
**HX8001- PROFESSIONAL READINESS FOR
INNOVATION,EMPLOYABILITY AND ENTREPRENEURSHIP**

INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

TEAM ID : PNT2022TMID45016

TEAM MEMBERS:

ARUN.P

DHARANEESWARAN.S

JAYASHREE.K

WESLIN JOEL RAJ.A

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

JJ COLLEGE OF ENGINEERING AND TECHNOLOGY

TRICHY

ANNA UNIVERSITY : CHENNAI 600 025

NOV-DEC 2022

TABLE OF CONTENT

CHAPTER	CONTENTS	PAGE NO
1	INTRODUCTION	4
	1.1 PROJECT OVERVIEW	4
	1.2 PURPOSE	5
2	LITERATURE SURVEY	6
	2.1 EXISTING PROBLEM	6
	2.2 REFERENCES	6
	3.4 PROBLEM STATEMENT DEFINITION	7
3	IDEATION & PROPOSED SOLUTION	8
	3.1 EMPATHY MAP CANVAS	8
	3.2 IDEATION & BRAINSTROMING	9
	3.3 PROPOSED SOLUTION	10
	3.4 PROBLEM SOLUTION FIT	11
4	REQUIREMENT ANALYSIS	12
	4.1 FUNCTIONAL REQUIREMENT	12
	4.2 NON-FUNCTIONAL REQUIREMENTS	13
5	PROJECT DESIGN	14
	5.1 DATA FLOW DIAGRAMS	14
	5.2 SOLUTION&TECHNICAL ARCHITECTURE	15
	5.3 USER STORIES	15
6	PROJECT PLANNING & SCHEDULING	16
	6.1 SPRINT PLANNING & ESTIMATION	16
	6.2 SPRINT DELIVERY SCHEDULE	17

	6.3 REPORTS FROM JIRA	20
7	CODING & SOLUTIONING	22
	7.1 FEATURE1	22
	7.2 FEATURE2	22
	7.3 DATABASE SCHEMA	23
8	TESTING	24
	8.1 TEST CASES	24
	8.2 USER ACCEPTANCE TESTING	25
9	RESULTS	26
	9.1 PERFORMANCE METRICS	33
10	ADVANTAGES & DISADVANTAGES	33
11	CONCLUSION	35
12	FUTURE SCOPE	36
13	APPENDIX	36
	SOURCE CODE	36
	GITHUB & PROJECT DEMO LINK	46

1.INTRODUCTION

Inventory management information system is high performance software, which speed up the business operation of the organization . Every organization , which deals with the raw materials, put its great effort in the efficient utilization of its raw, material according to its need and requirement . The organization has to perform number of tasks and operations in order to run its business in manual system .

For example From NaavebUROM

- Estimation of new raw material required.
- Preparation of purchase order.

Preparation of inward sale invoice This Software “Inventory Management System” , is used for recording the information about the day to day transaction of stock of an organization. It stores purchase information of the products with credit/debit information form the supplier. Similarly, it stores sales information with credit/debit about the customer. If a product is purchased, then the related information is stored in stocks , that is , stocks are up to date. Another part I it prepare sales report after product it sold. in the sales information, the information about who sold the product is also kept, so there is no problem for misunderstandings in future.

1.1. PROJECT OVERVIEW

Inventory management system is an application which is helpful for business operate. Inventory management is a challenging problem area in supply chain management. Companies need to have inventories in warehouses in order

to fulfil customer demand, meanwhile these inventories have holding costs and this is frozen fund that can be lost. Therefore, the task of inventory management is to find the quantity of inventories that will fulfil the demand, avoiding overstocks. This paper presents a case study for the assembling company on inventory management. It is proposed to use inventory management in order to decrease stock levels and to apply an agent system for automation of inventory management. processess Inventory management system (IMS) use for a departmental store.

1.2. PURPOSE:

A case study at ‘Guckenheimer’ (an on-site corporate restaurant management and catering company) cited issues regarding a basic resources requirement list that has to be maintained manually by the staff. To keep track of their inventory levels they have to calculate a list of the groceries utilized during a course of time, calculate and analyze the requirements for the future, and place their next order to the vendors if needed. This process takes up a lot of time and human effort, and is also prone to human error. This poses a problem of a situation that the staff at ‘Guckenheimer,’ as well as many other restaurants faces. It takes up a lot of time to manually keep track of sales and place correct orders to vendors, wasting useful labor in trivial works. A product which would assist in tackling the above mentioned problems would prove to be fruitful to clients such as ‘Guckenheimer’ and similar enterprises as this product would help convert the unproductive time to something more useful, by removing the unnecessary error prone complications and efforts.

