HX8001- PROFESSIONAL READINESS FOR INNOVATION,EMPLOYABILITY AND ENTREPRENEURSHIP

SUBMITTED BY

TEAM ID : PNT2022TMID31378

SRIVARSHAN S- 710719104096

SABARISHWARAN G- 710719104080

SARAVANAPRAKASH S - 710719104085

PRADEESH R - 710719104072

Pursuing final year of the degree
BACHELOR OF ENGINEERING
In
DR.N.G.P. INSTITUTE OF TECHNOLOGY

NOV 2022

COIMBATORE - 641048

ANNA UNIVERSITY: CHENNAI 600 025

ACKNOWLEDGEMENT

1. INTRODUCTION

1.1 PROJECT OVERVIEW

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.

1.2 PURPOSE

Retail inventory management tools and methods give retailers more information with which to run their businesses, including: Product locations. Quantities of each product type. Which stock sells well and which doesn't, by location and sales channel.

2. LITERATURE

2.1 EXISTING PROBLEMS

- Lack of Inventory Visibility.
- Inefficient Inventory Management Process or Software.
- Tracking Obsolete Material.
- Identifying Incorrectly Located Materials.
- Keeping up with Overstocks.
- Managing Inventory Waste & Defects.
- Lack of Centralized Inventory Hub.
- Changing Demand

2.2 REFERENCES

- Digital Reference by tranquilbs website
- Reference from tranquil content writer.
- Reference by https://www.tranquilbs.com/inventory-management-problems/

2.3 PROBLEM STATEMENT DEFINITION

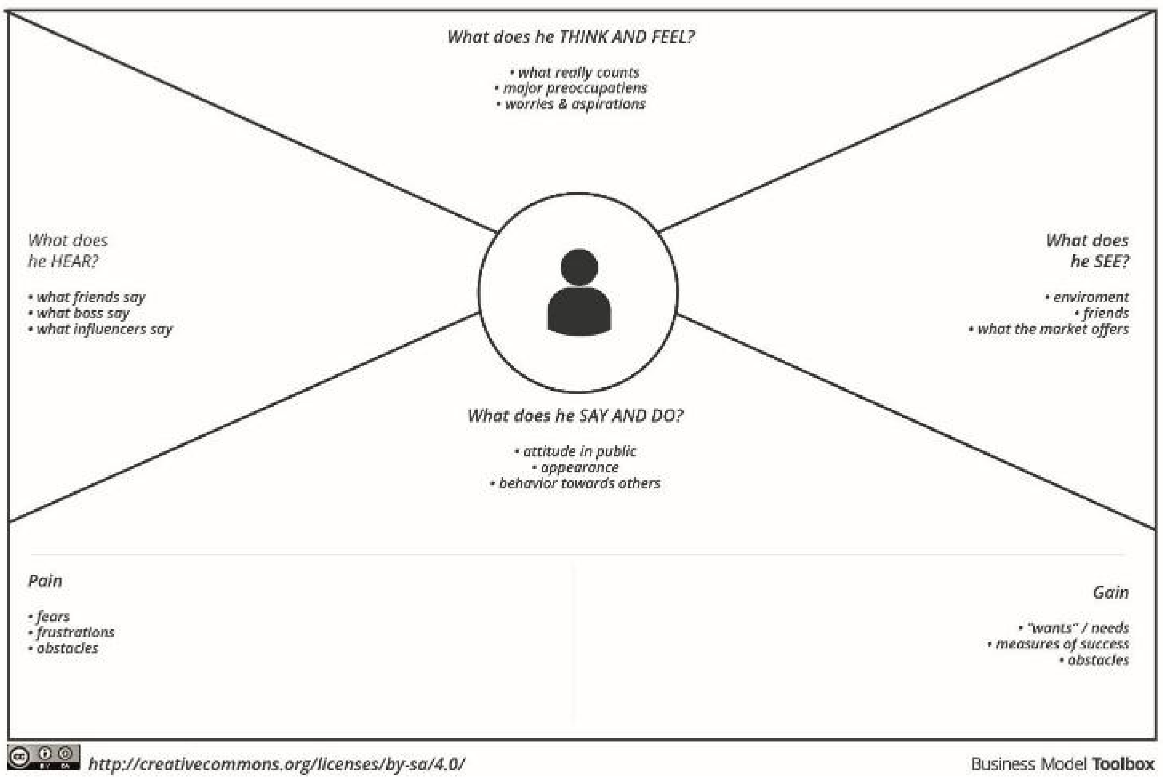
In inventory systems, demand is usually uncertain, and the lead-time can also vary. To avoid shortages, managers often maintain a safety stock. In such situations, it is not clear what order quantities and reorder points will minimize expected total inventory cost. Simulation models can address this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

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours 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.

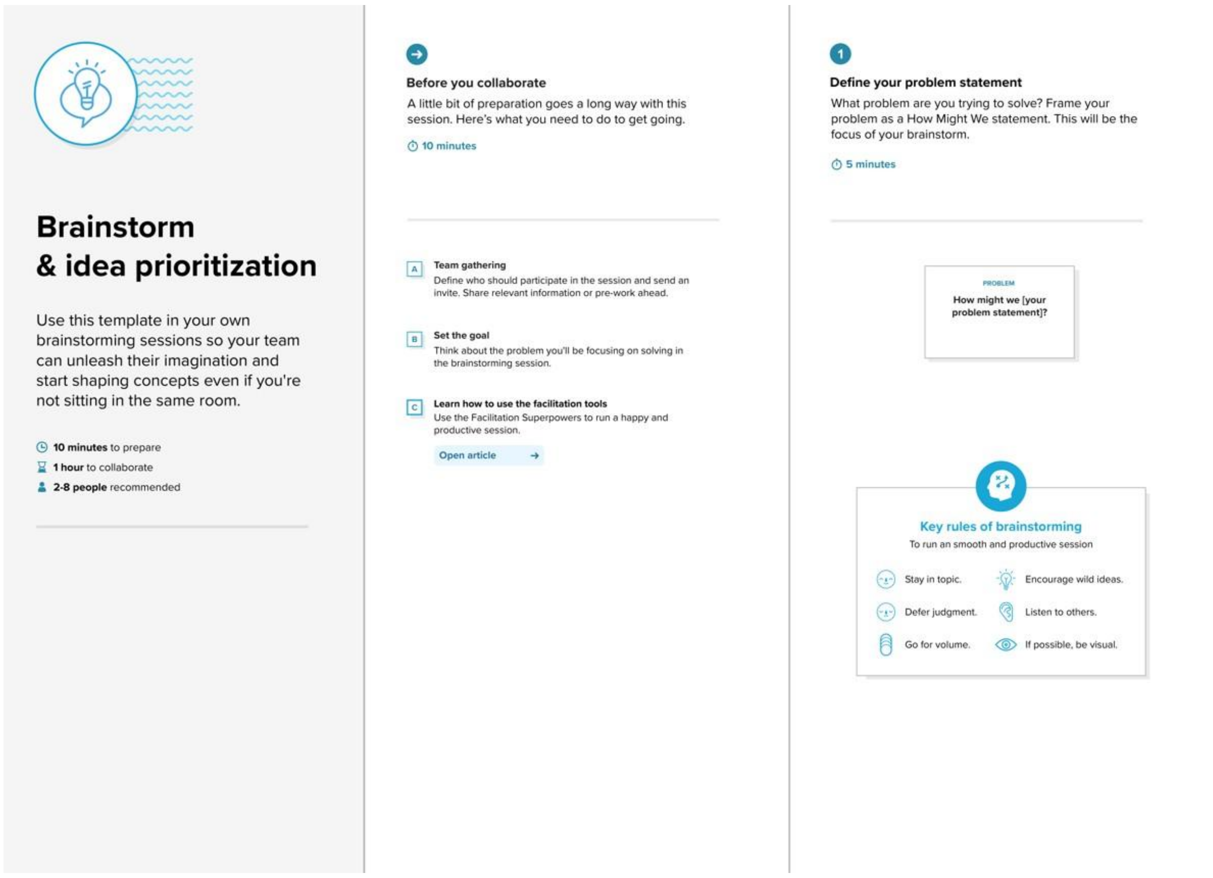


3.2 IDEATION AND BRAINSTORMING

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: https://www.mural.co/templates/empathy-map-canvas

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



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

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.

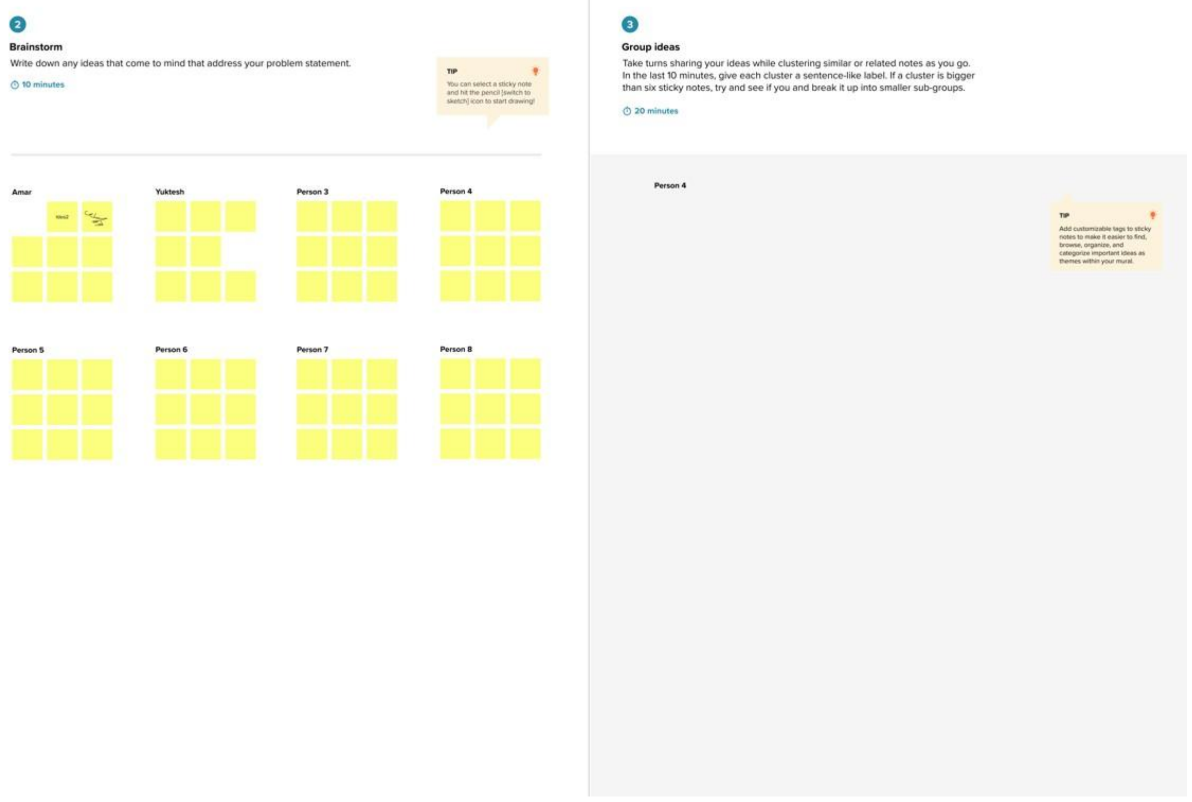
5 minutes

PROBLEM
How might we [your problem statement]?

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. In the last 10 minutes, 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.)

