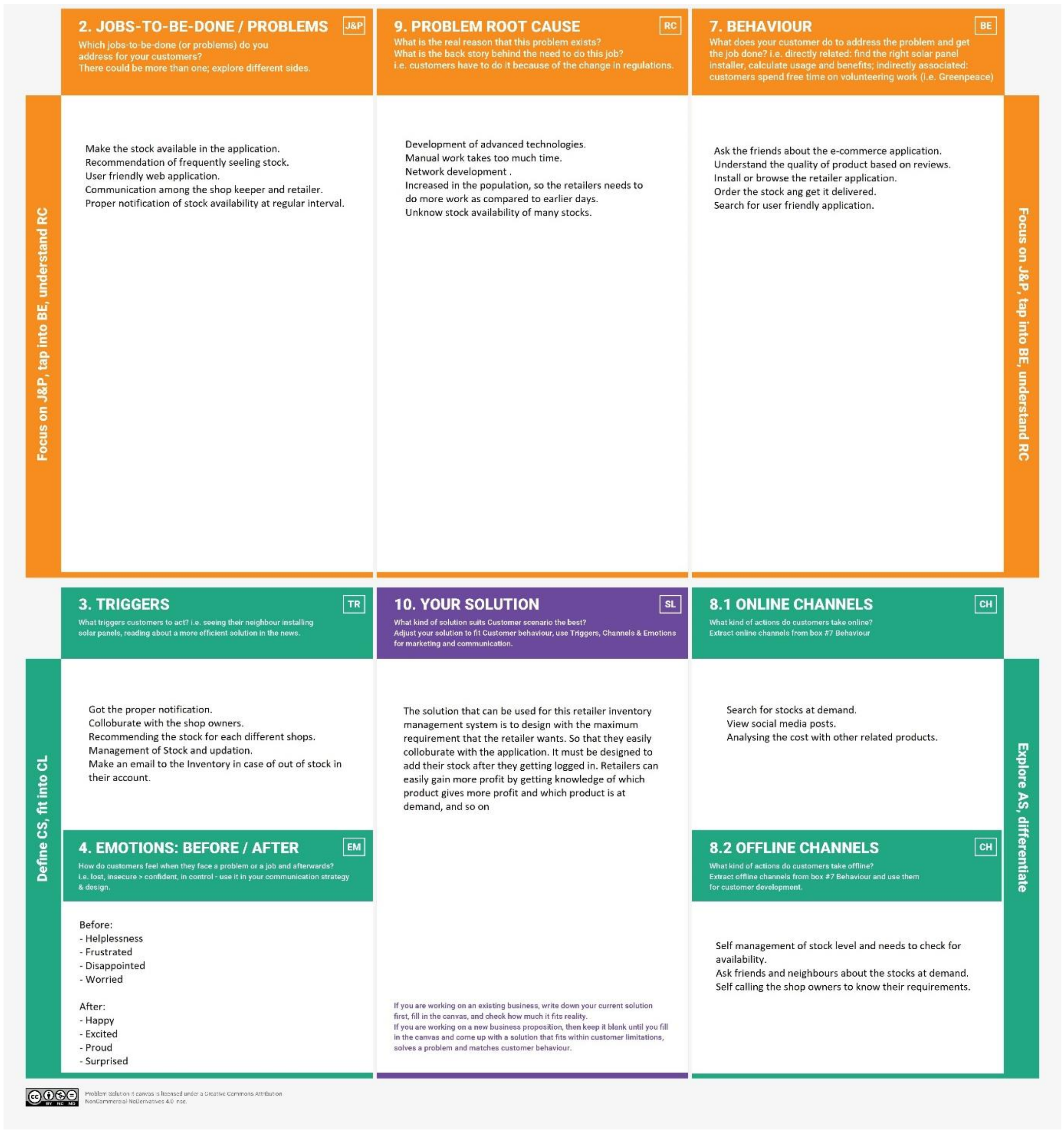
Template:



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS	6. CUSTOMER CONSTRAINTS CC	5. AVAILABLE SOLUTIONS AS
Who is your customer? i.e. working parents of 0-5 y.o. kids	What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.	Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking
Must be a retailer. Peple should be above 18 years old. Should have a bank account.	Efficient way to add their own product. Auto notification of level. Can logged to their account at any time and at any place. View the profit they gain. Network connection.	Can buy the inventory goods at near by retail shop. Get product from the nearby shop. Make an account note for calculation. Manual management of stocks.

Explore AS, differentiate



2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?
There could be more than one; explore different sides.

Make the stock available in the application.
Recommendation of frequently seeling stock.
User friendly web application.
Communication among the shop keeper and retailer.
Proper notification of stock availability at regular interval.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Development of advanced technologies.
Manual work takes too much time.
Network development .
Increased in the population, so the retailers needs to do more work as compared to earlier days.
Unknow stock availability of many stocks.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Ask the friends about the e-commerce application.
Understand the quality of product based on reviews.
Install or browse the retailer application.
Order the stock ang get it delivered.
Search for user friendly application.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Got the proper notification.
Colloburate with the shop owners.
Recommending the stock for each different shops.
Management of Stock and updation.
Make an email to the Inventory in case of out of stock in their account.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before:
- Helplessness
- Frustrated
- Disappointed
- Worried

After:
- Happy
- Excited
- Proud
- Surprised

10. YOUR SOLUTION SL

What kind of solution suits Customer scenario the best?
Adjust your solution to fit Customer behaviour, use Triggers, Channels & Emotions for marketing and communication.

The solution that can be used for this retailer inventory management system is to design with the maximum requirement that the retailer wants. So that they easily colloburate with the application. It must be designed to add their stock after they getting logged in. Retailers can easily gain more profit by getting knowledge of which product gives more profit and which product is at demand, and so on

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

8.1 ONLINE CHANNELS CH

What kind of actions do customers take online?
Extract online channels from box #7 Behaviour

Search for stocks at demand.
View social media posts.
Analysing the cost with other related products.

8.2 OFFLINE CHANNELS CH

What kind of actions do customers take offline?
Extract offline channels from box #7 Behaviour and use them for customer development.

Self management of stock level and needs to check for availability.
Ask friends and neighbours about the stocks at demand.
Self calling the shop owners to know their requirements.

Define CS, fit into CL

Explore AS, differentiate

Problem-Solution fit canvas is licensed under a Creative Commons Attribution NonCommercial NoDerivatives 4.0 Inter.

4. REQUIREMENT ANALYSIS

4.1. Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Username and Password
FR-2	User Confirmation	Confirmation via Phone number / E-mail address Confirmation via OTP
FR-3	Sign In	Retailers can log in to the application by their registered Username and Password.
FR-4	Dashboard	Retailers can view their stock details.
FR-5	Ordering	Order required products by putting them in a cart first.
FR-6	Restocking	Ordering more products when the stock is low.

4.2. Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Designing/Developing the site to be having alearning curve. Having user friendly website for users. Attractive interface for web-page. Making the web page responsive for desktops and mobile users.
NFR-2	**Security**	The security should be strong as to the attackerswon't be penetrating to the authorized users account or data. Log in system has the 2-step verification for their ID. They can log in to their id by receiving OTP for their registered Mobile or E-mail. Cookies based security system for authenticationand improved visiting experience on the site for clients.

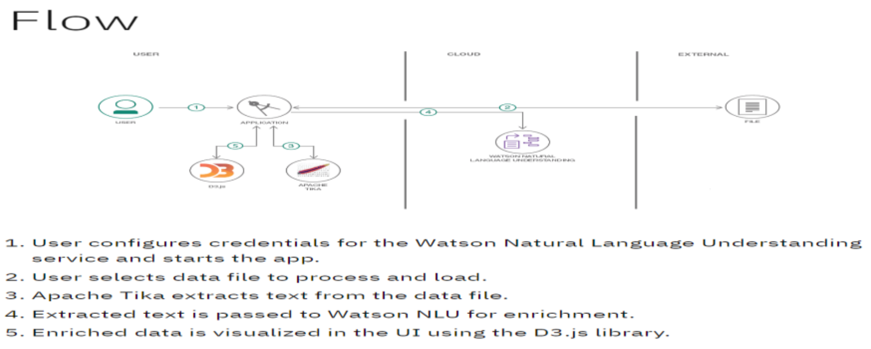
NFR-3	**Reliability**	The software having the capacity to handle sufficient numbers of users. And there is no lagging or discomfort feeling when browsing about stock when the web-page is busy. Should have minimum errors while executing theprograms. Should be available even at the times of calamity.
NFR-4	**Performance**	The response time for the software is low while operating in system. The convenience of this is it reduces the time periodof searching in aisle, searching for desired product, etc. It reduces costs, saves time, restocking period andpredicts the top selling products.
NFR-5	**Availability**	This uses IBM DB2 to ensure high availability of database servers and performances.
NFR-6	**Scalability**	As DB2 is highly scalable, the coding can be produced and developed efficiently and newfeatures can be introduced easily. Reusing the code can be done to add any new features. IBM Container in Docker registry is used which ishighly scalable.

5. PROJECT DESIGN

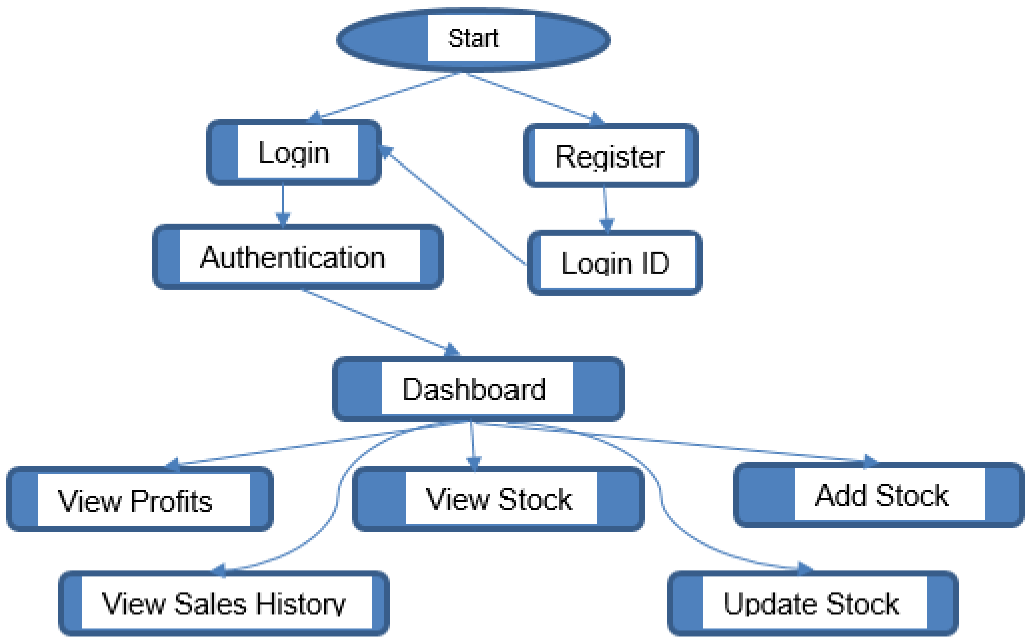
5.1. Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enter and leaves the system, what changes the information, and where data is stored.

Example: (Simplified)



1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.