2.LITERATURE SURVEY

2.1. EXISTING PROBLEM

Products are considered as the business resources for the organization. This includes managing the product with appropriate way to review any time as per the requirement. Therefore it is important to have a computer based IMS which has the ability to generate reports, maintain the balance of the stock, details about the purchase and sales in the organization. Before developing this application we came up with 2Inventory Management System existing in the market, which helps to give the knowledge for the development of our project. These application software are only used by the large organization but so we came up with the application which can be used by the small company for the management of their stock in the production houses. After analyzing the other inventory management system we decided to include some of common and key features that should be included in every inventory management system. So we decided to include those things that help the small organization in away or other.

2.2. REFERENCES

We have referred several documentations for the purpose of development phases.

[1]<https://www.camcode.com/asset-tags/what-is-an-inventory-managementsystem/>

[2] Jimmy Wales, online encyclopedia WiKipedia ,
<http://www.wikipedia.org>

[3] James Gosling. Java (Programming Language) ,
<http://www.java.com>

[4] Names Allaire, Netbeans-Fully-featured Java IDE,
<http://www.netbeans.org>

[5] James Gosling , Welcome to java world.com: how-to feature and columns by Java expert; news; Java applets; sample code ; tips , <http://www.javaworld.com> [6] Pressman, Roger S.

“Software Engineering A Practitioner” Approach [7]

John Osborn , JavaBeans:

Developing Component Software in Java [8]

Doug Lea Concurrent Programming in Java:

Design Principles and Pattern, Addison-Wesley , November,1996

[9] Design Report, submitted 9th November 2012.

https://skydrive.live.com/redir?resid=2CEDE9F7F5F99604!196&authkey=!AO5I_ghTCML6xAk8

[10] Testing Document, submitted 26th November 2012.

<https://www.camcode.com/asset-tags/what-is-an-inventorymanagement->

2.3. PROBLEM STATEMENT DEFINITION`

The problem statement aims to make desktop application for retailers and to track all areas of IMS like purchase details, sales details, stock management. The application helps the retailer to have complete insights about the products stored in the inventory and can manage them flexibly.

I am A retailer	I'm trying to Find the product details	But It's hard and difficult	Because It takes long time	Which makes me feel Feel irritated and frustrated on continuous validation
--------------------	---	--------------------------------	-------------------------------	---