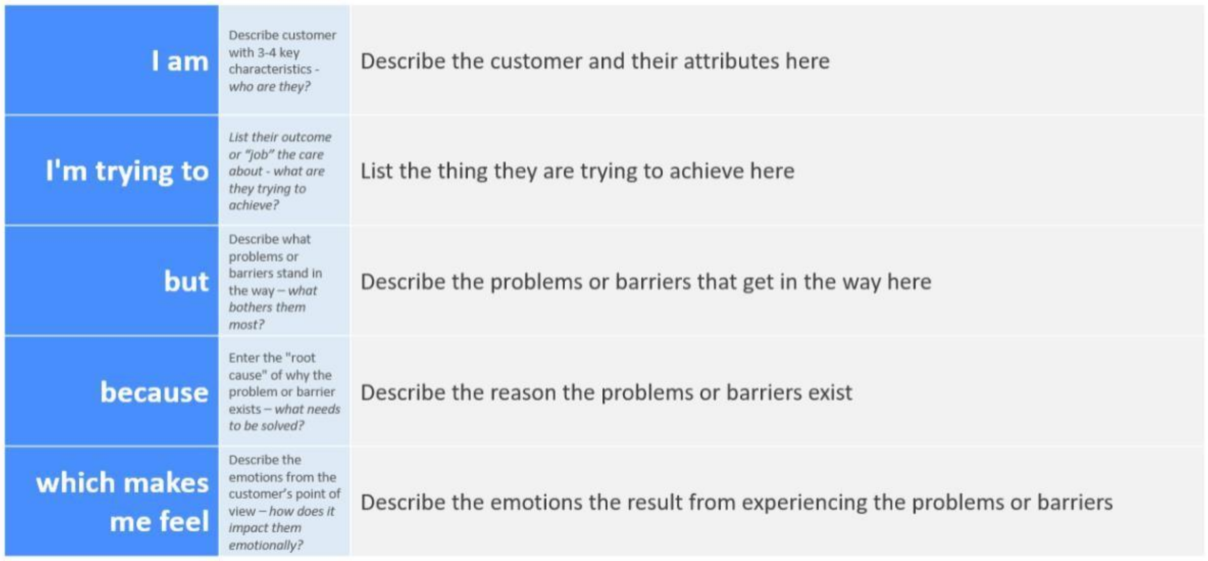
3.3 PROPOSED SOLUTION

- Centralized Tracking: Consider upgrading to tracking software that provides automated features for re-ordering and procurement.
- Transparent Performance.
- Stock Auditing.
- Demand Forecasting.
- Add Imagery.
- Go Paperless.
- Preventive Control.
- Measure Control.
- Safety stock

3.4 PROBLEM SOLUTION FIT

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

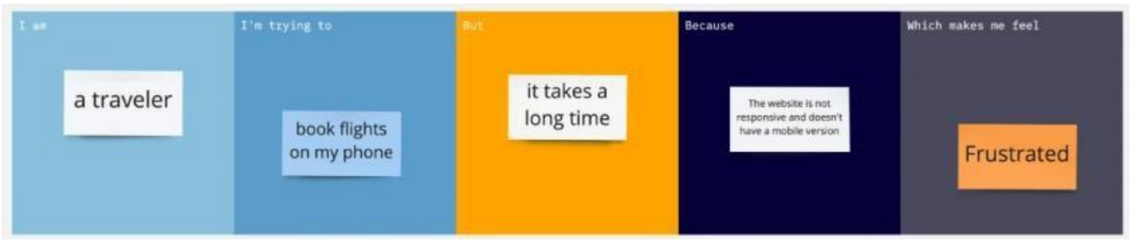
A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

Reference: https://miro.com/templates/customer-problem-statement/

Example:



Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1					
PS-2					

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

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 LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login	Login with username Login with password
FR-4	Centralized Record of all product	Product name, Stock keep unit, brand, retail price, product category, lot number , expire date, vendor details, wholesale cost, minimum reorder amount, case quantity amount, reorder lead time
FR-5	Stock location identification	Provide number label for- Shelf, Rack and Boxes
FR-6	Periodical stock checking	Physical counting and Cycle counting
FR-7	Integration of sales and inventory data	sales administration and database upkeep FIFO,LILO according to the goods
FR-8	Purchase management and Forecasting	Order review and placement, Avoid risk stock, review product, priorities purchases based on an item's profitability, popularity, and lead time, ABC,FSC,XYZ,JIT techniques
FR-9	Markdown and promotion	Show product discount, Maintain enough stock on hand to meet demand.
FR-10	Management of Receiving Stock	Accurately recording goods on an inventory
FR-11	Returns Management System	Check for damage or defects and return to vendor as needed If sellable add it to inventory counts
FR-12	Determination of death stock	Return to the vendor for credits
FR-13	Inventory KPIs(Key Performance Indicator)	Sale KPIs, Receive KPIs, Operational KPIs, Employee KPIs

4.2 NON-FUNCTIONAL REQUIREMENTS

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

FR No.	Non-Functional Requirement	Description

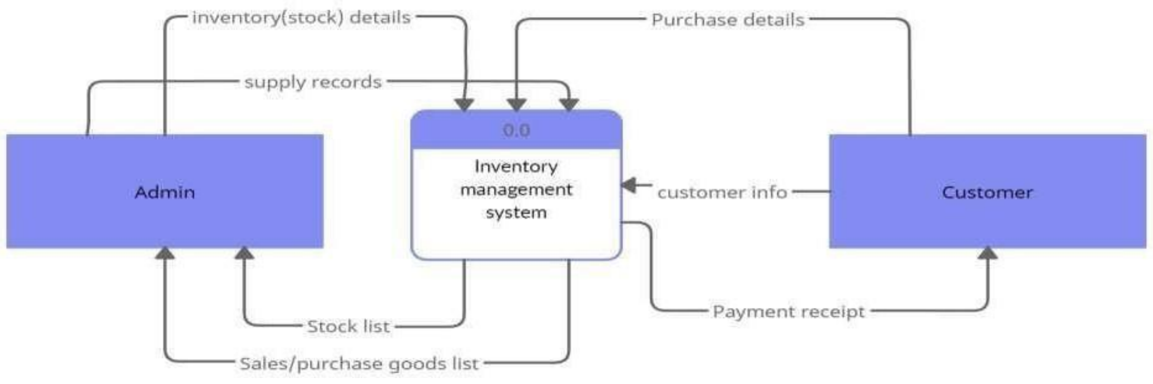
NFR-1	**Usability**	This system must be easy to use by both managers and chefs, such that they do not need to read an extensive number of manuals; it must be quickly accessible by both managers and chefs; it must be intuitive and simple in the way it displays all relevant data and relationships; and the menus of the system are easily navigable by the users with buttons that are easy to understand.
NFR-2	**Security**	The security requirements deal with the primary security. Only authorized users can access the system with user name and password of administrator.
NFR-3	**Reliability**	The system must give accurate inventory status to the user continuously. Any inaccuracies are corrected by regularly comparing the actual levels to the levels displayed in the system. The system must successfully add any recipe, ingredients, vendors, or special occasions given by the user and provide estimations and inventory status in relevance to the newly updated entities.
NFR-4	**Performance**	The system must not lag, because the workers using it don't have downtime to wait for it to complete an action. The system must successfully complete updating the databases, adding new recipes, ingredients, vendors, and occasions every time the user requests such a process. All the functions of the system must be available to the user every time the system is turned on. The calculations performed by the system must comply with the norms set by the user and should not vary unless explicitly changed by the user.
NFR-5	**Availability**	The software will be available only to administrator of the organization and the product as well as customer details will be recorded by him. He can add customers, Update and delete them as well as add new products and manage them
NFR-6	**Scalability**	The ability of a system to handle a growing amount of work.

5. PROJECT DESIGN

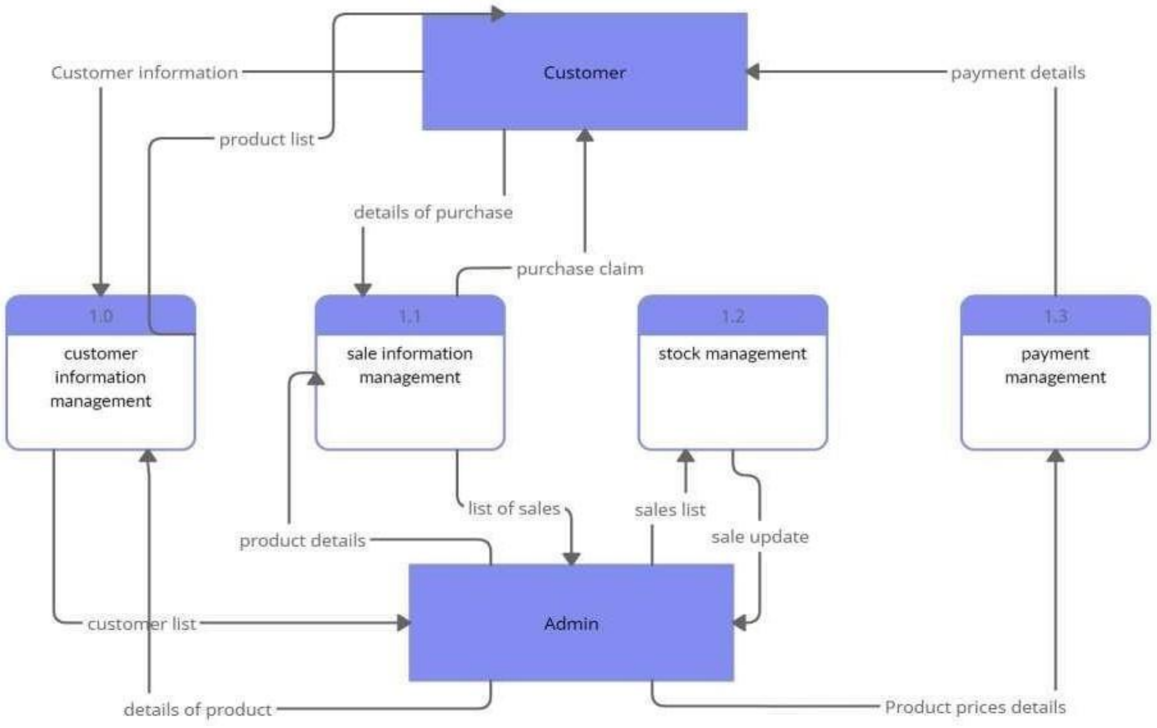
5.1 DATA FLOW DIAGRAMS

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 enters and leaves the system, what changes the information, and where data is stored.

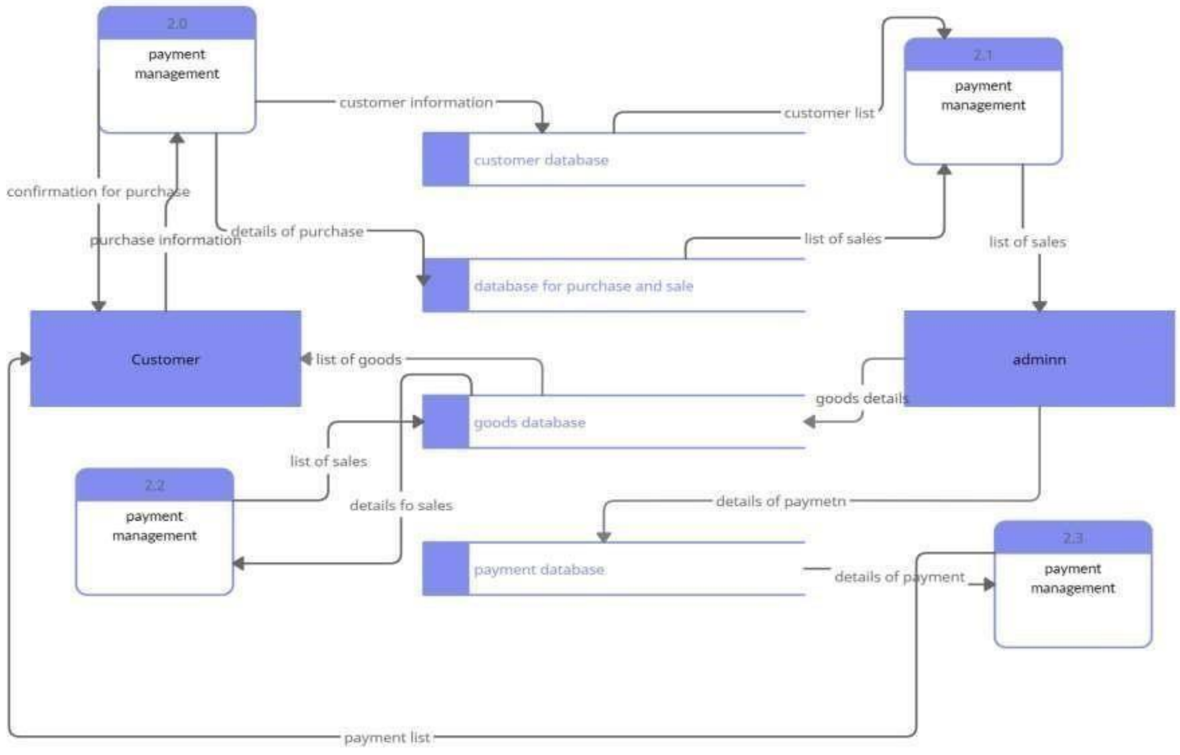
Level 0 Data Flow Diagram for inventory management system for retailers:



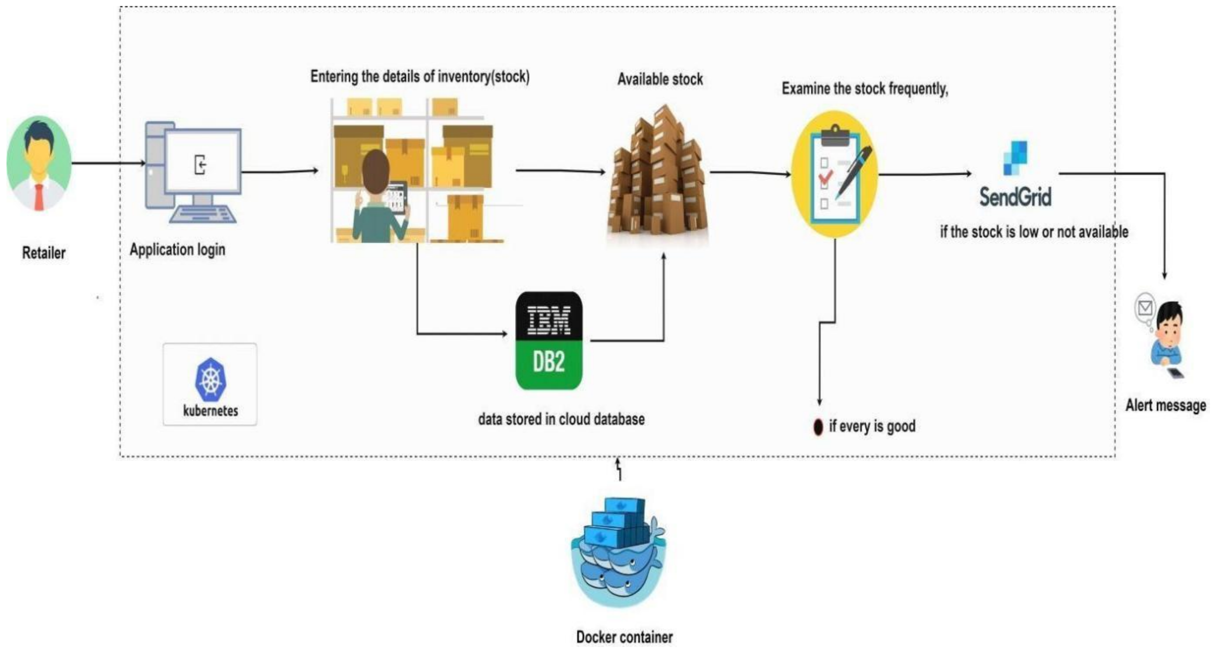
Level 1 Data Flow Diagram for inventory management system for retailers:



Level 2 Data Flow Diagram for inventory management system for retailers:



5.2 SOLUTION & TECHNICAL ARCHITECTURE


Retailer
Application login
Entering the details of inventory(stock)
Available stock
Examine the stock frequently,
SendGrid
if the stock is low or not available
IBM DB2
data stored in cloud database
kubernetes
if every is good
Alert message
Docker container

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
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 conformation email and click confirm button	Medium	Sprint-1
	Login	USN-5	As a user, I can sign in to the application by giving my email & password	I can access my account	High	Sprint-1
	Dashboard	USN-6	As a user, it displays the stock , current sale demand product	I can see available stock , daily sale	High	Sprint-2
Customer (Web user)	Application	USN-7	As a user, I can register, sign in, and shop the products simply	I can access account anywhere	High	Sprint-3
Customer Care Executive	Update inventory details	USN- 8	To monitor the track of inventory and availability	I can improve the productivity	High	Sprint-4
Administrator	Update purchased stock	USN-9	To update purchased goods in database	I can update the new purchased product	High	Sprint-3
Customer care executive	Customer feedback verification	USN-10	To get a clear understanding about our application and for the convenience of the user	I can fulfil the customer expectations	High	Sprint-4
	Inventory control	USN-11	To avoid stock overflow and run out	I can alert mail if stock run out	Medium	Sprint- 2
administrator	Quality checking	USN-12	To maintain the product and improving the customer relationship	I can improve my product quality	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

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	31 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:

Sprint Duration : 6 Days

Velocity of the Team : 20 (points per sprint)

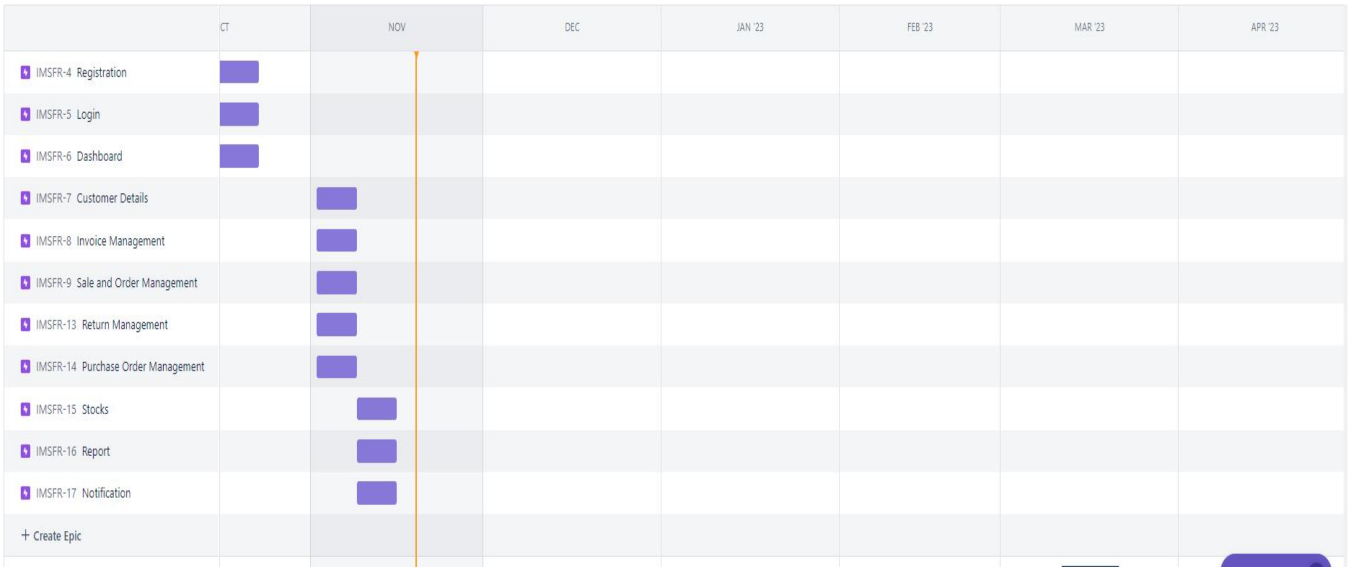
Team's Average Velocity :

AV = story points /*velocity sprint duration*

= 206

= 3. 3

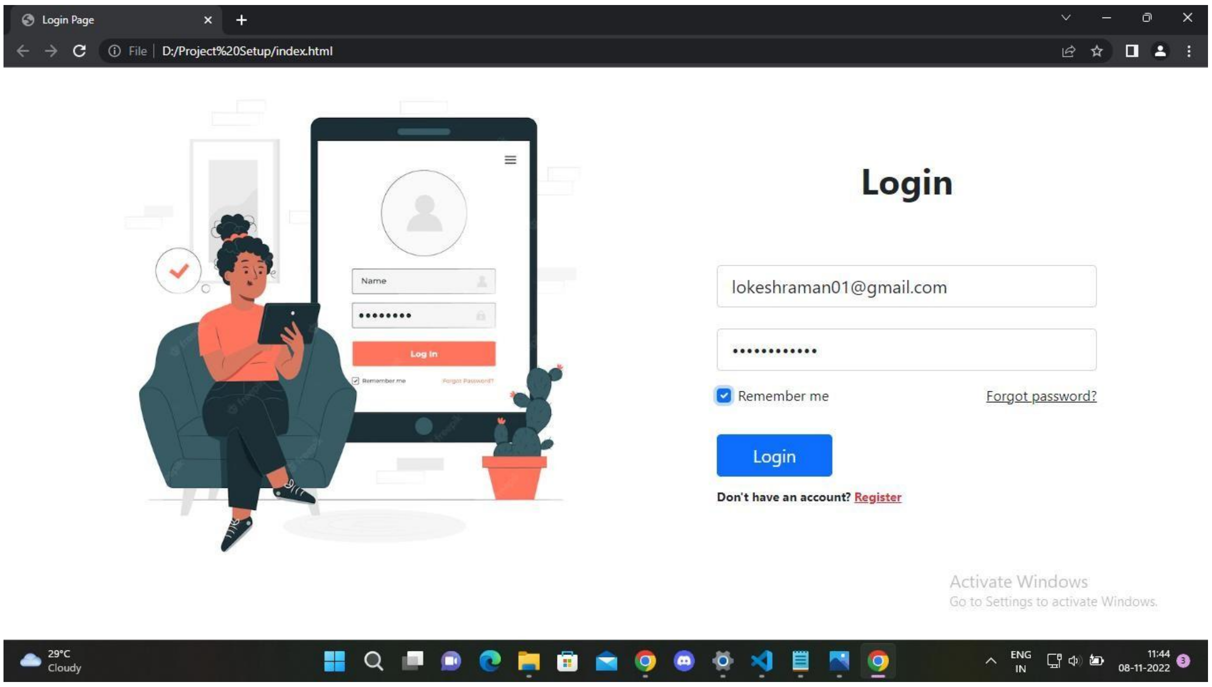
6.2 REPORTS FROM JIRA



7. CODING & SOLUTION

7.1 FEATURE 1

Log-In



Login

lokeshraman01@gmail.com

••••••••••••

Remember me   Forgot password?