5.2. Solution & Technical Architecture

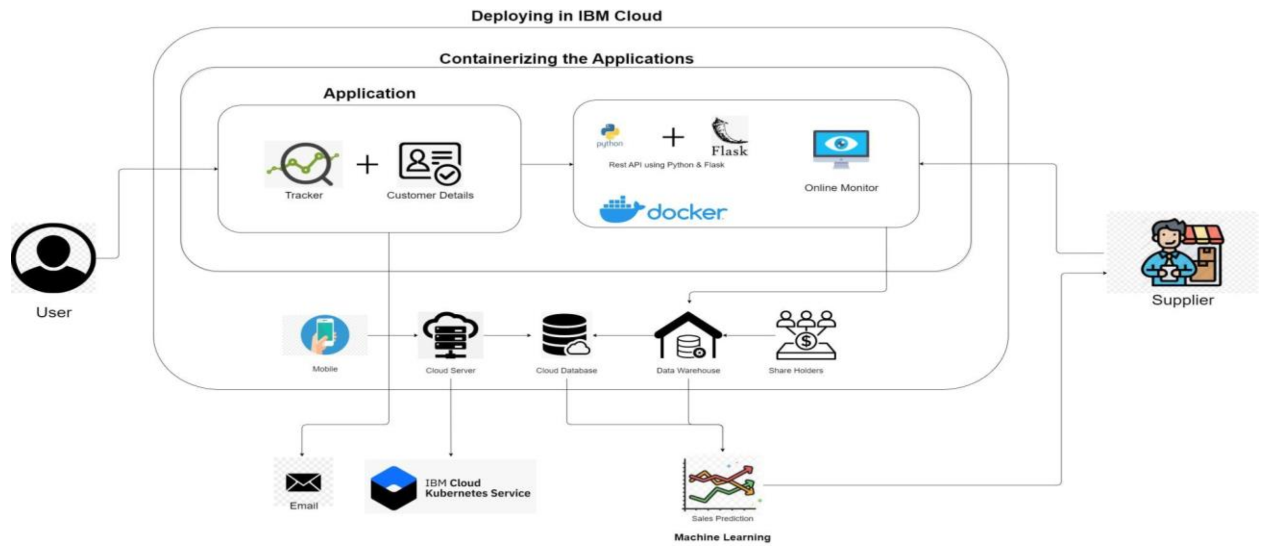


Table-1 : Components & Technologies:

S.No	**Component**	**Description**	**Technology**
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot, etc.	HTML, CSS, JavaScript, IBM CloudObject Storage, Python-Flask, Kubernetes, Docker, IBM DB2, IBM Container Registry.
2.	Application Logic	The logic for a process in the application	Python-Flask.
3.	Database	Data Type Configuration etc.	MySQL, etc.
4.	ChatBox	Chatbox for users to access help from a virtualassistant on the application.	IBM Watson Assistant
5.	Cloud Database	Database Service on Cloud	IBM DB2
6.	File Storage	File storage requirements	IBM Cloud Object Storage
7.	App Container	Contain the whole application in a single container.	Docker Container, IBM Container Registry
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal Server Configuration: port 5000 Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes.
9.	Send Mail	To send emails when low stock is present in the inventory to retailers.	IBM SendGrid

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	We use HTML, CSS, Bootstrap and Flask as the open source for our application	HTML, CSS, JavaScript, Bootstrap,Python-Flask.
2.	Security Implementations	User log in and authentication are done to providesecure access to their account.	IBM Cloud Security, Cookies..

3.	Scalable Architecture	The system can be scalable easily by using these technologies as to optimize, improve and add new features, allocate sufficient bandwidth to allow more users at a time, etc.	Docker, Kubernetes Cluster
4.	Availability	System availability is high as we make sure the unwanted database access is minimized throughSQL and code optimization.	IBM Db2, IBM Container Registry
5.	Performance	Deployment is easy and fast by containerizing theapplication. Providing fast access time and responsiveness bydeploying the application in cloud.	Flask, Docker, IBM Db2.

5.3. User Stories

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Retailer	Registration	USN-1	As a user, I can register for the application byentering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Retailer		USN-2	As a user, I can register through my E-mail	I can access my account / dashboard	Medium	Sprint-1
Retailer	Login ID	USN-3	As a user, I got the login ID.	I got the login ID so that I can access my account	High	Sprint-1
Retailer	Login	USN-4	As a user, I can log in to the authorized account by entering the Login ID and password	I can login with login ID provided.	Medium	Sprint-1
Retailer	Dashboard	USN-5	As a user, I can view the products that are available currently.	Inventory can be viewed once logged in.	High	Sprint-2

Retailer	Add Stock	USN-6	As a user, I can add my own stock which are needed for my inventory.	I can build my inventory	Medium	Sprint-2
Retailer	Stock Management	USN-7	As a user, I can add or remove the stocks.	I can increase the available inventory stocks.	Medium	Sprint-3
Retailer	Get stock notification	USN-8	As a user, I get the stock available notification at a regular time interval	I can easy manage the stock availability in my inventory	Low	Sprint-3
Retailer	Manage the History of product.	USN-9	As a user, I need to know the sales history for my inventory	I can view the history of my inventory sales.	Low	Sprint-4
Retailer	Income management	USN-10	As a user, I need to know the profit and loss of my inventory	I can view my profit of every day	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	Manojkumar M Madhan T
Sprint-1		USN-2	As a user, I can register through my E-mail	1	Medium	Manojkumar M Madhan T

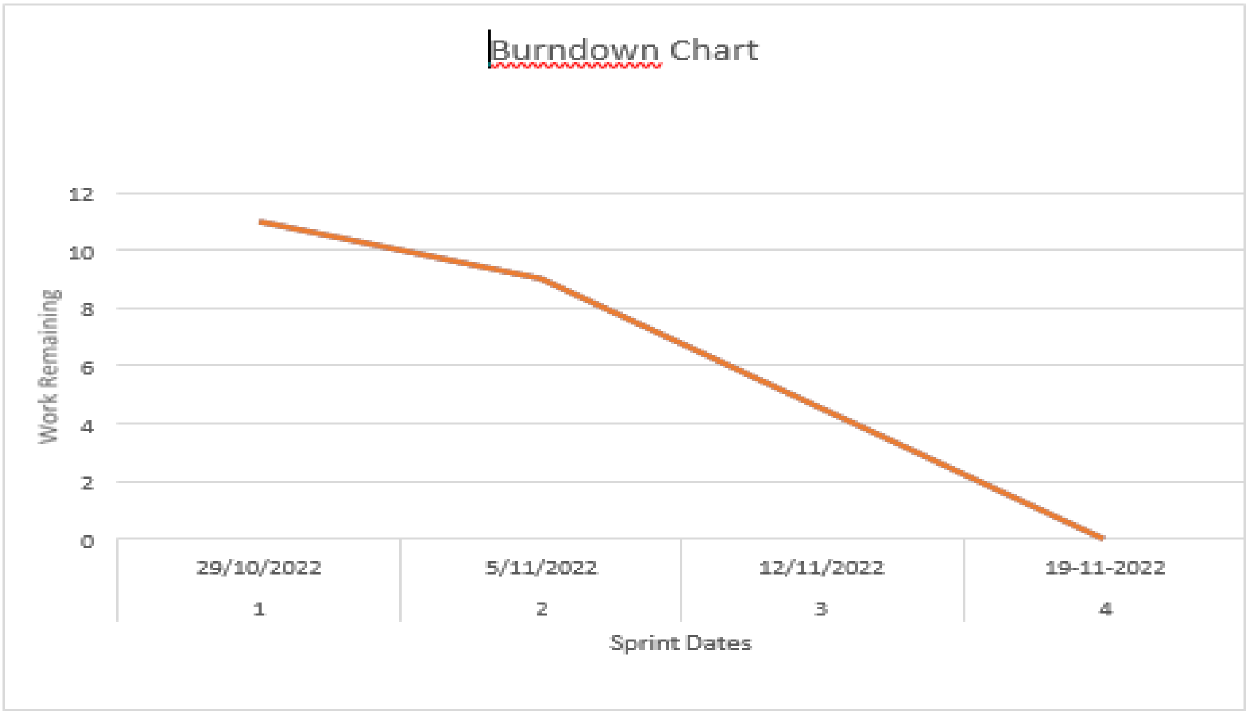
Sprint-1	Login ID	USN-3	As a user, I got the login ID.	3	High	Manojkumar M Madhan T
Sprint-1	Login	USN-4	As a user, I can log in to the authorized account by entering the Login ID and password	3	Medium	Manojkumar M Madhan T
Sprint-2	Dashboard	USN-5	As a user, I can view the products that areavailable currently.	2	High	Manojkumar M Bhuvanesh V Balasubramanian M
Sprint-2	Add Stock	USN-6	As a user, I can add my own stock which are needed for my inventory.	3	Medium	Manojkumar M Bhuvanesh V Balasubramanian M
Sprint-3	Stock Management	USN-7	As a user, I can add or remove the stocks.	2	Medium	Manojkumar M Madhan T Bhuvanesh V Balasubramanian M
Sprint-3	Get stock notification	USN-8	As a user, I get the stock available notification at a regular time interval	3	Low	Manojkumar M Madhan T Bhuvanesh V Balasubramanian M
Sprint-4	Manage the History of product.	USN-9	As a user, I need to know the sales history for my inventory	1	Low	Madhan T Bhuvanesh V Balasubramanian M
Sprint-4	Income management	USN-10	As a user, I need to know the profit and lose of my inventory	2	Medium	Madhan T Bhuvanesh V Balasubramanian M

6.2. Sprint Delivery Schedule

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	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	5	6 Days	31 Oct 2022	05 Nov 2022	5	05 Nov 2022
Sprint-3	5	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	3	6 Days	14 Nov 2022	19 Nov 2022	3	19 Nov 2022

6.3. Reports from JIRA

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1. Login

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet" href="\static\css\bootstrap.min.css" />
    <script src="\static\js\bootstrap.min.js"></script>
    <script src="\static\js\popper.min.js"></script>
</head>

<body>
    <div id="carouselId" class="carousel slide" data-bs-ride="carousel">
        <ol class="carousel-indicators">
            <li data-bs-target="#carouselId" data-bs-slide-to="0" class="active" aria-current="true"
                aria-label="First slide"></li>
            <li data-bs-target="#carouselId" data-bs-slide-to="1" aria-label="Second slide"></li>
            <li data-bs-target="#carouselId" data-bs-slide-to="2" aria-label="Third slide"></li>
            <li data-bs-target="#carouselId" data-bs-slide-to="3" aria-label="Fourth slide"></li>
            <li data-bs-target="#carouselId" data-bs-slide-to="4" aria-label="Fifth slide"></li>
            <li data-bs-target="#carouselId" data-bs-slide-to="5" aria-label="Sixth slide"></li>
        </ol>
        <div class="carousel-inner" role="listbox">
            <div class="carousel-item active">
                <img src="\static\images\in1.jpg" class="d-block" alt="First slide" height="250px"
width="100%">
            </div>
            <div class="carousel-item">
                <img src="\static\images\in2.jpg" class="d-block" alt="Second slide" height="250px"
width="100%">
            </div>
            <div class="carousel-item">
                <img src="\static\images\in3.jpg" class="d-block" alt="Third slide" height="250px"
width="100%">
            </div>
            <div class="carousel-item">
                <img src="\static\images\in4.jpg" class="d-block" alt="Fourth slide" height="250px"
width="100%">
            </div>
            <div class="carousel-item">
                <img src="\static\images\in5.jpg" class="d-block" alt="Fifth slide" height="250px"
width="100%">
```

```
            </div>
            <div class="carousel-item">
                <img    src="\static\images\in6.jpg"    class="d-block"    alt="Sixth    slide"    height="250px"
width="100%">
            </div>
        </div>
        <button    class="carousel-control-prev"    type="button"    data-bs-target="#carouselId"    data-bs-
slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Previous</span>
        </button>
        <button    class="carousel-control-next"    type="button"    data-bs-target="#carouselId"    data-bs-
slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Next</span>
        </button>
    </div>

    <!-- Nav tabs -->
    <ul class="nav nav-tabs d-flex justify-content-center" id="myTab" role="tablist" style="background-
color:gold;">
        <li class="nav-item" role="presentation">
            <button    class="nav-link    active"    id="login-tab"    data-bs-toggle="tab"    data-bs-target="#login"
type="button"
                role="tab" aria-controls="login" aria-selected="true" style="color:red;">Login</button>
        </li>
        <li class="nav-item" role="presentation">
            <button    class="nav-link"    id="register-tab"    data-bs-toggle="tab"    data-bs-target="#register"
type="button"
                role="tab" aria-controls="register" aria-selected="false" style="color:red;">Register</button>
        </li>
    </ul>

    <div class="row" style="max-width: 95vw;">
        <div class="col">
            <div class="row justify-content-center">
                <div class="col-md-6" style="width:25rem;height: fit-content;">
                    <div class="p-5 text-white bg-primary border rounded-3">
                        <center>
                            <h3>Customer Problems</h3>
                            <img
src="https://content.presentermedia.com/content/animsp/00010000/10579/figure_graph_top_earnings_300
_wht.gif "
                                style="width:70% ;background-color:beige;border-color: black;"
                                class="img-fluid rounded-top" alt="customerproblems"><br />
                            <br />
                            <h5>
                                <p>WE FULLFILL YOUR NEEDS !</p>
                            </h5>
```

```
          </div>
        </div>
      </div>
    </div>
    <div class="col">
      <!-- Tab panes -->
      <div class="tab-content">