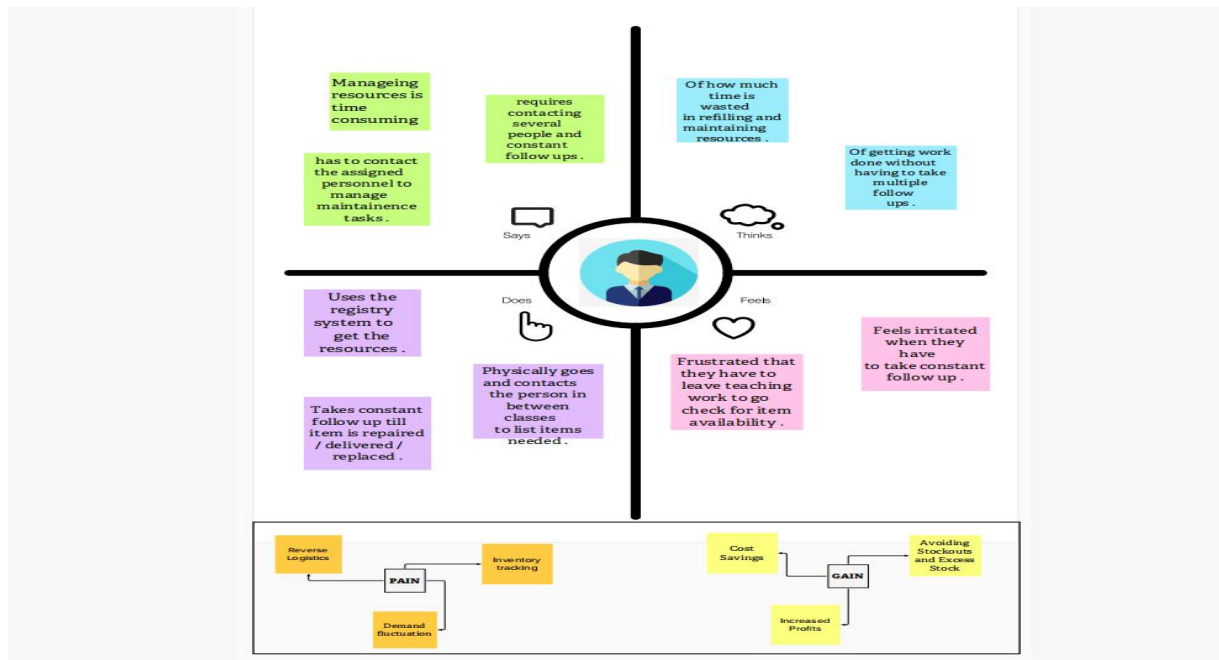
Problem statement(PS)	I am (customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Retailer	Find the product counts in the stock	It's hard	It takes more time	Tired
PS-2	Retailer	Calculate the bill for transportation purpose	It's hard	It take long time	Uninterested
PS-3	Retailer	Find the customer's review	It's hard to gathering information	I don't have enough contacts	Disappointed
PS-4	Retailer	Maintain the ledger	It's difficult to secure	It may be lost	Afraid
PS-5	Retailer	Find the high demand	It's difficult to calculate	It takes more	Challenging

3.IDEATION & PROPOSED SOLUTION

We have analyzed different systems and proposed an ideation phase of our developed management system.

3.1. EMPATHY MAP CANVAS

An empathy map canvas helps brands provide a better experience for users by helping teams understand the perspectives and mindset of their customers. Using a template to create an empathy map canvas reduces the preparation time and standardizes the process so you create empathy map canvases of similar quality.



3.2. IDEATION & BRAINSTORMING

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



3.3. PROPOSED SOLUTION PREPROCESSING PHASES

Home:

This first module manages Home Screen Which is Provide A Home Page of my Software. After clicking home button . button will provide Welcome Screen of the Software etc.

Sales

This is Provide Sales information And Sales Page it is contain sales_id, Product_code , Product_name , Quantity, Revenue, Sold by etc

Suppliers

Suppliers page contain suppliers details and its hold basic value with attributes it is provide a suppliers code, full name ,location ,phone etc Products: It is hold the details of product with product code, product name, cost price selling price brand etc.

Purchase

This is contain detail about purchase . It will provide purchase screen which is hold some value like purchase id ,product code ,product name ,quantity ,total

cost etc And Each page has refresh facilities And search facilities and Direct input value interface etc .

Edit

Many Module contains Edit facilities Which has control of editing value from data base directly and insert new value etc.

Clear and Delete

Clear and delete is provide advance facilities of this software Because it is Provide a deletion and clear data process etc

In the proposed system, all the business operations will be automated. Some of the features which the new system will provide are Auto generation of Daily Demand report, Auto generation of Purchase Order of various raw materials. As everything is auto generated, the production delays are avoided. It makes the system more secure as only authenticated users can access the system. Also, there are privileges in which we can authorize a particular user for accessing system or particular modules of the application.

3.4. PROBLEM SOLUTION FIT

- To develop a system that will enhance the monitoring of the sales and inventory
- To develop a module that can generate monthly sales and inventory report.
- To develop security in terms of keeping the records of the inventory

- To develop a system that can monitor the stocks inventory in a fast and efficient manner.
- To accurately record, compute and produce a report of sales.