Login

Don't have an account? Register

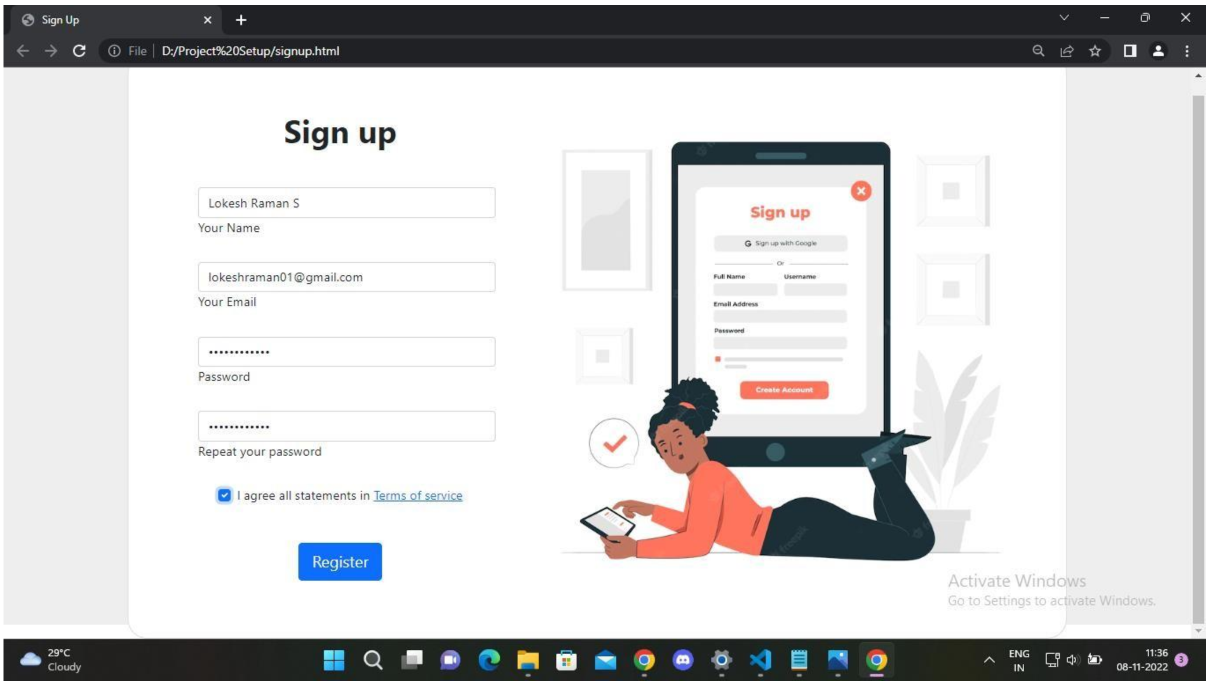
Login.html

```
<!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">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
    <style>
        .divider:after,
            .divider:before
                {content: "";
                flex: 1; height:
                1px;
                background: #eee;
            }
        .h-custom {
            height: calc(100% - 73px);
        }
        @media (max-width: 450px) {
            .h-custom
                { height: 100%;
            }
        }
    </style>
    <title>Login Page</title>
</head>
<body>
    <section class="vh-100">
        <div class="container-fluid h-custom">
          <div class="row d-flex justify-content-center align-items-center h-100">
```

```
            <div class="col-md-9 col-lg-6 col-xl-5">
              <img src="https://img.freepik.com/free-vector/tablet-login-concept-illustration_114360-
7863.jpg?w=740&t=st=1667711849~exp=1667712449~hmac=d4ef0be91c59f7ea273e94343f3799a0
bda80f45635cb1d6187e575f2d5b5fd9"
                class="img-fluid" alt="Sample image">
            </div>
            <div class="col-md-8 col-lg-6 col-xl-4 offset-xl-1">
                <p class="text-center h1 fw-bold mb-5 mx-1 mx-md-4 mt-4">Login</p>
              <form>

                <!-- Email input -->
                <br>
                <div class="form-outline mb-4">
                  <input type="email" id="form3Example3" class="form-control form-control-lg"
                    placeholder="Enter a valid email address" />
                  <!-- <label class="form-label" for="form3Example3">Email address</label> -->
                </div>

                <!-- Password input -->
                <div class="form-outline mb-3">
                  <input type="password" id="form3Example4" class="form-control form-control-lg"
                    placeholder="Enter password" />
                  <!-- <label class="form-label" for="form3Example4">Password</label> -->
                </div>

                <div class="d-flex justify-content-between align-items-center">
                  <!-- Checkbox -->
                  <div class="form-check mb-0">
                    <input class="form-check-input me-2" type="checkbox" value="" id="form2Example3"
/>
                    <label class="form-check-label" for="form2Example3">
                      Remember me
                    </label>
                  </div>
                  <a href="#!" class="text-body">Forgot password?</a>
                </div>

                <div class="text-center text-lg-start mt-4 pt-2">
                  <button type="button" class="btn btn-primary btn-lg"
                    style="padding-left: 2.5rem; padding-right: 2.5rem;">Login</button>
                  <p class="small fw-bold mt-2 pt-1 mb-0">Don't have an account? <a href="signup.html"
                      class="link-danger">Register</a></p>
                </div>

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

  </section>
</body>
</html>
```

7.2 FEATURE 2



Sign up

Lokesh Raman S
Your Name

lokeshraman01@gmail.com
Your Email

••••••••••••
Password

••••••••••••
Repeat your password

I agree all statements in Terms of service

Register

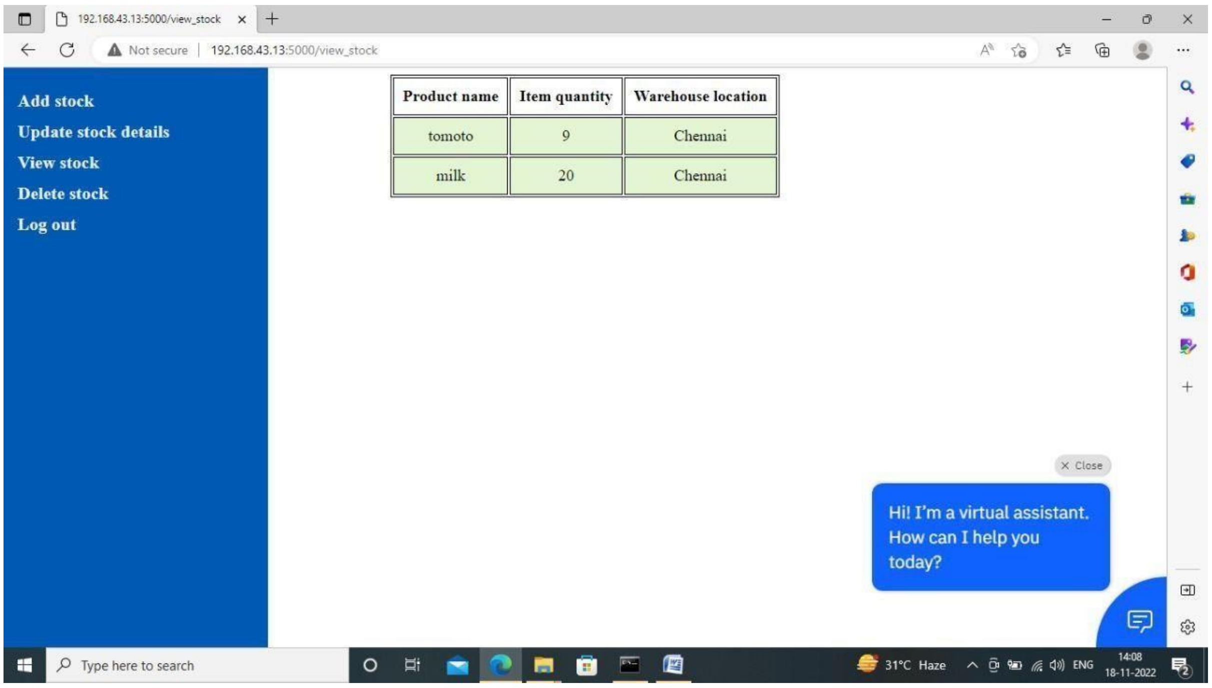
Signup.html

```
<!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">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

  <title>Sign Up</title>

</head>

<body>

  <section class="vh-100" style="background-color: #eee;">

    <div class="container h-100">

     <div class="row d-flex justify-content-center align-items-center h-100">

      <div class="col-lg-12 col-xl-11">

       <div class="card text-black" style="border-radius: 25px;">

        <div class="card-body p-md-5">

         <div class="row justify-content-center">

          <div class="col-md-10 col-lg-6 col-xl-5 order-2 order-lg-1">


           <p class="text-center h1 fw-bold mb-5 mx-1 mx-md-4 mt-4">Sign up</p>


           <form class="mx-1 mx-md-4">


            <div class="d-flex flex-row align-items-center mb-4">

             <i class="fas fa-user fa-lg me-3 fa-fw"></i>
```

```
                <div class="form-outline flex-fill mb-0">
                  <input type="text" id="form3Example1c" class="form-control" />
                  <label class="form-label" for="form3Example1c">Your Name</label>
                </div>
              </div>

              <div class="d-flex flex-row align-items-center mb-4">
                <i class="fas fa-envelope fa-lg me-3 fa-fw"></i>
                <div class="form-outline flex-fill mb-0">
                  <input type="email" id="form3Example3c" class="form-control" />
                  <label class="form-label" for="form3Example3c">Your Email</label>
                </div>
              </div>

              <div class="d-flex flex-row align-items-center mb-4">
                <i class="fas fa-lock fa-lg me-3 fa-fw"></i>
                <div class="form-outline flex-fill mb-0">
                  <input type="password" id="form3Example4c" class="form-control" />
                  <label class="form-label" for="form3Example4c">Password</label>
                </div>
              </div>

              <div class="d-flex flex-row align-items-center mb-4">
                <i class="fas fa-key fa-lg me-3 fa-fw"></i>
                <div class="form-outline flex-fill mb-0">
                  <input type="password" id="form3Example4cd" class="form-control" />
                  <label class="form-label" for="form3Example4cd">Repeat your password</label>
                </div>
              </div>

              <div class="form-check d-flex justify-content-center mb-5">
                <input class="form-check-input me-2" type="checkbox" value=""
id="form2Example3c" />
```

```
                    <label class="form-check-label" for="form2Example3">
                      I agree all statements in <a href="#!">Terms of service</a>
                    </label>
                  </div>

                  <div class="d-flex justify-content-center mx-4 mb-3 mb-lg-4">
                    <button type="button" class="btn btn-primary btn-lg">Register</button>
                  </div>

                </form>

              </div>
              <div class="col-md-10 col-lg-6 col-xl-7 d-flex align-items-center order-1 order-lg-2">

                <img src="https://img.freepik.com/free-vector/sign-up-concept-illustration_114360-
7865.jpg?w=740&t=st=1667712441~exp=1667713041~hmac=4224cf9893b8b9c5e20fa396de1b4c00
1b8f2f9241d21dc07e260e9918c34018"
                  class="img-fluid" alt="Sample image">

              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
</body>
</html>
```

7.3. DATABASE SCHEMA



Product name	Item quantity	Warehouse location
tomoto	9	Chennai
milk	20	Chennai

Hi! I'm a virtual assistant. How can I help you today?

8. TESTING

8.1 TEST CASES

Test case	feature	component	Test scenario	Expected result	Actual result	status	comments	bug	Executed by
Sign in	Functional	Login page	Verify user can see the sign in option	can visible	Yes visible	pass	successful	-	Sathish kumar
Sign up	Functional	Login page	Verify user has the option to sign up	Can visible	Yes visible	pass	Successful	-	Sivanesan

8.2.USER ACCEPTANCE TESTING

Test case : Testing the Add Recipe Interface and its functioning

Case 1: Testing the Quantity input field.

Case 2: Testing the Recipe Name field.

Case 3: Testing the Ingredients in recipe list and Quantity of ingredient list. Case 4: Testing the available ingredients list.

Case 5: Testing the all the above cases together and checking if the entries are updated to the tables in database.

Test Case : Check Threshold Interface

Case 1: Check if the Ingredients under the threshold values are shown in the Ingredients below threshold list.

Case 2: Check if the Create order button asks the user to enter values for all the ingredients listed under the ingredients below threshold list.

Case 3: Check if pressing the Process Order button creates a file with the order details in it.

Test Case : Testing the Update after sales interface

Case 1: Test the Recipe list box.

Case 2: Test the quantity text field..

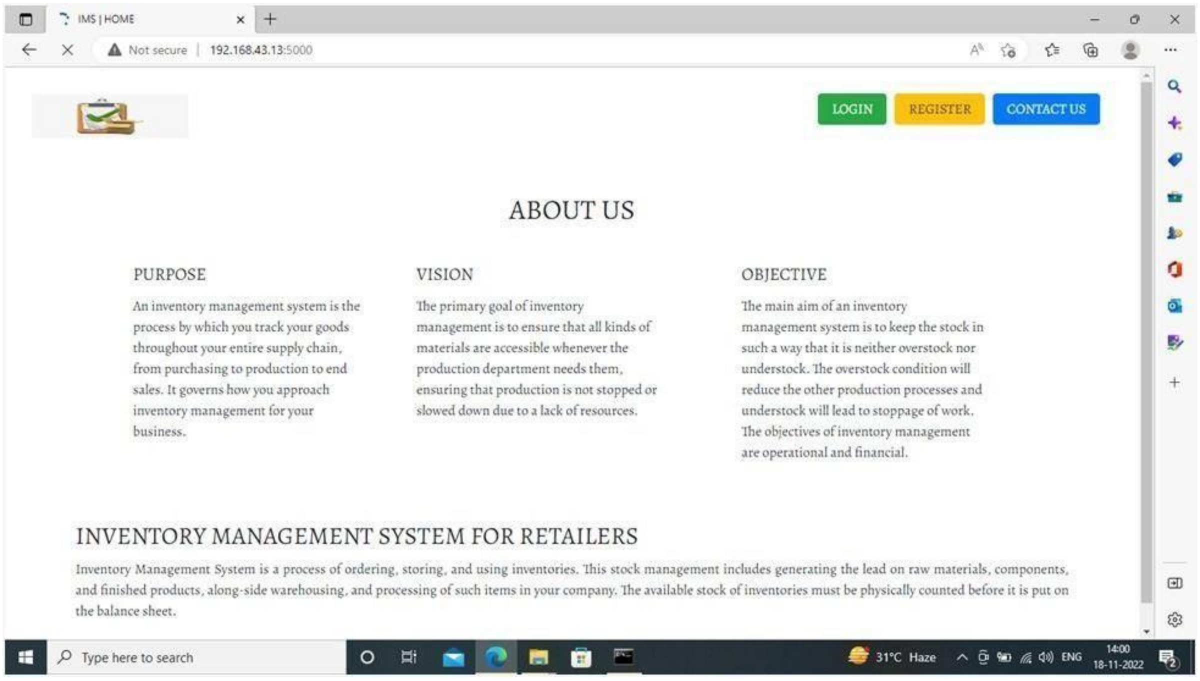
Case 3: Test the recipe sold list box quantity sold list box.