        <div class="tab-pane active" id="login" role="tabpanel" aria-labelledby="login-tab">
          <form action="/Login1" method="post" class="form d-flex justify-content-center">
            <div class="border-3 mt-5 card text-start d-flex p-2 bd-highligh"
              style="width: 25rem;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0,
0, 0.19);">
              <div style="align-self:center;margin-top: -3em;"><img src="\static\images\login.png"
                  style="width:85px;height: 75px ;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px
20px 0 rgba(0, 0, 0, 0.19);border-radius: 75%;background-color:beige;border-color: black;"
                  class="img-fluid rounded-top" alt="customerproblems"></div>
              <div class="card-body">
                <label class="form-control-label" for="userid">User ID</label>
                <input type="text" class="form-control" id="userid" name="userid">
                <label class="form-control-label" for="passwd">Password</label>
                <input type="password" class="form-control" id="passwd" name="passwd">
                <div class="d-flex justify-content-center mt-2">
                  <button type="submit" class="btn btn-primary">Login</button>
                </div>
                <div class="d-flex justify-content-center">
                  <label>If you are new user try register</label>
                </div>
              </div>
            </div>
          </form>
        </div>
        <div class="tab-pane" id="register" role="tabpanel" aria-labelledby="register-tab">
          <form action="/Login" method="post" class="form d-flex justify-content-center">
            <div class="border-3 mt-5 card text-start d-flex p-2 bd-highligh"
              style="width: 25rem;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0,
0, 0.19)">
              <div style="align-self:center;margin-top: -3em;"><img
src="\static\images\login_img.png"
                  style="width:75px ;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0
rgba(0, 0, 0, 0.19);border-radius: 75%;background-color:beige;border-color: black;"
                  class="img-fluid rounded-top" alt="customerproblems"></div>
              <div class="card-body">
                <label class="form-control-label" for="name">Name</label>
                <input type="text" class="form-control" id="name" name="name">
                <label class="form-control-label" for="contact">Contact</label>
                <input type="number" class="form-control" id="contact" name="contact">
                <label class="form-control-label" for="agencyname">Agency Name</label>
                <input type="text" class="form-control" id="agencyname" name="agencyname">
                <label class="form-control-label" for="company">Company</label>
                <input type="text" class="form-control" id="company" name="company">
                <label class="form-control-label" for="mailid">Mail ID</label>
                <input type="email" class="form-control" id="mailid" name="mailid">
                <div class="d-flex justify-content-center mt-2">
                  <button type="submit" class="btn btn-primary">Register</button>
                </div>
                <div class="d-flex justify-content-center">
                  <label>If you are existing user try login</label>
                </div>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
</body>

</html>
```

7.2. Dashboard

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dashboard</title>
  <link rel="stylesheet" href="\static\css\bootstrap.min.css" />
  <script src="\static\js\bootstrap.min.js"></script>
  <script src="\static\js\popper.min.js"></script>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
</head>

<body>
  <div class="mt-5">
    <div id="carouselId" class="carousel slide" data-bs-ride="carousel">
      <ol class="carousel-indicators">
        <li data-bs-target="#carouselId" data-bs-slide-to="0" class="active" aria-current="true"
          aria-label="First slide"></li>
        <li data-bs-target="#carouselId" data-bs-slide-to="1" aria-label="Second slide"></li>
        <li data-bs-target="#carouselId" data-bs-slide-to="2" aria-label="Third slide"></li>
        <li data-bs-target="#carouselId" data-bs-slide-to="3" aria-label="Fourth slide"></li>
        <li data-bs-target="#carouselId" data-bs-slide-to="4" aria-label="Fifth slide"></li>
        <li data-bs-target="#carouselId" data-bs-slide-to="5" aria-label="Sixth slide"></li>
      </ol>
      <div class="carousel-inner" role="listbox">
        <div class="carousel-item active">
          <img src="\static\images\in1.jpg" class="d-block" alt="First slide" height="250px"
width="100%">
        </div>
        <div class="carousel-item">
          <img src="\static\images\in2.jpg" class="d-block" alt="Second slide" height="250px"
width="100%">
        </div>
        <div class="carousel-item">
          <img src="\static\images\in3.jpg" class="d-block" alt="Third slide" height="250px"
width="100%">
        </div>
        <div class="carousel-item">
          <img src="\static\images\in4.jpg" class="d-block" alt="Fourth slide" height="250px"
width="100%">
        </div>
        <div class="carousel-item">
          <img src="\static\images\in5.jpg" class="d-block" alt="Fifth slide" height="250px"
width="100%">
        </div>
        <div class="carousel-item">
          <img src="\static\images\in6.jpg" class="d-block" alt="Sixth slide" height="250px"
width="100%">
        </div>
      </div>
      <button class="carousel-control-prev" type="button" data-bs-target="#carouselId" data-bs-
slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Previous</span>
      </button>
      <button class="carousel-control-next" type="button" data-bs-target="#carouselId" data-bs-
slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Next</span>
      </button>
    </div>
  </div>
  <nav class="navbar navbar-expand-sm navbar-light fixed-top" style="background-color:aqua;">
    <div class="container">
      <a class="navbar-brand" href="Dashboard"><i class="fa fa-bars"></i> Dashboard</a>
      <button class="navbar-toggler d-lg-none" type="button" data-bs-toggle="collapse"
        data-bs-target="#collapsibleNavId" aria-controls="collapsibleNavId" aria-expanded="false"
        aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="collapsibleNavId">
        <ul class="navbar-nav me-auto mt-2 mt-lg-0">
          <li class="nav-item">
```

```
                    <form class="d-flex my-2 my-lg-0">
                        <button class="nav-link active border-0"
                            style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                class="fa fa-home"></i>Sales </button>
                    </form>
                </li>
                <li class="nav-item">
                    <form action="Overview" method="post" class="d-flex my-2 my-lg-0">
                        <button class="nav-link active border-0"
                            style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                class="fa fa-home"></i>Overview</button>
                    </form>
                </li>
                <li class="nav-item">
                    <form action="ManageStock" method="post" class="d-flex my-2 my-lg-0">
                        <button class="nav-link active border-0"
                            style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                class="fa fa-home"></i>Manage Stock </button>
                    </form>
                </li>
            </ul>
            <form class="d-flex my-2 my-lg-0">
                <button class="btn btn-outline-success my-2 my-sm-0" style="right:5px;width:fit-content;"
                    type="submit">Logout</button>
            </form>
        </div>
    </div>
</nav>

<div class="row d-flex justify-content-center" style="padding:5%;width: 95vw;">
    <div class="col-md-6" style="width: 21rem;height: fit-content;">
        <a href="ManageStock" style="text-decoration:none ;">
            <div class="p-5 text-white bg-primary border rounded-3">
                <h3>Make sale</h3>
                <img src="\static\images\i3.jpg" class="img-fluid rounded-top" alt="customerproblems">
                <br />
                <p>Sell your stock and make profit</p>
            </div>
        </a>
    </div>
    <div class="col-md-6" style="width: 25rem;height: fit-content;">
        <a href="Overview" style="text-decoration:none ;">
            <div class="p-5 text-white bg-primary border rounded-3">
                <h3>Product overview</h3>
                <img
src="http://content.presentermedia.com/files/animsp/00004000/4298/business_binder_pa_md_wm.gif"
                    class="img-fluid rounded-top" alt="customerproblems">
                <br />
                <p>Have a view on your Inverntory</p>
                </div>
            </a>
        </div>
        <div class="col-md-6" style="width: 21rem;height: fit-content;">
            <a href="ManageStock" style="text-decoration:none ;">
                <div class="p-5 text-white bg-primary border rounded-3">
                    <h3>Manage Stock</h3>
                    <img src="\static\images\p1.jpg" class="img-fluid rounded-top" alt="customerproblems">
                    <br />
                    <p>Add your own stocks for your Inverntory</p>
                </div>
            </a>
        </div>
        <div class="col-md-6" style="width: 25rem;height: fit-content;">
            <div class="p-5 text-white bg-primary border rounded-3">
                <h3>Sales History</h3>
                <img src="\static\images\Customer Problems.jpg" class="img-fluid rounded-top"
alt="customerproblems">
                <br />
                <p>We have full filled these problems</p>
            </div>
        </div>
        <div class="col-md-6" style="width: 25rem;height: fit-content;">
            <div class="p-5 text-white bg-primary border rounded-3">
                <h3>Daily Marketting Report</h3>
                <img src="https://www.hdfcsec.com/hsl.images/ELSS%20GIF%20final-
202203241627287702685-202207281608037620770.gif"
                    class="img-fluid rounded-top" alt="customerproblems123">
                <br />
                <p>We have full filled these problems</p>
            </div>
        </div>
    </div>
</body>
</html>
```

7.3. Product Overview

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Overview</title>
    <link rel="stylesheet" href="\static\css\bootstrap.min.css" />
    <script src="\static\js\bootstrap.min.js"></script>
    <script src="\static\js\popper.min.js"></script>
    <link        rel="stylesheet"        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
</head>

<body>
    <nav class="navbar navbar-expand-sm navbar-light fixed-top" style="background-color:aqua;">
        <div class="container">
            <a class="navbar-brand" href="Dashboard"><i class="fa fa-bars"></i> Dashboard</a>
            <button class="navbar-toggler d-lg-none" type="button" data-bs-toggle="collapse"
                data-bs-target="#collapsibleNavId" aria-controls="collapsibleNavId" aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="collapsibleNavId">
                <ul class="navbar-nav me-auto mt-2 mt-lg-0">
                    <li class="nav-item">
                        <form class="d-flex my-2 my-lg-0">
                            <button class="nav-link active border-0"
                                style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                    class="fa fa-home"></i>Sales </button>
                        </form>
                    </li>
                    <li class="nav-item">
                        <form action="Overview" method="post" class="d-flex my-2 my-lg-0">
                            <button class="nav-link active border-0"
                                style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                    class="fa fa-home"></i>Overview</button>
                        </form>
                    </li>
                    <li class="nav-item">
                        <form action="ManageStock" method="post" class="d-flex my-2 my-lg-0">
                            <button class="nav-link active border-0"
                                style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                    class="fa fa-home"></i>Manage Stock </button>
                        </form>
                    </li>
                </ul>
                <form class="d-flex my-2 my-lg-0">
                    <button class="btn btn-outline-success my-2 my-sm-0" style="right:5px;width:fit-content;"
                        type="submit">Logout</button>
                </form>
            </div>
        </div>
    </nav>
    <div class="row d-flex justify-content-center " style="padding:5%;width: 95vw;">
        {% for stock,available in stocks %}
        <div class="card mb-3 border-3"
            style="max-width: 540px;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0,
0.19);">
```