Case 4: Test if the details are updated to the database when requested.

9. RESULT

Inventory Management System for Retailers is developed using Cloud and executed at the level of completed progress.

HOME PAGE



IMS | HOME

Not secure | 192.168.43.13:5000

ABOUT US

PURPOSE

An inventory management system is the process by which you track your goods throughout your entire supply chain, from purchasing to production to end sales. It governs how you approach inventory management for your business.

VISION

The primary goal of inventory management is to ensure that all kinds of materials are accessible whenever the production department needs them, ensuring that production is not stopped or slowed down due to a lack of resources.

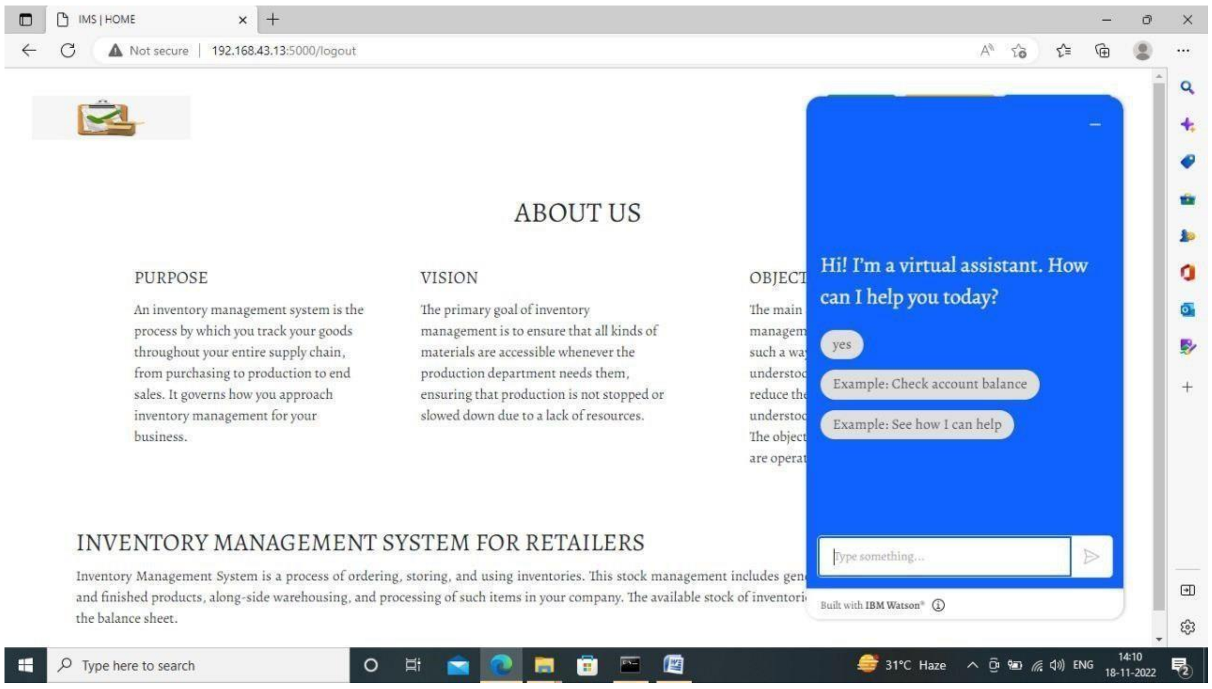
OBJECTIVE

The main aim of an inventory management system is to keep the stock in such a way that it is neither overstock nor understock. The overstock condition will reduce the other production processes and understock will lead to stoppage of work. The objectives of inventory management are operational and financial.

INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

Inventory Management System is a process of ordering, storing, and using inventories. This stock management includes generating the lead on raw materials, components, and finished products, along-side warehousing, and processing of such items in your company. The available stock of inventories must be physically counted before it is put on the balance sheet.

HOMEPAGE WITH VIRTUAL ASSISTANT

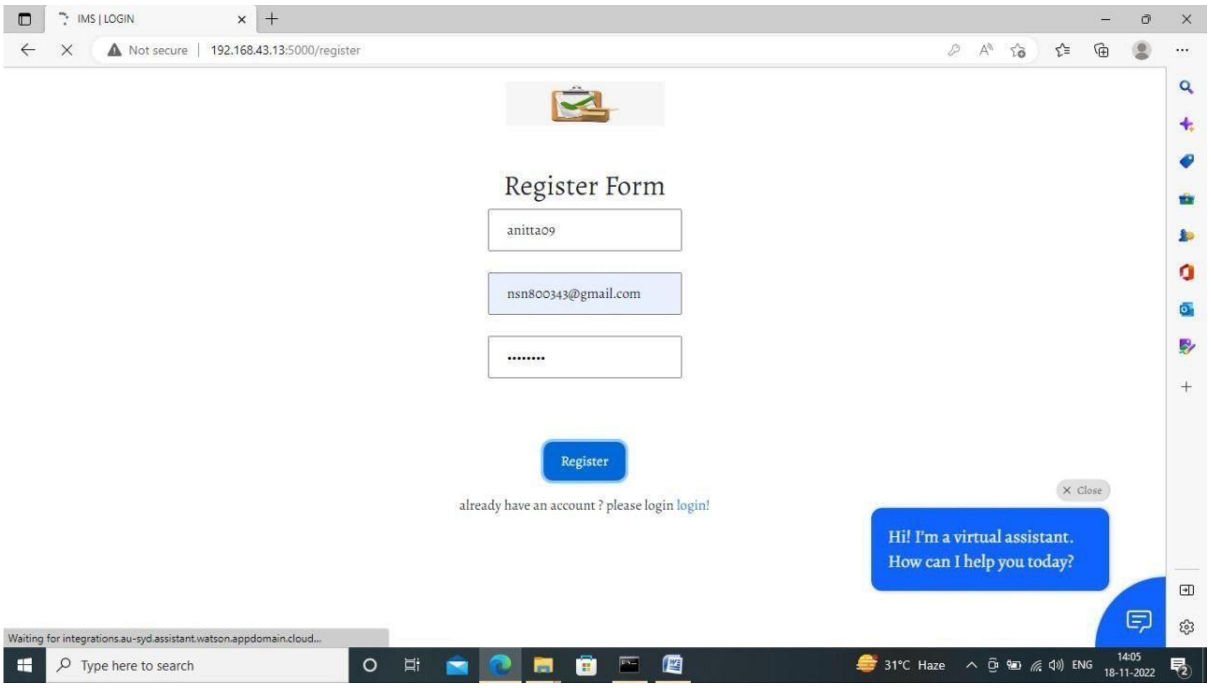


IMS | HOME

Not secure | 192.168.43.13:5000/logout

Hi! I'm a virtual assistant. How can I help you today?

REGISTRATION FORM



IMS | LOGIN

Not secure | 192.168.43.13:5000/register

Register Form

anitta09

nsn800343@gmail.com

••••••••

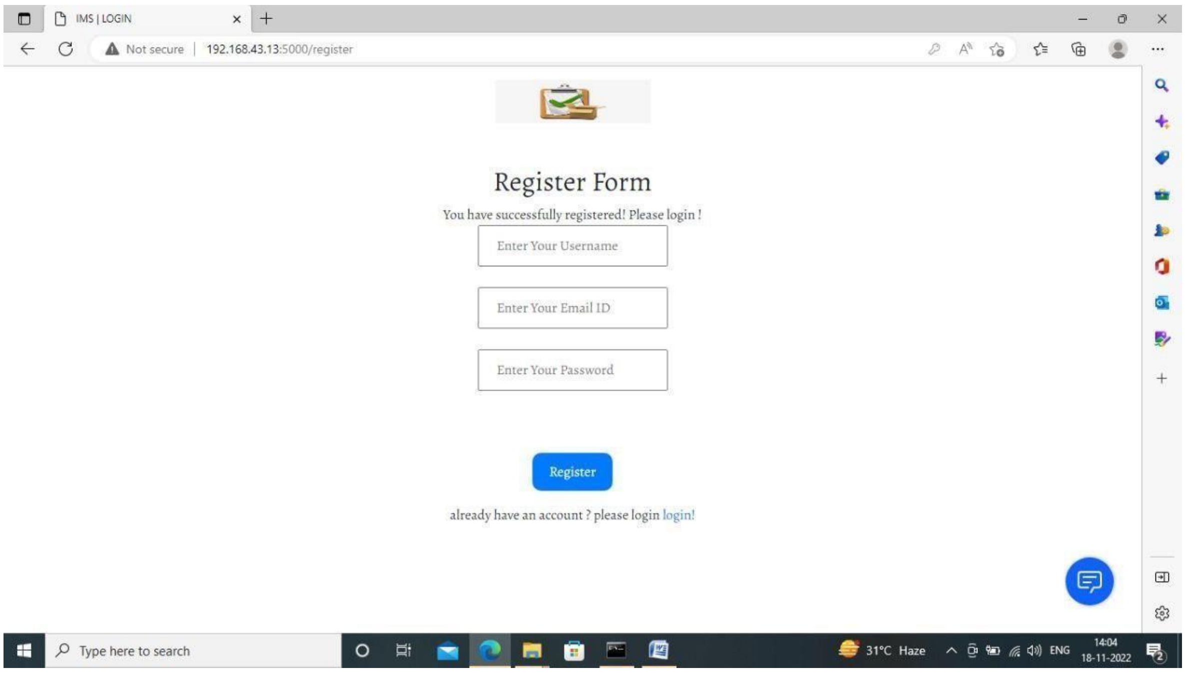
Register

already have an account ? please login login!

Hi! I'm a virtual assistant. How can I help you today?

Waiting for integrations.au-syd.assistant.watson.appdomain.cloud...

REGISTRATION SUCCESSFULLY



IMS | LOGIN

Not secure | 192.168.43.13:5000/register

Register Form

You have successfully registered! Please login !

Enter Your Username

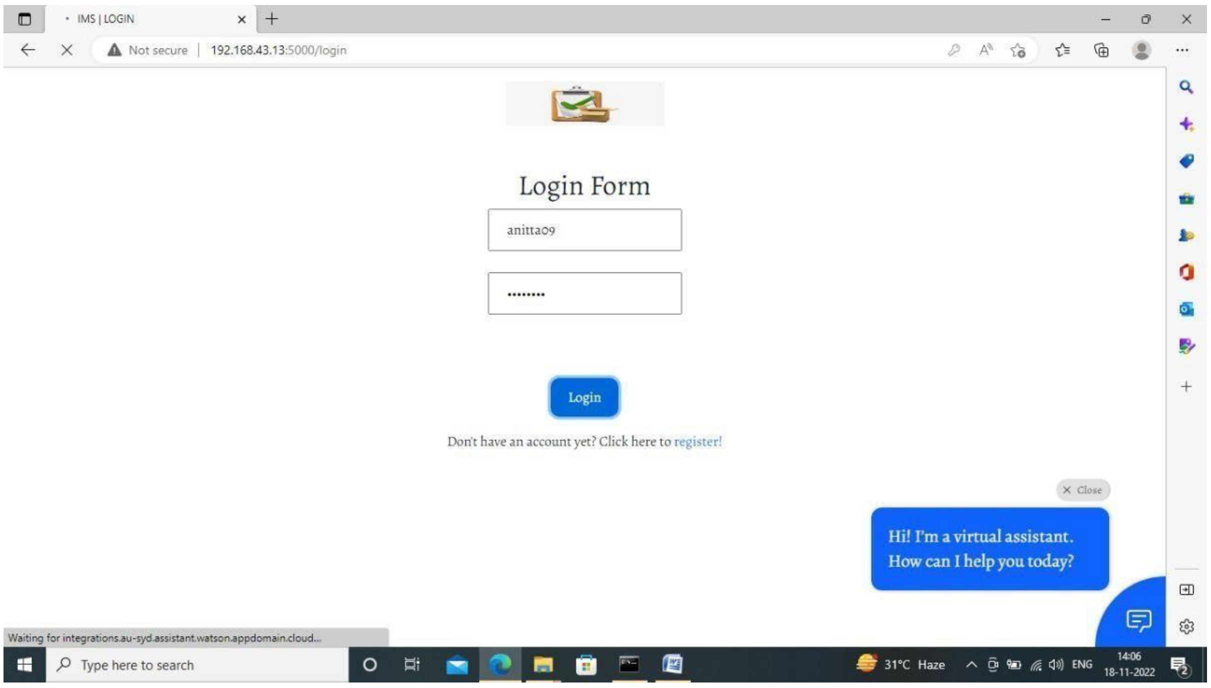
Enter Your Email ID

Enter Your Password

Register

already have an account ? please login login!

LOGIN FORM



IMS | LOGIN

Not secure | 192.168.43.13:5000/login

Login Form

anitta09

••••••••

Login

Don't have an account yet? Click here to register!

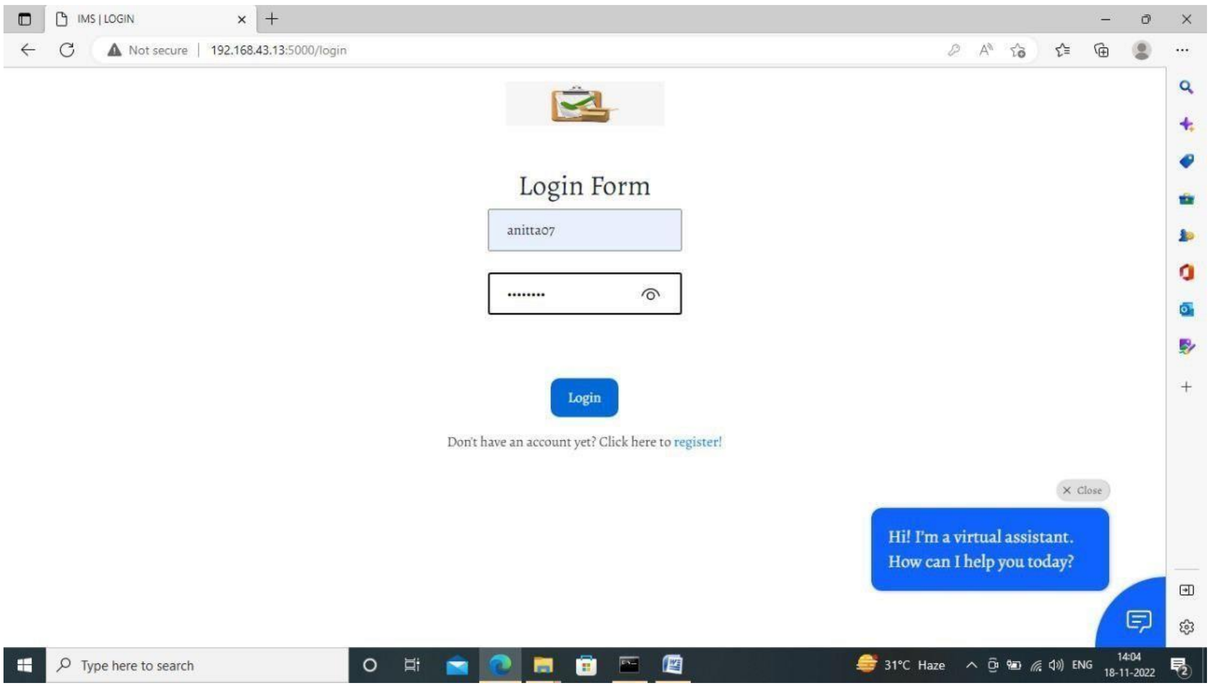
Close

Hi! I'm a virtual assistant. How can I help you today?

Waiting for integrations.au-syd.assistant.watson.appdomain.cloud...

Type here to search

31°C Haze ENG 14:06 18-11-2022



IMS | LOGIN

Not secure | 192.168.43.13:5000/login

Login Form

anitta07

••••••••

Login

Don't have an account yet? Click here to register!

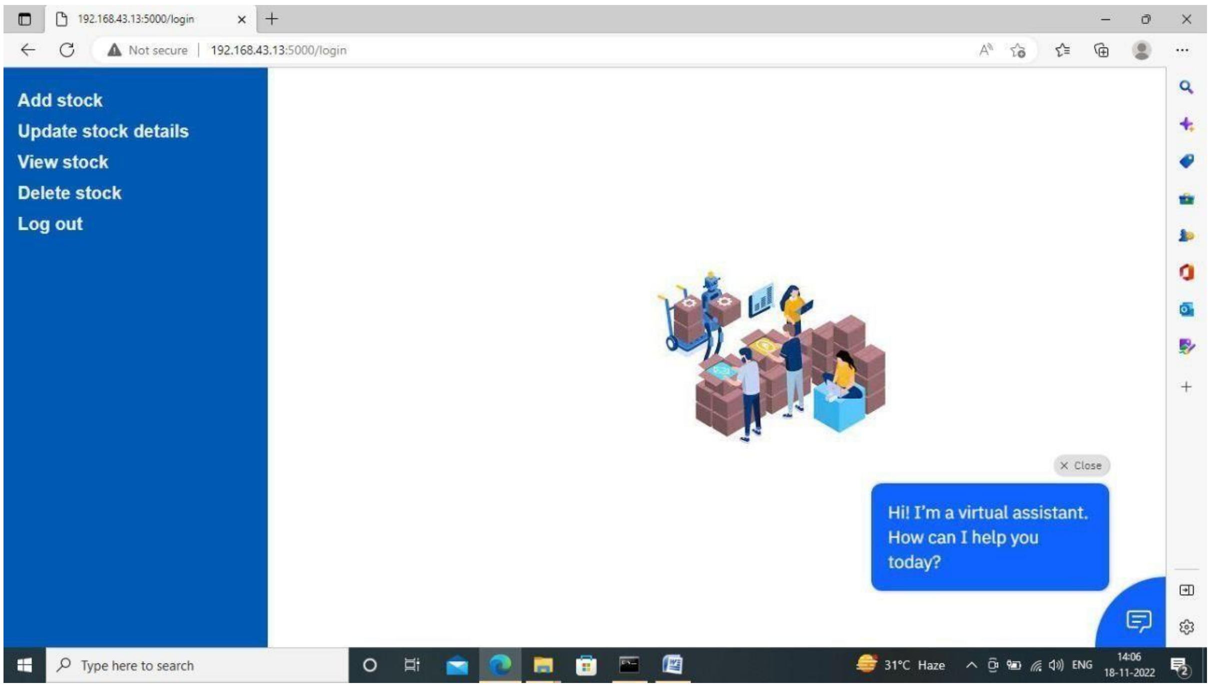
Close

Hi! I'm a virtual assistant. How can I help you today?

Type here to search

31°C Haze ENG 14:04 18-11-2022

DASHBOARD

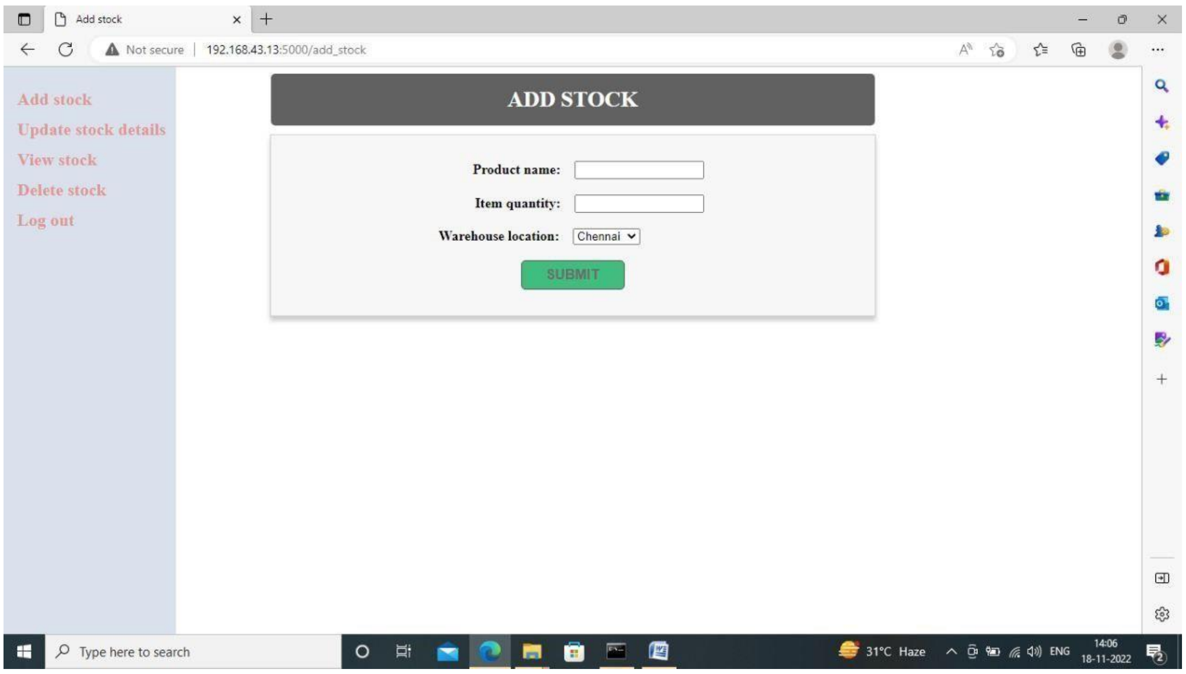


192.168.43.13:5000/login

Add stock
Update stock details
View stock
Delete stock
Log out

Hi! I'm a virtual assistant. How can I help you today?