```
            <div class="row g-0">
                <div class="col-md-4">
                    <img src="https://s3.jp-tok.cloud-object-storage.appdomain.cloud/imsfr/{{stock[5]}}"
                        class="img-fluid rounded-start" alt="Card title">
                </div>
                <div class="col-md-8">
                    <div class="card-body">
                        <div style="float:right;">
                            {% if available == 1 %}
                            <div style="width:2vw;height:3vh;background-color: green; border-radius:
50px;"></div>
                            {% elif available == 0 %}
                            <div style="width:2vw;height:3vh;background-color: orange; border-radius:
50px;"></div>
                            {% else %}
                            <div style="width:2vw;height:3vh;background-color: red; border-radius: 50px;"></div>
                            {% endif %}
                        </div>
                        <h5 class="card-title"><u><b>{{stock[0]}}</b> - {{stock[1]}}</u></h5>
                        MRP : Rs. {{stock[4]}}<br />Manufacturing Date : {{stock[2]}}<br />Expiry Date :
                        {{stock[3]}}<br /><small class="text-muted">Available : {{stock[7]}}</small>
                    </div>
                </div>
            </div>
        </div>
        {% endfor %}
    </div>
</body>
</html>
```

7.4. Manage Stock

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Manage Stock</title>
    <link rel="stylesheet" href="\static\css\bootstrap.min.css" />
    <script src="\static\js\bootstrap.min.js"></script>
    <script src="\static\js\popper.min.js"></script>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
</head>

<body>
    <nav class="navbar navbar-expand-sm navbar-light fixed-top" style="background-color:aqua;">
        <div class="container">
        <a class="navbar-brand" href="Dashboard"><i class="fa fa-bars"></i> Dashboard</a>
        <button class="navbar-toggler d-lg-none" type="button" data-bs-toggle="collapse"
            data-bs-target="#collapsibleNavId" aria-controls="collapsibleNavId" aria-expanded="false"
            aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="collapsibleNavId">
            <ul class="navbar-nav me-auto mt-2 mt-lg-0">
                <li class="nav-item">
                    <form class="d-flex my-2 my-lg-0">
                        <button class="nav-link active border-0"
                            style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                class="fa fa-home"></i>Sales </button>
                    </form>
                </li>
                <li class="nav-item">
                    <form action="Overview" method="post" class="d-flex my-2 my-lg-0">
                        <button class="nav-link active border-0"
                            style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                class="fa fa-home"></i>Overview</button>
                    </form>
                </li>
                <li class="nav-item">
                    <form action="ManageStock" method="post" class="d-flex my-2 my-lg-0">
                        <button class="nav-link active border-0"
                            style="right:5px;width:fit-content;background-color: transparent;" type="submit"><i
                                class="fa fa-home"></i>Manage Stock </button>
                    </form>
                </li>
            </ul>
            <form class="d-flex my-2 my-lg-0">
                <button class="btn btn-outline-success my-2 my-sm-0" style="right:5px;width:fit-content;"
                    type="submit">Logout</button>
            </form>
        </div>
    </div>
</nav>
<div class="row" style="margin-top: 5%;">
    <div class="col">
        <a class="btn btn-primary" href="#addStockDivTitle">Add</a>
    </div>
    <div class="col">
        <a class="btn btn-primary" href="#updateStockDivTitle">Update</a>
    </div>
    <div class="col">
        <a class="btn btn-primary" href="#deleteStockDivTitle">Delete</a>
    </div>
</div>
<div class="row d-flex justify-content-center " style="padding:5%;width: 95vw;">
```

```
        {% for stock,available in stocks %}
        <div class="card mb-3 border-3"
            style="max-width: 540px;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0,
0.19);">
            <div class="row g-0">
                <div class="col-md-4">
                    <img src="https://s3.jp-tok.cloud-object-storage.appdomain.cloud/imsfr/{{stock[5]}}"
                        class="img-fluid rounded-start" alt="Card title">
                </div>
                <div class="col-md-8">
                    <div class="card-body">
                        <div style="float:right;">
                            {% if available == 1 %}
                            <div style="width:2vw;height:3vh;background-color: green; border-radius:
50px;"></div>
                            {% elif available == 0 %}
                            <div style="width:2vw;height:3vh;background-color: orange; border-radius:
50px;"></div>
                            {% else %}
                            <div style="width:2vw;height:3vh;background-color: red; border-radius: 50px;"></div>
                            {% endif %}
                        </div>
                        <h5 class="card-title"><u><b>{{stock[0]}}</b> - {{stock[1]}}</u></h5>
                        MRP : Rs. {{stock[4]}}<br />Manufacturing Date : {{stock[2]}}<br />Expiry Date :
                        {{stock[3]}}<br /><small class="text-muted">Available : {{stock[7]}}</small>
                    </div>
                </div>
            </div>
        </div>
        {% endfor %}
    </div>

    <h1 id="addStockDivTitle" style="text-align-last:center;color:brown;font-family:fantasy;"><i>ADD
THE STOCK</i></h1>
    <center>
        <div id="addStockDiv">
            <form action="addStock" method="post" enctype="multipart/form-data">
                <div class="border-3 mt-2 card text-start d-flex p-2 bd-highligh"
                    style="width: 75rem;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0,
0.19)">
                    <div class="card-body" style="background-color:darkkhaki;">
                        <label class="form-control-label" for="addproductName">Product Name</label>
                        <input type="text" class="form-control" id="addproductName" name="addproductName">
                        <div class="row">
                            <div class="col"><label class="form-control-label" for="manDate">Manufacturing
Date</label>
                                <input type="date" class="form-control" id="manDate" name="manDate">
                            </div>
                            <div class="col">
                                <label class="form-control-label" for="expDate">Expiry Date</label>
                                <input type="date" class="form-control" id="expDate" name="expDate">
                            </div>
                        </div>
                        <div class="row">
                            <div class="col"><label class="form-control-label" for="mrp">MRP</label>
                                <input type="number" class="form-control" id="mrp" name="mrp">
                            </div>
                            <div class="col">
                                <label class="form-control-label" for="qty">Quantity</label>
                                <input type="number" class="form-control" id="qty" name="qty">
                            </div>
                        </div>

                        <label class="form-control-label" for="addProductImage">Product</label>
                        <input          type="file"          class="form-control"          id="addProductImage"
name="addProductImage">
                        <div class="d-flex justify-content-center mt-2">
                            <button type="submit" class="btn btn-primary">Add Stock</button>
                        </div>
                        <div class="d-flex justify-content-center">
                            <label>Make sure you have entered correct details</label>
                        </div>
                    </div>
                </div>

            </form>
        </div>
    </center>

    <br /><br /><br />
    <h1          id="updateStockDivTitle"          style="text-align-last:center;color:brown;font-
    family:fantasy;"><i>UPDATE THE STOCK</i>
    </h1>
    <center>
        <div id="updateStockDiv">
            <form action="updateStock" method="post">
                <div class="border-3 mt-2 card text-start d-flex p-2 bd-highligh"
                    style="width: 75rem;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0,
0.19)">
                    <div class="card-body" style="background-color:darkgoldenrod;">
                        <div class="row">
                            <div class="col"><label class="form-control-label" for="pid">Product ID</label>
                                <input type="number" class="form-control" id="pid" name="pid" {% if getStock %}
                                    value="{{getStock[0]}}" {% endif %}>
                            </div>
                            <div class="col">
                                <input type="submit" class="btn" style="background-color:aqua;margin-top:21px;"
                                    name="submit" value="Get Stock Details">
                            </div>
                            <div class="col">
                                <img {% if getStock
                                    %}src="https://s3.jp-tok.cloud-object-
        storage.appdomain.cloud/imsfr/{{getStock[5]}}"
                                    {% elif not getStock %}
```

src="data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAkGBxISEBUSEBIVFRU
VFRUVFRUXFRUVFRUVFRUYFxUVFRUYHSggGBolGxUVITEhJSkrLi4uFx8zODMtNygtLisBCgo
KDg0OFxAQGS8lHx0tMi0tMi0vLSsvLS0tKzcrKy0tLS0tKy0tLS0tKy0tLS0tMisrLS0rNy0tLS0tLS0tK//
AABEIAM0A9gMBIgACEQEDEQH/xAAbAAEAAgMBAQAAAAAAAAAAAAAAAAQIDBAUGB//E
ADsQAAIBAgEIBgcIAgMAAAAAAAAABAgMRBAUSITFBUXGBBjJhkaGxIiNCssHR4RMzUmJyc4K
SJPFTY6L/xAAZAQEAAwEBAAAAAAAAAAAAAAAAAAQMEAgX/xAAjEQEAAgEDBAMBAQAA
AAAAAAAAAQIDBBExExEiEyQRNRcTNC/9oADAMBAAIRAxEAPwD7iAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABSpVjHrNLi0gLg0quVaS9q
/BN+Oo1KmXF7MG+LS8rlc5aR7dxjtPp2AeeqZYqvVmx4K78TWljKjd3OXfbwWgrnU19LI09vb1QPN
0srVVtT4r5WN2jlpe1B/x0+DOq56S5nDaHXBr0sZCWp24nrzNhMtiYnhXMTHTHIACUAAAAAAAAAA
AAAAAAAMOLxUaUHObGt2b221LichdK8M3aEm3uazH3Ts/A4teteZdVpa3EO6Dz9XL8vfZglxbflY
06uVK0vba4WXitJVOppHCyMgZfb/AM4NbHJ+TLS8E0dcs2dVCWxEJXunZRtdWWvjSZ2ugppjF
06uVK0vba4WXitJVOppHCyMFperlJLS3Y1auU6Udc0+HpeR5Sc29LbfFt+ZBVOqn1C2NNHuXoKmX
YLqxk+NkjVq5bqPqqMe9s5S7DLGhJ7O8rnNktw8w7+KlWWpj6stc5cnbyNe5k+xW2S4KFKP4Z
c7Lw1+BHTaeXXVWOVvctGm3qTNhyS1Zq5kSrr8z528h0xHMnVqqnUtQyXffMiJ7TD9zvEgz
+33JIrKo3tZG9INrS2c2K1pc2FXitXgrGoB8n1B0fbZlityIji5rVJrga4I67faeiG/SypVW1PivlY26WW/xw5
p/BnGuGzuMt49onFWfT0tHKdKXtJPdLR4vQbaZ4qVZFsPjpqUVBuN5x26H6SurauzmW11X3Cq2m+p
e0ABsZAAAAAAAAAAAaGXo3w1b9uT7lf4HzaUU1ZmPnbnpPp2VxXFnzMwazyhs03U3EqQU
o/dzlHsvnR/rLVysbbFPKVWPXhGa3webLjWsyRLQ7GHxVOTSu03sas/qb8FBbuf1POumno2bth
hecn6E5R4O6/rK67i2LRHpxMb+3qnXS29xjlWWxX4X4tvweo89TylVj/ZG5QyvSlZOW
ZJ+zNZjb3JvQ+TZPyTJ0Q6kq8uBRyb1sgg5mZl1tCwIuUlWitpCWQk1ZYxIxyxj2EbwnZvEOa3nNdaT
2kNkdRs35YmJjeL3I1CyI6pTsyyrye0xuQCI3SsZ8CvW0/wByHvowGzk1eup/rh7yOq8B9YvEgA9l5IA
AAAAAAAAAMeIjeElvi13tEo+WRPQpqiaSjsMM2LIvrvvExmXMmbQa3jW06d1Pc/gpU0Mm
mrp7HqfTearq8HtxpxL+MeJNIHwrz4hXDb7hoEUwREUnfVGLYaWoYVAj2QcO9nMTo6zW5gl+K6yCUWyaV
2AkIEEpgSjayXG9an+ns/+uL7Jy9fT4Iy9f8Ze0AB7LygAAAAAAAAAAD5fjI2qzX7cn3Sa+B
PmeVo2xFf8Abr/y+Y9xZxDTpuZmbHkjhkbXx2DIjXqGhPMNrJ2i1yrWKNLJ6t5+G5MXPsADoqMR
j8cDJB/cEWWcGuWsVAC0FoUpSzLuR5u6yyyQJG3mnB7UA2T14NPqQ9T8TTA9bqLDu56uR8M2a
SPFlPoixLjkc9Bz2SsnjsWUNGXjYGXTvXBLIJHpwPWZr/Ln6ckZvYLZPQ/7+ZNq1gHrL3KTYppbEWmGfzsR+jl
B7mdvK8ZNbsH3PYy/wC/aaXb8/x3owpBA7Nuk0TL5vdXOV+Pld8e+bRzkSepP2PhqNBn4k/4N5W59SIK+4
Zc/Y8Ryzo4nH0e/MLwW4XSKXyn0PxU8dPhEWl9P90qzwDDXVU0QVPFSAEi/KqpmAyWMfdk1zc4ZUI0IKS
0KqCFYJn1hb/X4uvDnoS3RzVYJKCgRrTNNtZJvbaDDAzm6AqeAydJpsSrHVKDF0RaxvEmFyMV5bJaXK
XSF07fPlVp1aI0kBCdbJvppCdhmeMfgdvjhZpACZWmV+FZEv6dy9w5tU4Jh4MEGD5M4WOsu4Y7mIPn8
4Ns6KXXqgI7n3JD+M61XMsMuEqDtQvv5SHOGRDu6JXv9dpyK1RoG3J5vXfk1oCWYK9nzAcoIcBFYm+Vl
Ib6+VBnYDCc6+GYgSyFytbj1vyYvXjPeoKjw8aVu3VyP6xmsVNBRcMdsXsDgeHL5GnJl0qdTP9qYvPx
lL4VehuSW5pRnppMdrSo6VS9XG3JalBUKWOpgg6LS3BlO2WKZCH2gQRvU55BH3MgCXz+aAO08+Tzm
ePDz1UJo6TQVRnBLrR6m5TT6Q9uKs2YhmQKTdLhPcxE1fjOMuUWZgaRdSU3qe4UFAEq4Nek3mupzM
xDuCxm0wsGdL2fCcf9AWqkfOhSakUhZvR9ap2ZZfh2CIbLhHN11ntmbK7xdOJ8BCfX6Zu4qZp67NB
pf1X2UkkIHZM8yw6iFDDcbS8MUCrgwk8sGc3WUVhFuWQZCb5jPMDpgGCF5mFnu7qZeTyPbq1n+C4
A2o/wPB2ItoT+HZ+Hk1vkVKEjC5V2dE3TXTSpfZRZnR2ozr3H4F5RPJbemGL4KBLLx4IoxSu0s