ADDSTOCK



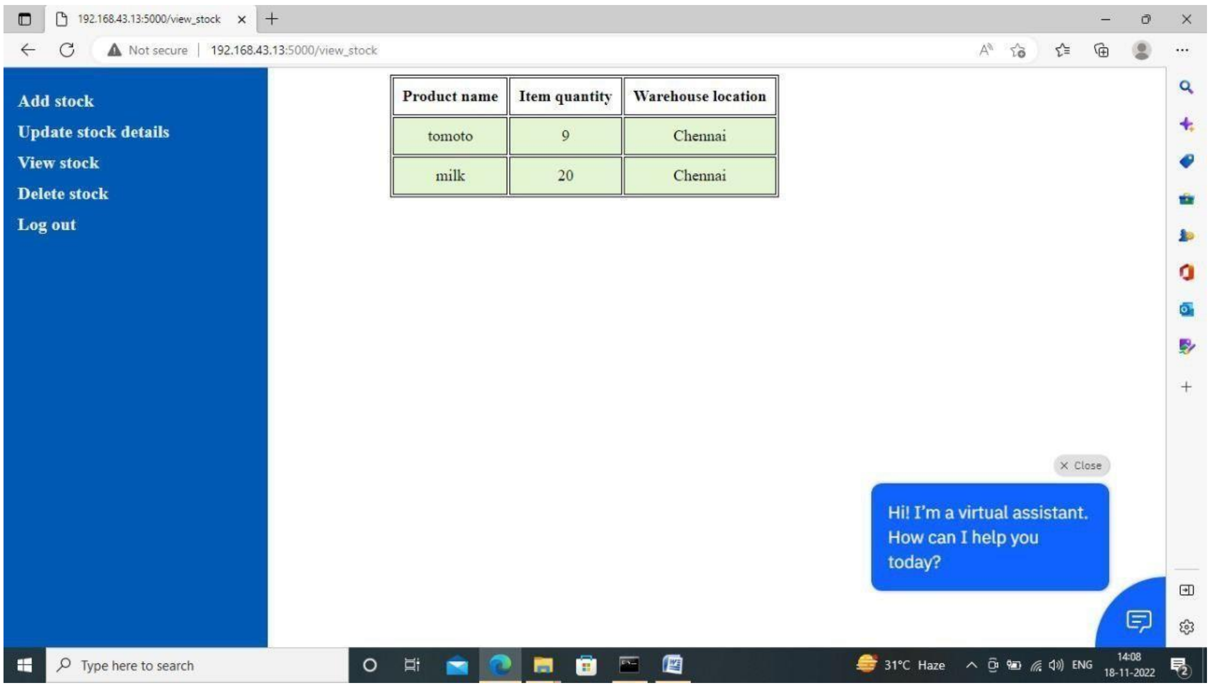
192.168.43.13:5000/add_stock

Add stock
Update stock details
View stock
Delete stock
Log out

ADD STOCK

Product name:
Item quantity:
Warehouse location: Chennai
SUBMIT

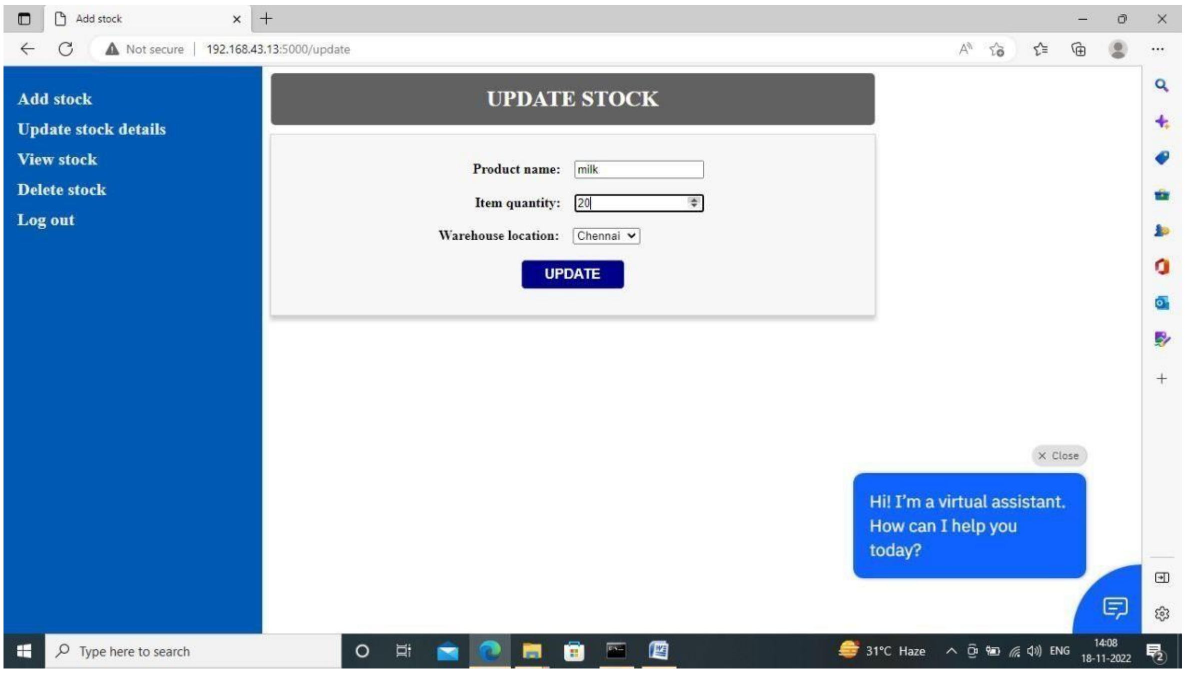
VIEW STOCK



Product name	Item quantity	Warehouse location
tomoto	9	Chennai
milk	20	Chennai

192.168.43.13:5000/view_stock
Add stock
Update stock details
View stock
Delete stock
Log out
Hi! I'm a virtual assistant. How can I help you today?

UPDATE STOCK



UPDATE STOCK

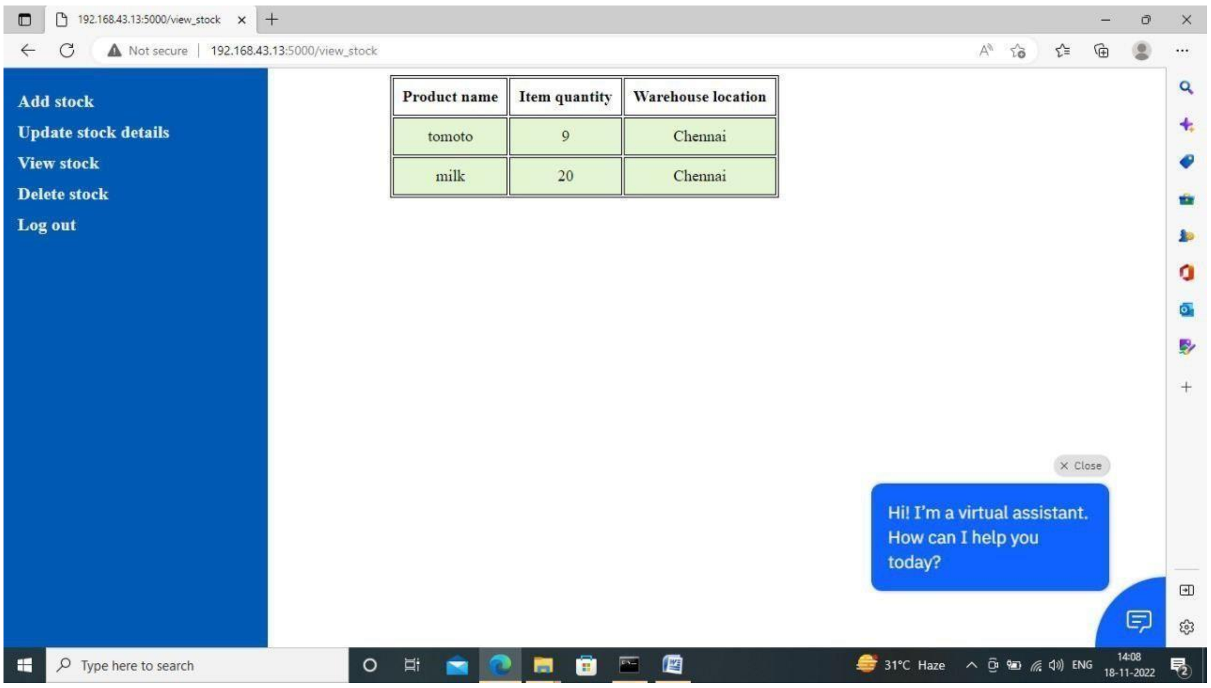
Product name: milk

Item quantity: 20

Warehouse location: Chennai

192.168.43.13:5000/update
Add stock
Update stock details
View stock
Delete stock
Log out
UPDATE
Hi! I'm a virtual assistant. How can I help you today?

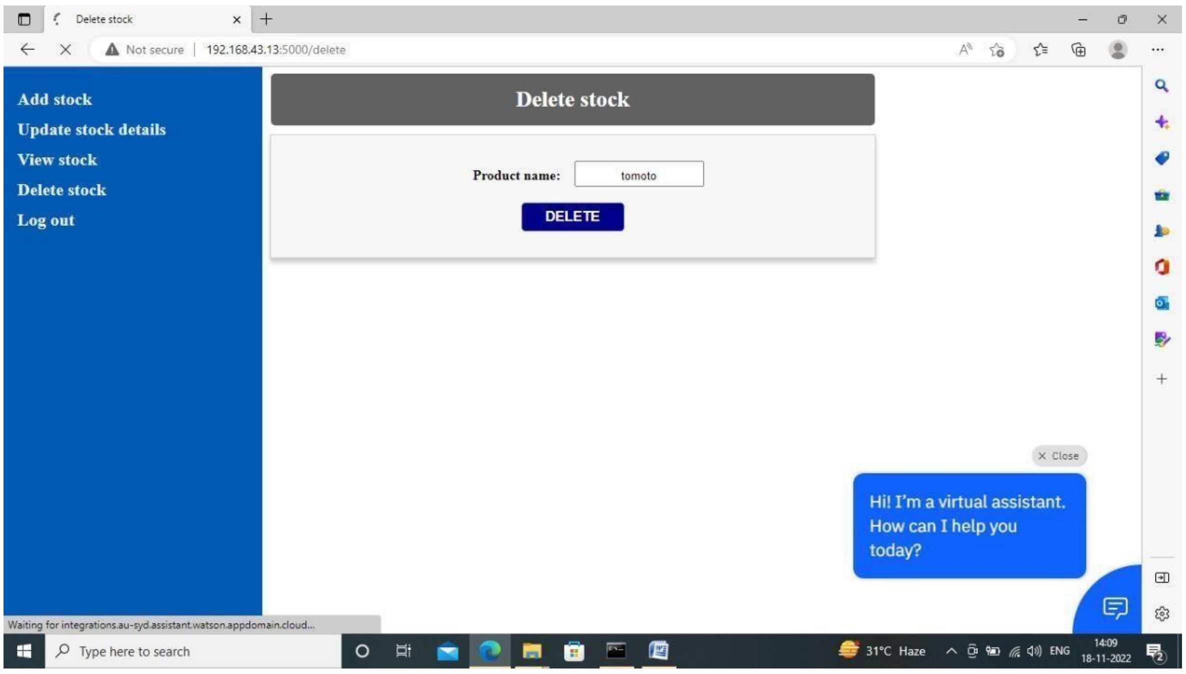
UPDATESTOCKDETAILS



192.168.43.13:5000/view_stock

Product name	Item quantity	Warehouse location
tomoto	9	Chennai
milk	20	Chennai

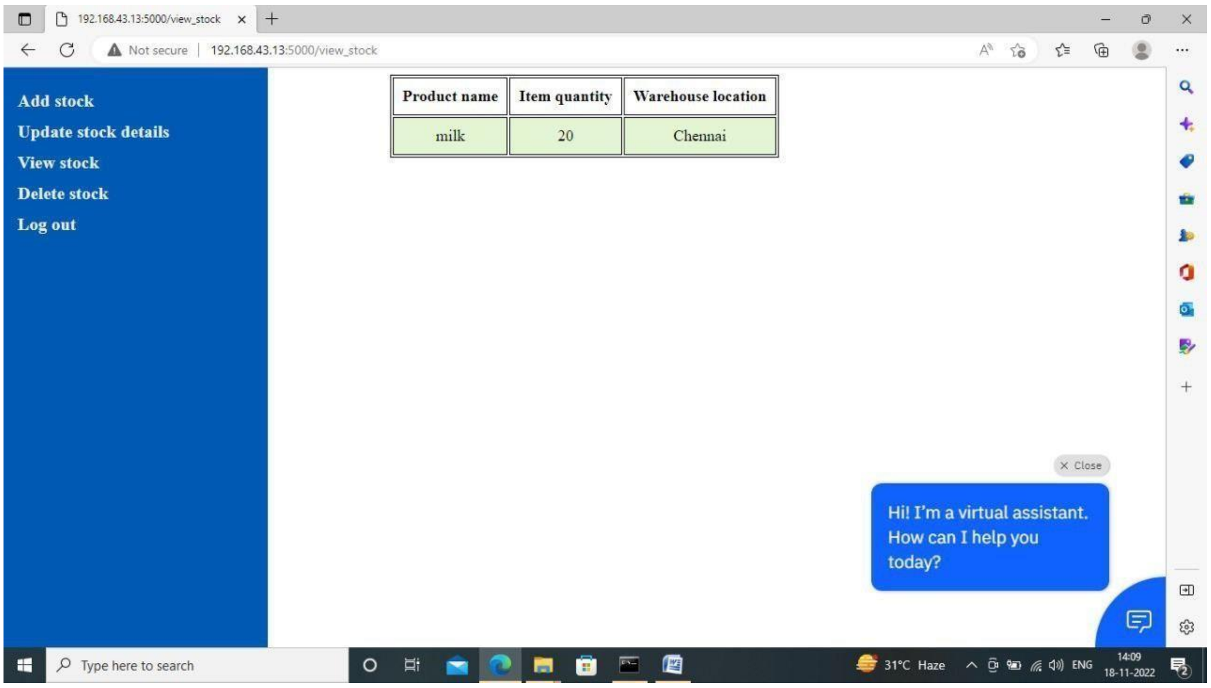
DELETE STOCK



192.168.43.13:5000/delete

Delete stock

Product name: tomoto

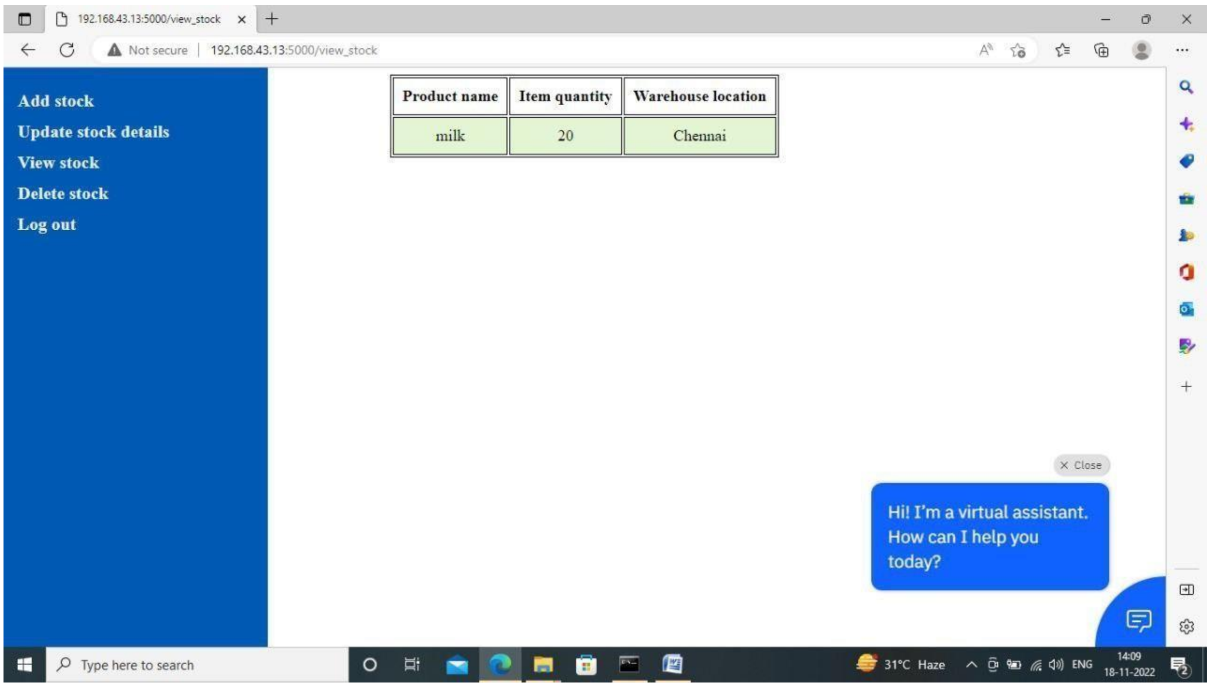


192.168.43.13:5000/view_stock

Not secure | 192.168.43.13:5000/view_stock

Product name	Item quantity	Warehouse location
milk	20	Chennai

LOGOUT



192.168.43.13:5000/view_stock

Not secure | 192.168.43.13:5000/view_stock

Product name	Item quantity	Warehouse location
milk	20	Chennai

9.1. PERFORMANCE METRICS

Inventory Performance is a measure of how effectively and efficiently inventory is used and replenished. The goal of inventory performance metrics is to compare actual on-hand dollars versus forecasted cost of goods sold. Many Lean practitioners claim that inventory performance is the single best indicator of the overall operational performance of a facility. Inventory performance looks at and is measured using either Inventory Days OnHand (DOH) or Inventory Turns.

- **Inventory Days On-Hand:** The number of days it would take to consume current on-hand inventory. Always measure multiple inventory item numbers in terms of currency (i.e. COGS).

- **Inventory Turns**: The number of times inventory is replaced in a year.

10. ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

- It helps to maintain the right amount of stocks
- It leads to a more organized warehouse
- It saves time and money
- Improves efficiency and productivity
- A well-structured inventory management system leads to improved customer retention
- Avoid lawsuits and regulatory fines
- Schedule maintenance

10.2 DISADVANTAGES

- Bureaucracy

- Impersonal touch
- Production problem
- Increased space is need to hold the inventory
- Complexity
- High implementation costs

11. CONCLUSION

Inventory management is a very complex but essential part of the supply chain. An effective inventory management system helps to reduce stock-related costs such as warehousing, carrying, and ordering costs.

12. FUTURE SCOPE

According to Easy Post, 'Companies can reap a 25% increase in productivity, a 20% gain in space usage, and a 30% improvement in stock use efficiency if they use integrated order processing for their inventory system. Advanced mobile applications allow companies to manage their inventory and supply chains effectively.

14.APPENDIX

App.py

```
from flask import Flask, render_template, request, redirect, url_for,
session, flash import ibm_db import sqlite3 as sql import re


app = Flask(name)


app.secret_key = 'a'


conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-
d84a-
4bb0-
85b9ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.
cloud;PORT=322
86;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID
=w jy24066;PWD=3w6H3sui635KMvWX",'','') print(conn)
print("Connecting Successful!!!!!!!!")

@app.route('/')

def homer()  return
render_template('home.html')


@app.route('/login',methods =['GET',
'POST']) def login():
   global userid
msg = ''



   if request.method == 'POST' :
      username = request.form['username']        password =
request.form['password']        sql = "SELECT * FROM users WHERE
username =? AND password=?"        stmt = ibm_db.prepare(conn,
sql)        ibm_db.bind_param(stmt,1,username)
```

```
ibm_db.bind_param(stmt,2,password)        ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)        print (account)        if
account:
          session['loggedin'] = True            session['id']
= account['USERNAME']            userid=
account['USERNAME']            session['username'] =
account['USERNAME']            msg = 'Logged in
successfully !'            return
render_template('dashboard.html', msg = msg)
else:
          msg = 'Incorrect username / password
!'    return render_template('login.html', msg
= msg)


 @app.route('/register', methods =['GET',
'POST']) def registet():     msg = ''     if
request.method == 'POST' :
       username = request.form['username']
email = request.form['email']        password =
request.form['password']        sql = "SELECT *
FROM users WHERE username =?"        stmt =
ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt)        account =
ibm_db.fetch_assoc(stmt)        print(account)
if account:

          msg = 'Account already exists !'        elif
not re.match(r'[^@]+@[^@]+\.[^@]+', email):

          msg = 'Invalid email address !'
elif not re.match(r'[A-Za-z0-9]+',
username):

          msg = 'name must contain only characters and
numbers !'        else:

          insert_sql = "INSERT INTO  users VALUES (?, ?,
?)"           prep_stmt = ibm_db.prepare(conn,
insert_sql)           ibm_db.bind_param(prep_stmt, 1,
username)           ibm_db.bind_param(prep_stmt, 2,
email)           ibm_db.bind_param(prep_stmt, 3,
password)           ibm_db.execute(prep_stmt)
msg = 'Please fill out the form !'        if
request.method == 'POST':

        msg = 'You have successfully registered! Please login
!'    return render_template('register.html', msg = msg)


@app.route('/add_stock',methods=['GET','POST'])
```

```
def add_stock():
   msg=''     if
request.method == "POST":
      prodname=request.form['prodname']
quantity=request.form['quantity']
warehouse_location=request.form['warehouse_loc
ation']         sql='SELECT * FROM product WHERE
prodname =?'        stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,prodname)
ibm_db.execute(stmt)
acnt=ibm_db.fetch_assoc(stmt)        print(acnt)
        if acnt:
msg='Product already exits!!'
else:
          insert_sql='INSERT INTO product VALUES
(?,?,?)'            pstmt=ibm_db.prepare(conn,
insert_sql)
ibm_db.bind_param(pstmt,1,prodname)
ibm_db.bind_param(pstmt,2,quantity)
ibm_db.bind_param(pstmt,3,warehouse_location)
ibm_db.execute(pstmt)           msg='You have
```

```
successfully added the products!!'            return
render_template("dashboard.html")


else:

        msg="fill out the form first!"        return
render_template('add_stock.html',meg=msg)
@app.route('/delete_stock',methods=['GET','P
OST']) def delete_stock():
if(request.method=="POST"):

        prodname=request.form['prodname']
sql2="DELETE FROM product WHERE prodname=?"
stmt2 = ibm_db.prepare(conn, sql2)
ibm_db.bind_param(stmt2,1,prodname)
ibm_db.execute(stmt2)


        flash("Product Deleted", "success")


        return render_template("dashboard.html")

@app.route('/update_stock',methods=['GET','PO
ST']) def update_stock():

    mg=''     if
request.method == "POST":
        prodname=request.form['prodname']
quantity=request.form['quantity']
quantity=int(quantity)        print(quantity)
print(type(quantity))
warehouse_location=request.form['warehouse_loc
ation']         sql='SELECT * FROM product WHERE
prodname =?'        stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,prodname)
ibm_db.execute(stmt)
acnt=ibm_db.fetch_assoc(stmt)

        print(acnt)
          if acnt:            insert_sql='UPDATE product SET
quantity=?,warehouse_location=? WHERE prodname=? '

            pstmt=ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(pstmt,1,quantity)
ibm_db.bind_param(pstmt,2,warehouse_location
)            ibm_db.bind_param(pstmt,3,prodname)
ibm_db.execute(pstmt)            mg='You have
successfully updated the products!!'
limit=5 print(type(limit))
if(quantity<=limit):
```

```
            ("Please update the quantity of the product {}, Atleast {}
number of pieces must be added!".format(prodname,10))
return render_template("dashboard.html",meg=mg)


else:

          mg='Product not found!!'



else:

       msg="fill out the form first!"         return
render_template('update_stock.html',meg=msg)


@app.route('/view_stoc
k') def view_stock():


   sql = "SELECT * FROM product"
   stmt = ibm_db.prepare(conn, sql)
result=ibm_db.execute(stmt)
print(result)
```

```
    products=[]     row =
ibm_db.fetch_assoc(stmt)
print(row)     while(row):


products.append(row)         row
= ibm_db.fetch_assoc(stmt)
print(row)
products=tuple(products)
print(products)


    if result>0:
        return render_template('view.html', products =
products)     else:

        msg='No products found'         return
render_template('view.html', msg=msg)




@app.route('/delete')
def delete():

    return render_template('delete_stock.html')
```

```
@app.route('/updat
e') def update():
    return render_template('update_stock.html')



@app.route('/logout')

def logout():
   session.pop('loggedin', None)
session.pop('id', None)
session.pop('username', None)
return
render_template('home.html')


if name == 'main':
   app.run(host='0.0.0.0')
```

home.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-
scale=1"> <link   rel="stylesheet"
```

```
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css
/font-awesome.min.css">


<style> body {   font-
family: "Lato", sans-serif;


}

/* Fixed sidenav, full height */
.sidenav
{height:
100%;
width:
300px;
position:
fixed;   z-
index: 1;
top: 0;   left:
0;
  background-color:
#0059b3;   overflow-x:
hidden;   padding-top:
20px;
}

/* Style the sidenav links and the dropdown button */
.sidenav a{   padding:
6px 8px 6px 16px;   text-
decoration: none;   font-
size: 20px;   color:
rgb(239, 239, 239);
display: block;   border:
none;   background:
none;   width: 100%;
  text-align:
left;   cursor:
pointer;
outline:
none;
}

/* On mouse-over */
.sidenav
a:hover{color:
#111;
}
```

```
/* Some media queries for responsiveness */
@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}
</style>
</head>
<body>


> Deeps:
<div class="sidenav">


   <a href="{{url_for('add_stock') }}"><strong>Add stock<strong></a>
   <a href="{{url_for('update') }}"><strong>Update stock
details<strong></a>
   <a href="{{url_for('view_stock') }}"><strong>View
stock<strong></a>
    <a href="{{url_for('delete')}}"><strong>Delete stock<strong></a>
  <a href="{{url_for('logout') }}"><strong>Log out<strong></a>
    </div>
    <nav>
      <script>        window.watsonAssistantChatOptions =
{ integrationID: "4bd6f313-33d4-4e87-8825-22b90b8e3c2c", // The
IDof this integration.       region: "au-syd", // The region your
```

```
integration is hosted in.          serviceInstanceID: "60e1396a-421f-
4091-b39a-a23a546843e8", // The ID of your service instance.

          onLoad: function(instance) { instance.render(); }
        };
        setTimeout(function(){          const
t=document.createElement('script');
t.src="https://web-

chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +

"/WatsonAssistantChatEntry.js";
document.head.appendChild(

 });
      </script>
    </nav>


   </body>
</html>
```

GITHUB:

https://github.com/IBM-EPBL/IBM-Project-38526-1660382075

DEMOLINK:

https://1drv.ms/v/s!Au3985NsWb5CgQIUPkrXBiC7qaLV?e=62Fjjp