3ju81ShKTtFOXD4vZzOphMh1Jdb0V3v5LxPV4fAQgrJI2owSLqaWkc91VtTaeOzi4LIEI6Wrve9L5buR
1qeHjHUjMDTERthe EbQomZnlCRIBKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAB//Z"

```
                  {% endif %} style="width:100px ;" class="img-fluid rounded-start" alt="Card
title">
              </div>
            </div>
            <label class="form-control-label" for="addproductName">Product Name</label>
            <input type="text" class="form-control" id="addproductName" name="addproductName"
{% if getStock
              %} value="{{getStock[1]}}" {% endif %}>
            <div class="row">
              <div class="col"><label class="form-control-label" for="manDate">Manufacturing
Date</label>
                <input type="date" class="form-control" id="manDate" name="manDate" {% if
getStock %}
                  value="{{getStock[2]}}" {% endif %}>
              </div>
              <div class="col">
                <label class="form-control-label" for="expDate">Expiry Date</label>
                <input type="date" class="form-control" id="expDate" name="expDate" {% if
getStock %}
                  value="{{getStock[3]}}" {% endif %}>
              </div>
            </div>
            <div class="row">
              <div class="col"><label class="form-control-label" for="mrp">MRP</label>
                <input type="number" class="form-control" id="mrp" name="mrp" {% if getStock
%}
                  value="{{getStock[4]}}" {% endif %}>
              </div>
              <div class="col">
                <label class="form-control-label" for="qty">Quantity</label>
                <input type="number" class="form-control" id="qty" name="qty" {% if getStock %}
                  value="{{getStock[7]}}" {% endif %}>
              </div>
            </div>
            <div class="d-flex justify-content-center mt-2">
              <input type="submit" class="btn" style="background-color:aqua;margin-top:21px;"
                name="submit" value="Update Stock">
            </div>
            <div class="d-flex justify-content-center">
              <label>Make sure you have entered correct details</label>
            </div>
          </div>
        </div>

      </form>
```

```
        </div>
    </center>

    <br /><br /><br />
    <h1                    id="deleteStockDivTitle"                    style="text-align-last:center;color:brown;font-
family:fantasy;"><i>REMOVE THE STOCK</i>
    </h1>
    <center>
        <div id="deleteStockDiv">
            <form action="deleteStock" method="post">
                <div class="border-3 mt-2 card text-start d-flex p-2 bd-highligh"
                    style="width: 75rem;box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0,
0.19)">
                    <div class="card-body" style="background-color:darksalmon;">
                        <div class="row">
                            <div class="col"><label class="form-control-label" for="pid">Product ID</label>
                                <input type="number" class="form-control" id="pid" name="pid" {% if getStock %}
                                    value="{{getStock[0]}}" {% endif %}>
                            </div>
                            <div class="col">
                                <input type="submit" class="btn" style="background-color:aqua;margin-top:21px;"
                                    name="submit" value="Get Stock Details">
                            </div>
                            <div class="col">
                                <img {% if getStock
                                    %}src="https://s3.jp-tok.cloud-object-
storage.appdomain.cloud/imsfr/{{getStock[5]}}"
                                    {% elif not getStock %}

src="data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAkGBxISEBUSEBIVFRU
VFRUVFRUXFRUVFRUVFRUYFxUVFRUYHSggGBolGxUVITEhJSkrLi4uFx8zODMtNygtLisBCgo
KDg0OFxAQGS8lHx0tMi0tMi0vLSsvLS0tKzcrKy0tLS0tKy0tLS0tKy0tLS0tMisrLS0rNy0tLS0tLS0tK//
AABEIAM0A9gMBIgACEQEDEQH/xAAbAAEAAgMBAQAAAAAAAAAAAAAAAAQIDBAUGB//E
ADsQAAIBAgEIBgcIAgMAAAAAAAAAABAgMRBAUSITFBUXGBBjJhkaGxIiNCssHR4RMzUmJyc4K
SJPFTY6L/xAAZAQEAAwEBAAAAAAAAAAAAAAAAAAQMEAgX/xAAjEQEAAgEDBAMBAQAA
AAAAAAAAAQIDBBExEiEyQRNRcTNC/9oADAMBAAIRAxEAPwD7iAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABSpVjHrNLi0gLg0quVaS9q
/BN+Oo1KmXF7MG+LS8rlc5aR7dxjtPp2AeeqZYqvVmx4K78TWljKjd3OXfbwWgrnU19LI09vb1QPN
0srVVtT4r5WN2jlpe1B/x0+DOq56S5nDaxWZCWp24przNhMtiYnhXMTHRIACUAAAAAAAAA
AAAAAAMOLxUaUHObtGOt2b221LichdK8M3aEm3uazH3Ts/A4teteZdVpa3EO6Dz9XL8vZGlxbflY
06uVK0vba4WXitJVOppHCyMFpHpR5Sc29LbfFt+ZBVOqn1C2NNHuXoKmX
YLqxk+NkjVq5Sq5bqPqqMe9s5S7DLGhJ7O8rnNkyk+NkjVWpj6stc5cnbyNe5k+xW2S4RTk+5GWFKP4Z
c7Lw1+BHTaeXXVWOGvctGm3qTNhyS1Zq5Z3yKSrr8z528h0xHMvnZX0tVvqjJ29w5ZEg
+33JIrKo3tZG9INrS2c2K1pc2FXitXgrGoB8n1B0fbZlityIji5rVJrga4I67faeiG/SypVW1PivlY26WW/xw5
p/BnGuGzuMt49onFWfT0tHKdKXtJPdLR4vQbaZ4qVZFsPjpqUBuN5x26H6SurauzmW11X3Cq2m+p
e0ABsZAAAAAAAAAAaGXo3w1b9uT4lf4HzaUU1ZpNbnpNbnpPp2VIXoFhbjuaYzlfc03EqQU
o/dzlHsvnR/rLVysbFPKVWPXXhGa3webLjmy0eJiJWsyRLQ7GHxVOTSu03sas/qb8FBbuf1POumno2bth
hecn6E5R4O6/rK67i2LRHpxMb+3qnXS29xjlWWxX4X4X4X6Q4bnrWPyTUqbtZi2rw7OQWsyRcsyn9J+J6dS0r
o/dz1HsvnR/rLVysbFPKVWPXXhGa3webLjmy0eJiJWsyRLQ7GHxVOTSu03sas/qb8FBbuf1POumno2bth
hecn6E5R4O6/rK67i2LRHpxMb+3qnXS29xjlWWxX4tvweo89TylVj1Vj1oRqLqLfF5kv6y0dbdv/ZG5QyvSlZ
ZJ+zNZb3TaTyEaGqk0eF4H9X1D7pnBUzpYW45udl9b9WlnmP5Suh1mZ2wXc6eDtllhSe2bC8e6hZ0vmbO7Su
ZJ+zNZb3TaTyEaGqk0eF4H9X1D7pnBUzpYW45udl9b9WlnmP5Suh1mZ2wXc6eDtllhSe2bC8e6hZ0vmbO7Su
```

2kNkdRs35YmJjeL3I1CyI6pTsyyrye0xuQCI3SsZ8CvW0/wByHvowGzk1eup/rh7yOq8wi3EvcgA9l5IA
AAAAAAAAAMeIjeElvi13o+WRPqzPlclZ23aDDrP8tem9oJiQSYmps2NeprZMajW0ictJ1ad0RCpEo3V
mrp7HqfIkWOEq4f1TvTearq8U3mNbfR1J9qOjLGNnLxPVZtnW6YZXVb2lbkIlEJSi1ipZASiSCwEkkE
2AklEEpgSjayXG9an+uL7jVN7Iy9fT4/BnWPyj9c38Ze0AB7LygAAAAAAAAAAD5fjI2qTW6cl3SZ9Q
PmeVo2xFVf8AbPxk2Y9ZxDTpuZapJAuee2JBCJAgEgDDiuq+DNtalwNXEdV8H5GxSfox4LyJTDIiyK
osglKJISJAsiSESgJJIJSIEokgsAOhkNevhxfus0EdLo+v8iH8vdZZi86/rjJ4T+PXoAHsPLAAAAAAAAA
AAZ856QRtiqv6r96T+J9GPn3SmNsXU7c1/wDiJk1nhH60abylygQDzm1IFwBIIsSBjr9V8GZ6PUj+leRg
rajNh+pHgvIlMMyZJVFglJJCLASiSCUQJLFSyAEoEgEdXo6vXrhLyOWdbo39/wDxl8PmW4fOrjL4S9
WAD13lgAAAAAAAAAAHgumEbYp9sYvzXwPeNnhumq/yIvfSj4Tn9DNq/wCa/T+bgggHmNywZFw
BYkqSEq1NRlw/UjwMc9Rkw3UjwJIZSyIRKCUolAkCUSQSQLEoqWQEkoqWQEnY6Mffv9uXvROO
dnowvWyf5H70S7B/SqvN4S9QAD1nmAAAAAAAAAArI8X03japTe+L8H9T2xxOkuRHiIpwlacFLN
T6sr20N2utS0+BTnrNqTELMVoreJl8/uTcYmjOnPMqRcJa82W1b09Ul2q6KJnlTEx2l6ETvwumSUuSQl
dMkqmTcJTLUZMN1I8/MxsyYZ+gufmSQzFkVRZBKUSiESiBJKIJQEkkACyJFKnKTtGLk+xX79x18
HkCctNR5q3LTL5LxLKYr34hxbJWvMuSmd/oxQkpSk4tRcbJtWvpvo36jp4PJFKnpUbv8UtL+nI37GzD
pprMWmWXLqItG0QkAGxlAAAAAAAAAAAGrlDJ9KvDMrQjOOuzWp7HF64vtWk8blofVp3lh
pfaR/45tKouyFR6JcJW7ZM94GV3xVvzDut7V4fIG7ScJJxlHrQknpGS7XF6baNep7Lkpn07KuR6OIjm1o
KVr5stMZxb2wmvSi+DPHZU6KVaSvRbrR3PNjUS8Iz8HxZhyaW1e9e7VTPE89nETLpmCM9LWlNO
0k01KL3Si9MXxMkWZWhlMuG6q5+ZhizPhurzfmEsqJRAQSsSVubFDCVJ9WLfbqXe9fK51Ws2naIR
NoryxC/03vgjt4To63pnLkvn/AKO5hMk04aopefN7TTTSWny7M9tTWOO7y2GyXVnqjmre/lr77HZwnR
6K0zvLjoXd87nehTS1FzVTT0r6ZrZ729tehhYxVkkluWhGdIkF6oAAAAAAAAAAAAAAAAAA
CGiQBycrZCo116cE2tUlonHhJabdmo8blPo3Wo3cPWx7ElUS7Y6pcrP8AKfSCk6aZVkw0vzCymS1eHy
alNO+9aGtTT3Na0+xm3Q6vN+Z7XKnR6nVec42lqU1okuy+1djujVwnReEeteXH5Ky6O2/aezTGprt
3ju81ShKTtFOXD4vZzOphMh1Jdb0V3v5LxPV4fAQgrJI2owSLqaWkc91VtTaeOzi4LIEI6Wrve9L5buR
1qeHjHUjMDTEREbQomZnlCRIBKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAB//Z"

```
                          {% endif %} style="width:100px ;" class="img-fluid rounded-start" alt="Card
title">
                    </div>
                 </div>
                 <label class="form-control-label" for="addproductName">Product Name</label>
                 <input type="text" class="form-control" id="addproductName" name="addproductName"
{% if getStock
                    %} value="{{getStock[1]}}" {% endif %}>
                 <div class="row">
                    <div class="col"><label class="form-control-label" for="manDate">Manufacturing
Date</label>
                       <input type="date" class="form-control" id="manDate" name="manDate" {% if
getStock %}
                          value="{{getStock[2]}}" {% endif %}>
                    </div>
                    <div class="col">
                       <label class="form-control-label" for="expDate">Expiry Date</label>
                       <input type="date" class="form-control" id="expDate" name="expDate" {% if
getStock %}
                          value="{{getStock[3]}}" {% endif %}>
                    </div>
                 </div>
                 <div class="row">
```

```
                        <div class="col"><label class="form-control-label" for="mrp">MRP</label>
                            <input  type="number"  class="form-control"  id="mrp"  name="mrp"  {%  if  getStock
%}
                                value="{{getStock[4]}}" {% endif %}>
                        </div>
                        <div class="col">
                            <label class="form-control-label" for="qty">Quantity</label>
                            <input type="number" class="form-control" id="qty" name="qty" {% if getStock %}
                                value="{{getStock[7]}}" {% endif %}>
                        </div>
                    </div>
                    <div class="d-flex justify-content-center mt-2">
                        <input type="submit" class="btn" style="background-color:aqua;margin-top:21px;"
                            name="submit" value="Delete Stock">
                    </div>
                    <div class="d-flex justify-content-center">
                        <label>Make sure you have choosen correct stock</label>
                    </div>
                </div>
            </div>

        </form>
    </div>
  </center>
</body>
</html>
```

7.5. Database Schema

Retailer Table

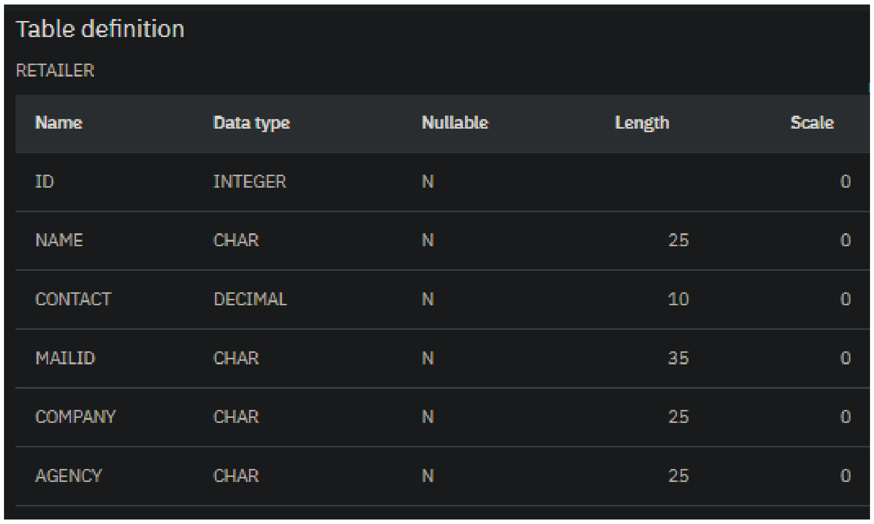


Table definition

RETAILER

Name	Data type	Nullable	Length	Scale
ID	INTEGER	N		0
NAME	CHAR	N	25	0
CONTACT	DECIMAL	N	10	0
MAILID	CHAR	N	35	0
COMPANY	CHAR	N	25	0
AGENCY	CHAR	N	25	0

Stock Table

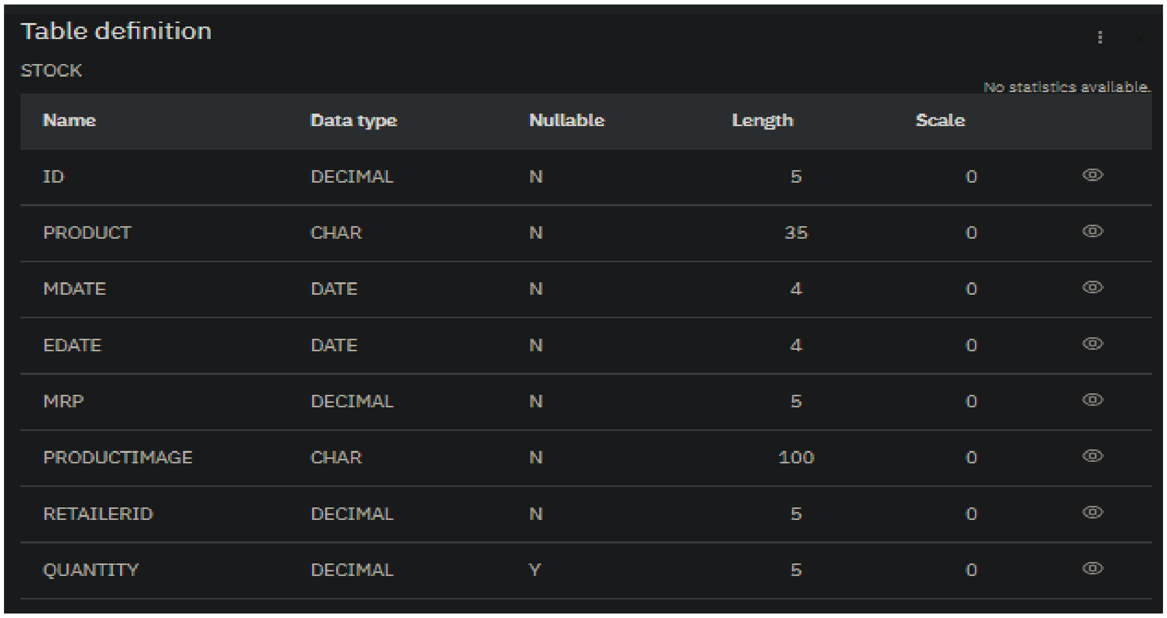


Table definition

STOCK

No statistics available.

Name	Data type	Nullable	Length	Scale
ID	DECIMAL	N	5	0
PRODUCT	CHAR	N	35	0
MDATE	DATE	N	4	0
EDATE	DATE	N	4	0
MRP	DECIMAL	N	5	0
PRODUCTIMAGE	CHAR	N	100	0
RETAILERID	DECIMAL	N	5	0
QUANTITY	DECIMAL	Y	5	0

8. TESTING

8.1. Test Cases

Register

Sl. No.	**Input**	**Output**	**Status**
1	Mariyappan A 9976627039 Sindhu Agecies Susee Trades marimanoj12@gmail.com	Login ID : 1001 Registration Successful	Success
2	Manojkumar M 7358734464 Sindhu Agecies Susee Trades marimanoj12@gmail.com	Login ID : 1002 Registration Successful	Success

Login

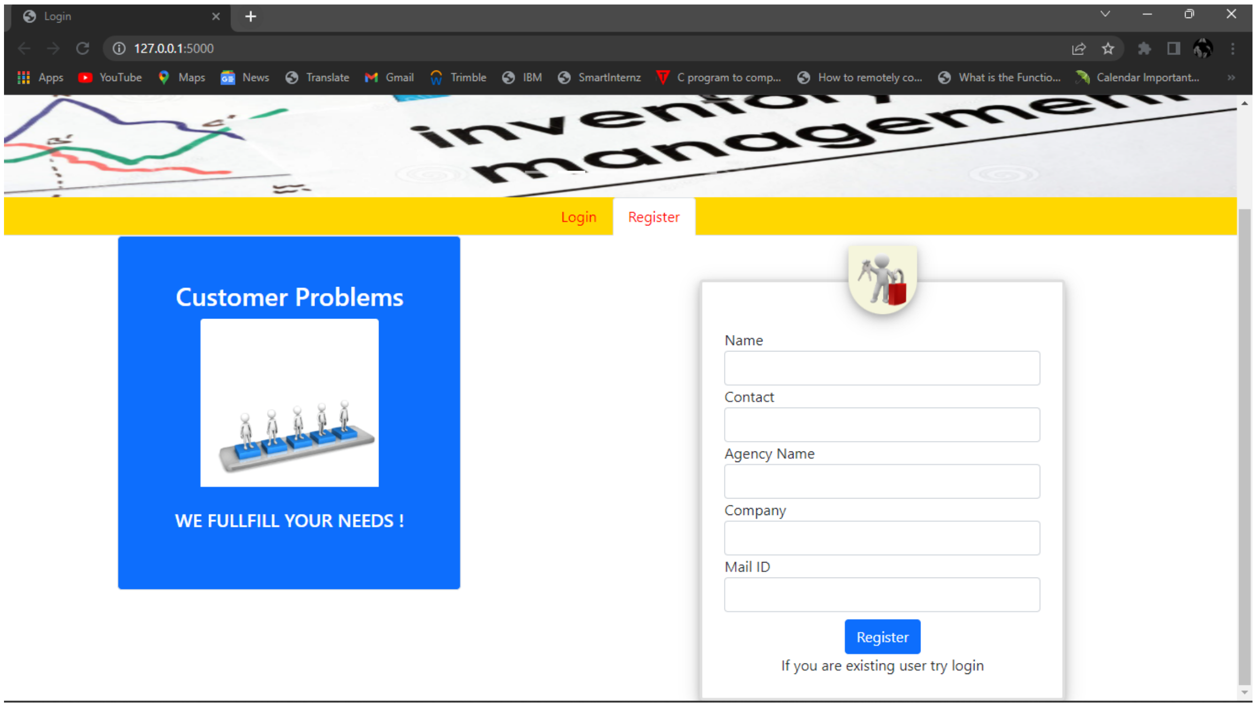
Sl. No.	**Input**	**Output**	**Status**
1	Login ID : 1001 Password : 9976627039	Login Successful	Success
2	Login ID : 1002 Password : 7358734464	Login Successful	Success
3	Login ID : 1002 Password : 9976627039	Login Failed	Fail

9. RESULTS

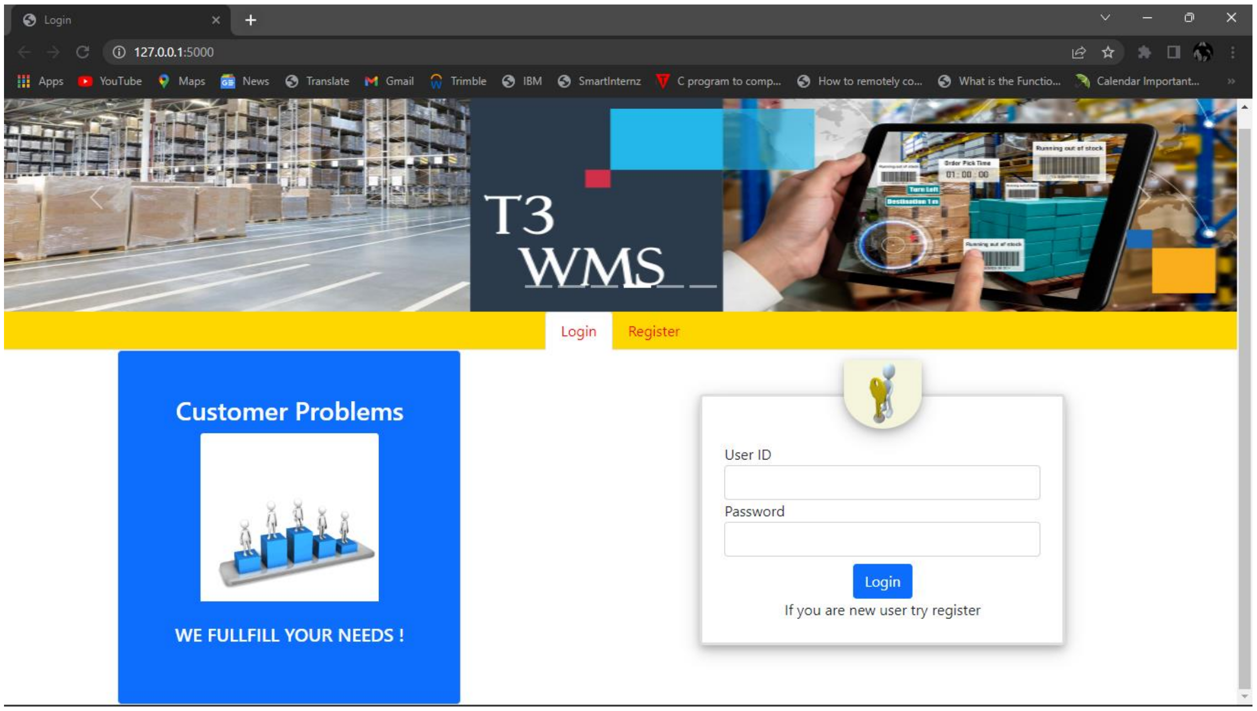
9.1. Performance Metrics

Outputs

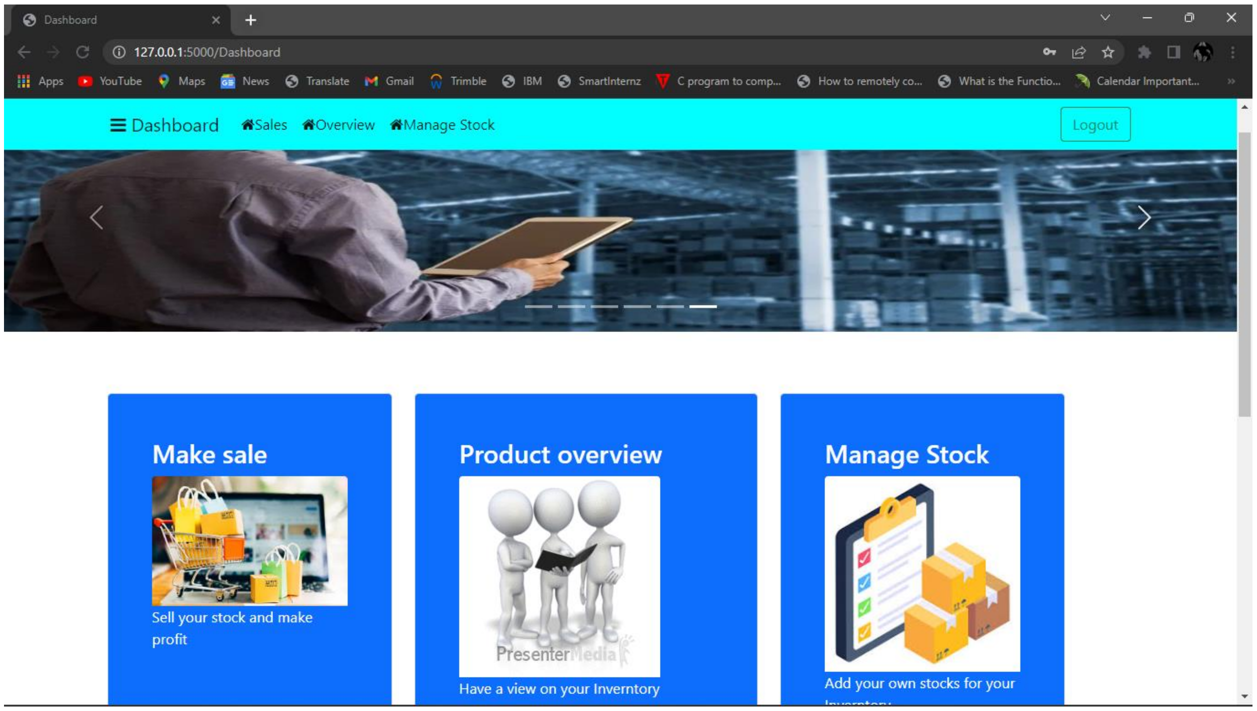
Register



Login



Dashboard


Dashboard
127.0.0.1:5000/Dashboard
Apps YouTube Maps News Translate Gmail Trimble IBM SmartInternz C program to comp... How to remotely co... What is the Functio... Calendar Important...
Dashboard
Sales
Overview
Manage Stock
Logout
PresenterMedia

Make sale

Sell your stock and make profit

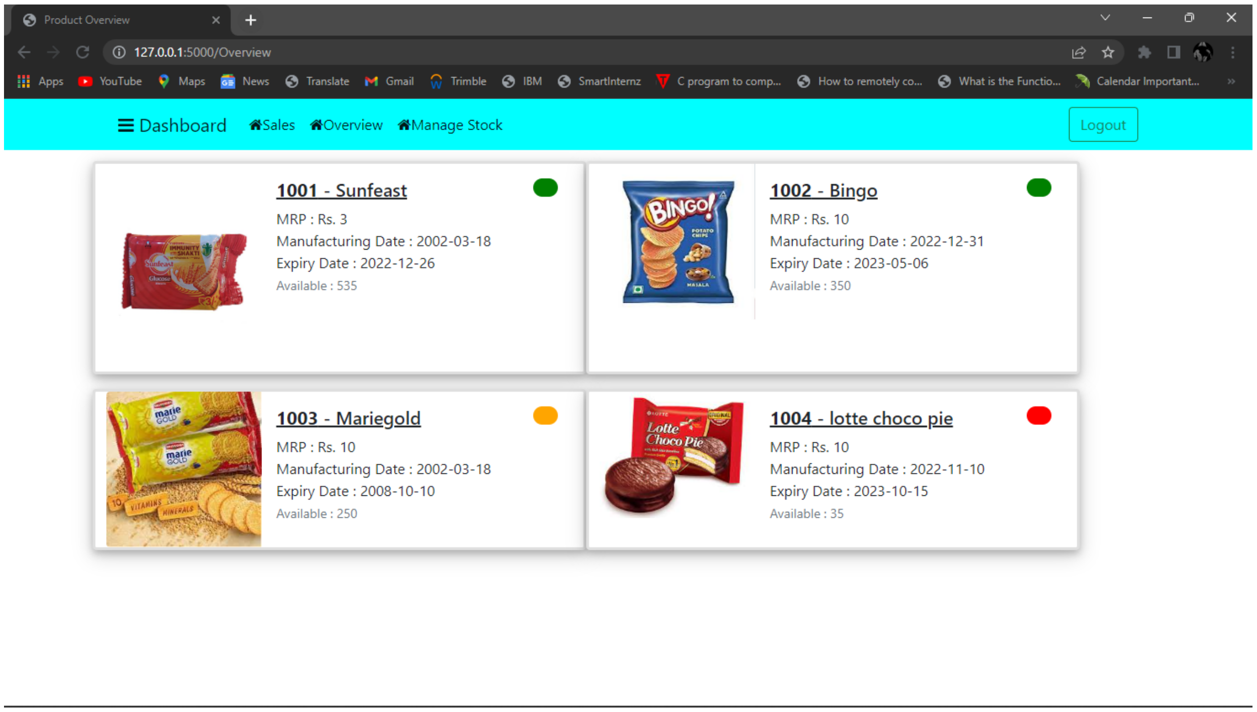
Product overview

Have a view on your Inverntory

Manage Stock

Add your own stocks for your Inventory

Product Overview


Product Overview
127.0.0.1:5000/Overview
Apps YouTube Maps News Translate Gmail Trimble IBM SmartInternz C program to comp... How to remotely co... What is the Functio... Calendar Important...
Dashboard
Sales
Overview
Manage Stock
Logout

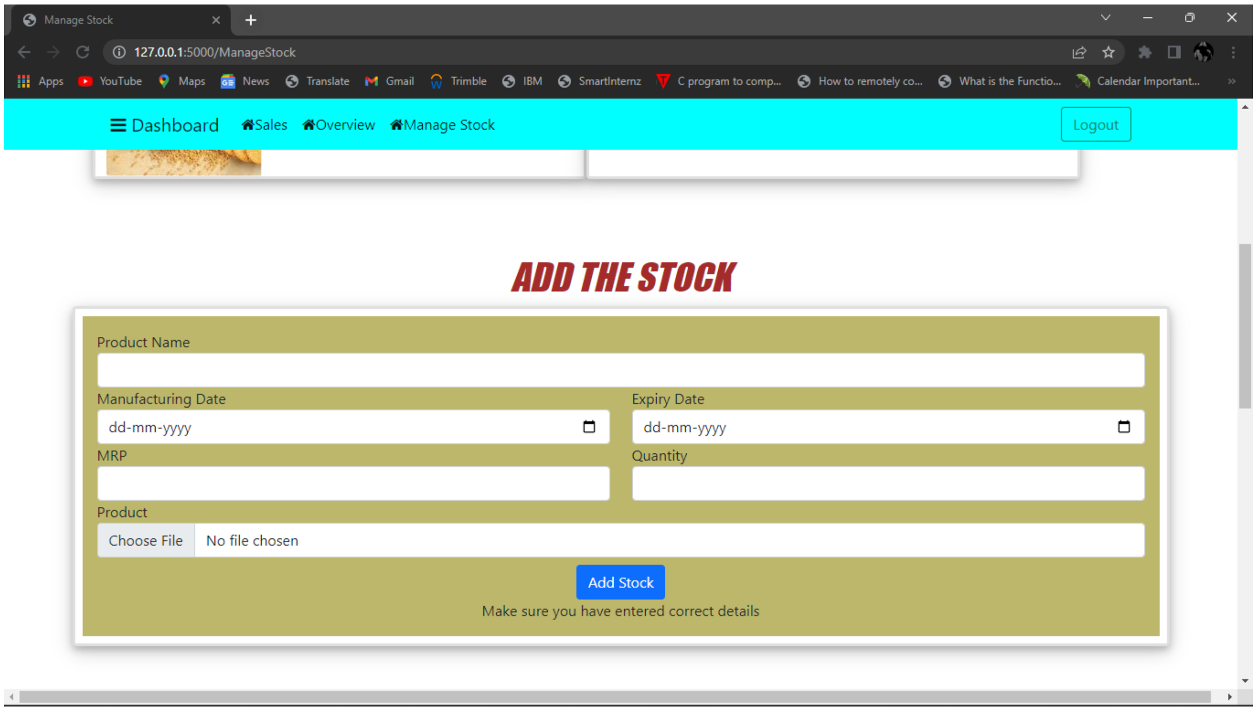
1001 - Sunfeast
MRP : Rs. 3
Manufacturing Date : 2002-03-18
Expiry Date : 2022-12-26
Available : 535

1002 - Bingo
MRP : Rs. 10
Manufacturing Date : 2022-12-31
Expiry Date : 2023-05-06
Available : 350

1003 - Mariegold
MRP : Rs. 10
Manufacturing Date : 2002-03-18
Expiry Date : 2008-10-10
Available : 250

1004 - lotte choco pie
MRP : Rs. 10
Manufacturing Date : 2022-11-10
Expiry Date : 2023-10-15
Available : 35

Add Stock



ADD THE STOCK

Product Name

Manufacturing Date: dd-mm-yyyy

Expiry Date: dd-mm-yyyy

MRP

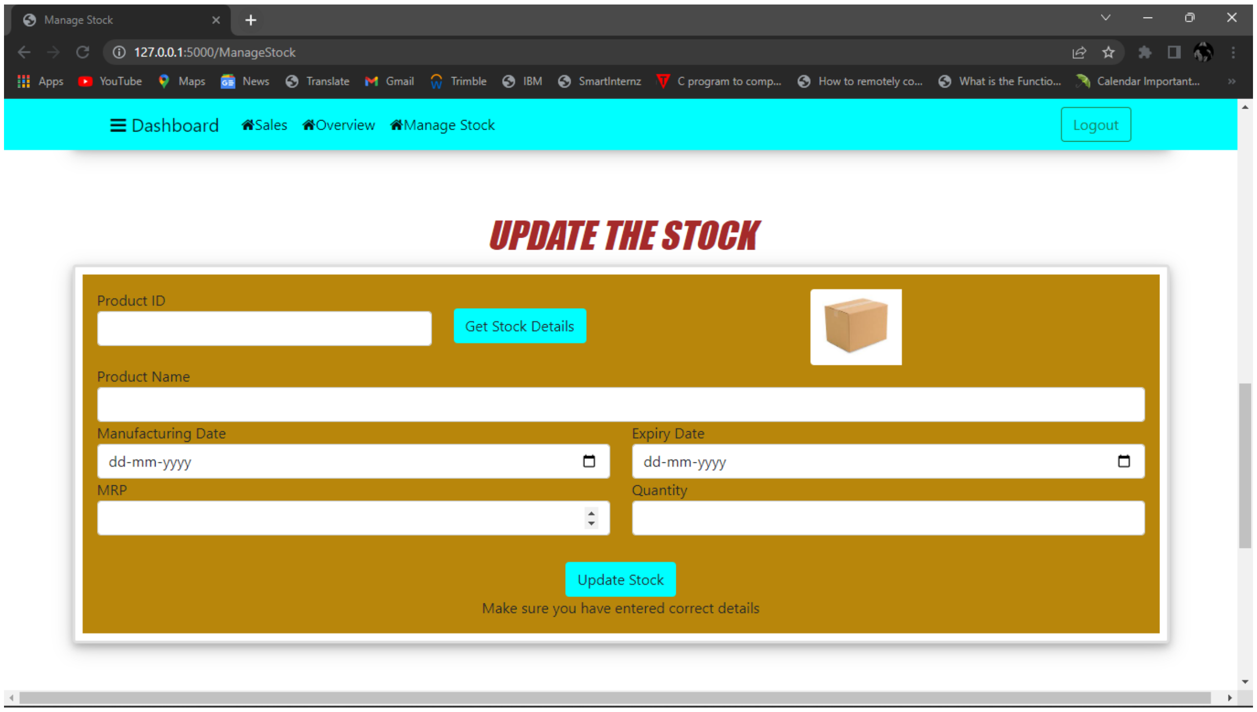
Quantity

Product: Choose File No file chosen

Add Stock

Make sure you have entered correct details

Update Stock



UPDATE THE STOCK

Product ID

Get Stock Details

Product Name

Manufacturing Date: dd-mm-yyyy

Expiry Date: dd-mm-yyyy

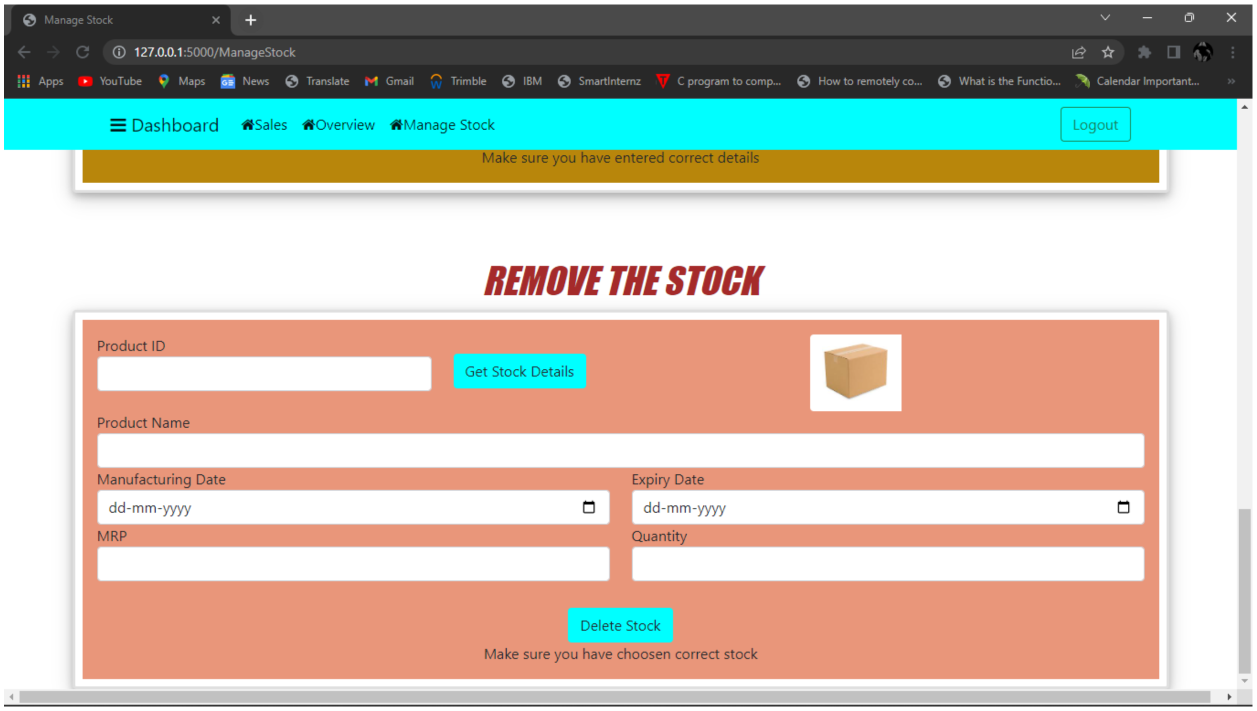
MRP

Quantity

Update Stock

Make sure you have entered correct details

Delete Stock



10. ADVANTAGES & DISADVANTAGES

Advantages

- Very easy navigation
- Less time Consuming
- User Friendly

Disadvantages

- Should be scalable
- Recommendation needed
- Increase Security

11. CONCLUSION

Thus the Inventory management system for retailers have been successfully implemented to notify their current availability in their own inventories with the different color indications

12. FUTURE SCOPE

The future scope for this system decided by us to make a generically and graphically maintenance and need to get recommendation for the trending stock in advance.

13. APPENDIX

Source Code

App.py

```
from asyncio.windows_events import NULL
import base64
from collections import UserDict
import tkinter  as tk
from tkinter import *
import io
from PIL import Image, ImageTk
from threading import activeCount
from turtle import st
from flask import Flask,render_template,request, session, redirect,url_for
import ibm_db
import re
from werkzeug.utils import secure_filename
from werkzeug.datastructures import  FileStorage
import PIL
import ibm_boto3
from ibm_botocore.client import Config, ClientError


COS_ENDPOINT="https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID="OVhg-B9OTj0Iwjjo1iioS9k-qKY6wahPUGzElt8at-EX"
COS_INSTANCE_CRN="crn:v1:bluemix:public:cloud-object-
storage:global:a/0644cd8c36ef4b70aed577434e50bbc4:b5d94bd0-cad9-4822-9c64-f953cb105e72::"


cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)

app = Flask(__name__)
app.secret_key = 'a'
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;
SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=xbh29234;PWD=Hdy6hbDtvv4SomNt;",",")
@app.route('/')

def homer():
  return render_template('Login.html')

@app.route('/Dashboard', methods = ['POST', "GET"])
def dashboard():
  return render_template('Dashboard.html')

@app.route('/Overview', methods = ['POST', "GET"])
def overview():
  stocks = ManageStock1()
  return render_template('ProductOverview.html',stocks=stocks)

@app.route('/Login1', methods = ['POST', "GET"])
def login():
  if request.method == 'POST':
    userid = request.form.get("userid")
    password = request.form.get("passwd")

    if(userid!="" and password!=""):
      sql = "select CONTACT from retailer where ID = ?"
      stmt = ibm_db.prepare(conn, sql)
      ibm_db.bind_param(stmt, 1, userid)
      ibm_db.execute(stmt)
      obj = ibm_db.fetch_assoc(stmt)
      try:
        passwd_contact=int(obj['CONTACT'])
        actual_contact = int(password)
      except:
        passwd_contact=0
        actual_contact = -1
      if (passwd_contact!=0) and (passwd_contact == actual_contact):
        session['USERID'] = userid
```

```
            return redirect(url_for("dashboard"))
    return render_template('Login.html')

@app.route('/Login', methods = ['POST', "GET"])
def signup():
    if request.method == 'POST':
        name = request.form.get("name")
        contact = request.form.get("contact")
        mailid = request.form.get("mailid")
        company = request.form.get("company")
        agency = request.form.get("agencyname")

        if(name!="" and contact !="" and mailid !="" and company != "" and agency !=""):
            stmt = ibm_db.prepare(conn, 'select max(ID)+1 as ID from retailer')
            ibm_db.execute(stmt)
            obj = ibm_db.fetch_assoc(stmt)
            nextID = 0
            print(obj)
            try:
                nextID = int(obj['ID'])
            except:
                nextID = 1001
            sql = f"insert into Retailer
values({nextID},'{name}',{contact},'{mailid}','{company}','{agency}')"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            return render_template('login.html',msg = 'Account created Successfully')
    return render_template('Login.html')

def ManageStock1():
    try:
        if not session.get("USERID"):
            return redirect("/login")
        userid = session.get("USERID")
        query = f"select * from stock where retailerid = {userid}"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.execute(stmt)
        stocks = []
        available = []
        while True:
            obj = ibm_db.fetch_tuple(stmt)
            if obj :
                stocks.append(obj)
                if int(obj[7]) > 300 :
                    available.append(1)
                elif int(obj[7]) > 50:
                    available.append(0)
                else:
                    available.append(-1)
            else:
                break
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
        return render_template('Dashboard.html')
    except Exception as e:
        print("Unable to retrieve bucket contents: {0}".format(e))
        return render_template('Dashboard.html')
    return zip(stocks,available)

@app.route('/ManageStock', methods = ['POST', "GET"])
def ManageStock():
    try:
        if not session.get("USERID"):
            return redirect("/login")
        userid = session.get("USERID")
        query = f"select * from stock where retailerid = {userid}"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.execute(stmt)
        stocks = []
        available = []
        while True:
            obj = ibm_db.fetch_tuple(stmt)
            if obj :
                stocks.append(obj)
                if int(obj[7]) > 300 :
                    available.append(1)
                elif int(obj[7]) > 50:
                    available.append(0)
                else:
                    available.append(-1)
            else:
                break
        # files = cos.Bucket('imsfr').objects.all()
        # files_names = []
        # for file in files:
        #     files_names.append(file.key)
        #     # print(file)
        #     # print("Item: {0} ({1} bytes).".format(file.key, file.size))
        return render_template('ManageStock.html',stocks = zip(stocks,available),getStock=NULL)

    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
        return render_template('Dashboard.html')
    except Exception as e:
        print("Unable to retrieve bucket contents: {0}".format(e))
        return render_template('Dashboard.html')

@app.route('/addStock',methods=['POST'])
def upload():
    if request.method == 'POST':
        sname = request.form.get("addproductName")
        mdate = request.form.get("manDate")
        edate = request.form.get("expDate")
        mrp = request.form.get("mrp")
        qty = request.form.get("qty")
    if sname and mdate and edate and mrp and qty:
        if not session.get("USERID"):
            return redirect("/login")
        userid = session.get("USERID")
        query = f"select max(ID)+1 as SID from STOCK where RETAILERID = {userid}"
        stmt = ibm_db.prepare(conn, query)
```

```
ibm_db.execute(stmt)
obj = ibm_db.fetch_assoc(stmt)
try:
   StockID=int(obj['SID'])
except:
   StockID = 1001

f = request.files['addProductImage']
Stock_img_name=f"img_{userid}_{StockID}"
try:
   part_size = 1024 * 1024 * 5

   file_threshold = 1024 * 1024 * 15

   transfer_config = ibm_boto3.s3.transfer.TransferConfig(
         multipart_threshold=file_threshold,
         multipart_chunksize=part_size
      )

   content = f.read()
   cos.Object('imsfr', Stock_img_name).upload_fileobj(
            Fileobj=io.BytesIO(content),
            Config=transfer_config
         )

except ClientError as be:
      # print("CLIENT ERROR: {0}\n".format(be))
      return redirect(url_for('login'))

except Exception as e:
      # print("Unable to complete multi-part upload: {0}".format(e))
      return redirect(url_for('login'))

query = f"insert into STOCK values({StockID},'{sname}','{mdate}','{edate}',{mrp},'{Stock_img_name}',{userid},{qty})"
stmt = ibm_db.prepare(conn, query)
ibm_db.execute(stmt)
```

```
        return redirect(url_for('ManageStock'))



        try:
            query = f"select * from stock where retailerid = {userid}"
            stmt = ibm_db.prepare(conn, query)
            ibm_db.execute(stmt)
            obj = ibm_db.fetch_assoc(stmt)
            print(obj)
            files = cos.Bucket('imsfr').objects.all()
            files_names = []
            for file in files:
                files_names.append(file.key)
                # print(file)
                # print("Item: {0} ({1} bytes).".format(file.key, file.size))
            return render_template('ManageStock.html',files=files_names)

        except ClientError as be:
            # print("CLIENT ERROR: {0}\n".format(be))
            return render_template('login.html')
        except Exception as e:
            # print("Unable to retrieve bucket contents: {0}".format(e))
            return render_template('login.html')

@app.route('/updateStock',methods=['POST'])
def update():
    if not session.get("USERID"):
        return redirect("/login")
    userid = session.get("USERID")
    if request.method == 'POST':
        if request. form['submit'] == 'Get Stock Details':
            pid = request.form.get("pid")
            query = f"select * from stock where ID = {pid} and retailerid = {userid}"
            stmt = ibm_db.prepare(conn, query)
            ibm_db.execute(stmt)
            stock = ibm_db.fetch_tuple(stmt)
            print(stock)
```

```
                stocks = ManageStock1()
                return render_template('ManageStock.html',getStock = stock,stocks=stocks)
            elif request. form['submit'] == 'Update Stock':
                stockID = request.form.get("pid")
                sname = request.form.get("addproductName")
                mdate = request.form.get("manDate")
                edate = request.form.get("expDate")
                mrp = request.form.get("mrp")
                qty = request.form.get("qty")
            if sname and mdate and edate and mrp and qty:
                if not session.get("USERID"):
                    return redirect("/login")
                userid = session.get("USERID")

                query = f"update STOCK set PRODUCT='{sname}',MDATE='{mdate}',EDATE='{edate}',MRP={mrp},Quantity={qty} where RETAILERID={userid} and ID={stockID}"
                stmt = ibm_db.prepare(conn, query)
                ibm_db.execute(stmt)
                stocks = ManageStock1()
            return redirect(url_for('ManageStock',getStock = stocks))

    @app.route('/deleteStock',methods=['POST'])
    def delete():
        if not session.get("USERID"):
            return redirect("/login")
        userid = session.get("USERID")
        if request.method == 'POST':
            if request. form['submit'] == 'Get Stock Details':
                pid = request.form.get("pid")
                query = f"select * from stock where ID = {pid} and retailerid = {userid}"
                stmt = ibm_db.prepare(conn, query)
                ibm_db.execute(stmt)
                stock = ibm_db.fetch_tuple(stmt)
                print(stock)
                stocks = ManageStock1()
                return render_template('ManageStock.html',getStock = stock,stocks=stocks)
    elif request. form['submit'] == 'Delete Stock':
        stockID = request.form.get("pid")
        sname = request.form.get("addproductName")
        mdate = request.form.get("manDate")
        edate = request.form.get("expDate")
        mrp = request.form.get("mrp")
        qty = request.form.get("qty")
    if sname and mdate and edate and mrp and qty:
        if not session.get("USERID"):
            return redirect("/login")
        userid = session.get("USERID")

        query = f"delete from STOCK where RETAILERID={userid} and ID={stockID}"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.execute(stmt)
        stocks = ManageStock1()
    return redirect(url_for('ManageStock',getStock = stocks))


if __name__ == '__main__':
    app.run(debug=True)
```

GitHub & Project Demo Link

GitHut Link **:** https://github.com/IBM-EPBL/IBM-Project-2496-1658472843

Project Demo Link **:**

https://drive.google.com/file/d/1tpj7hSMKM7NAe4G2VF1AB0VzjFA-ie-m/view?usp=sharing