4.REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

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,LIFO 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

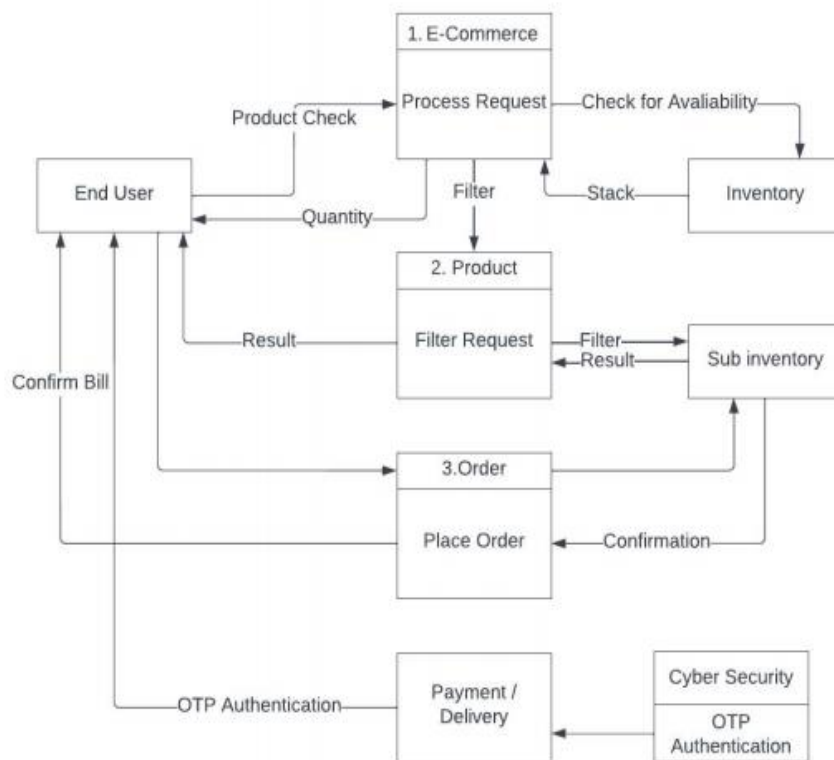
Non-functional Requirements:

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.DATAFLOWDIAGRAMS



5.2. SOLUTION & TECHNICAL ARCHITECTURE

- There was no efficient solution available in many companies these days.
 - Every process was based on paper work.
- Human fault rate were high.
- Tracing the inventory losses were not possible
- There was no efficient login system.
- After the computer age, every process is started to be integrated into computer environment.
- Now, we have qualified technology to implement new solution to these problems.



5.3.USER STORIES

A user story is the smallest unit of work in an agile framework. It's an end goal, not a feature, expressed from the software user's perspective.

User Stories

Use the below template to list all the user stories for the product.

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 register for the application through Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can log into the application by entering email & password	High	Sprint-1
	Request	USN-6	Request an item	Happy to find ease to use	High	Sprint-2
	Search	USN-7	Search an item	Easy to find an item	High	Sprint-3
	Orders	USN-8	As an user can place the order	User can receive a confirmation message	High	Sprint-3
	Payments	USN-9	Make a payment for purchasing	I can pay and get confirmation bill	High	Sprint-4
	Monitoring	USN-10	Monitoring quantity of the items	I can update how much of item needed	High	Sprint-4
	Feedback	USN-11	Users can share their experience about the items	users friendly customer support	High	Sprint-5

6. PROJECT PLANNING & SCHEDULING

6.1. SPRINT PLANNING & ESTIMATION

Sprint 1:

- 1.We created a Flask Project.
- 2.Added all the routes needed for our project
- 3.Created Tables in IBM Cloud.

Sprint 2:

1.We added all the html templates needed for our project.

2.We styled those pages using CSS and Bootstrap

3.We wrote Queries to connect IBM Cloud Database

4.Finished all the Fetching and Posting Stuff of IBM Cloud Database Integration.

Sprint 3:

1.Integration of Send grid into our application

Sprint 4:

1.Deploying the application using Docker and Kubernetes

6.2. SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Initial Registration by customer	USN-1	As a user, I can register for the application by entering my email or number, and password, and confirming my password.	3	High	1. P.Arun 2. S.Dharaneeswaran
Sprint-1	Confirmation for registering	USN-2	As a user, I can receive my confirmation toghemail or number once I have registered for the application.	3	Medium	1. P.Arun 2. K.Jayashree
Sprint-1	Initial Login	USN-3	As a user, I can log in to the authorized account by entering the registered email or number and password.	2	High	1. P.Arun 2. K.Jayashree 3. A.Weslin Joel Raj
Sprint-1	Viewing Dashboard	USN-4	As a user, I can view my account details which I have given during my registration process.	3	High	1. P.Arun 2. S.Dharaneeswaran
Sprint-2	Adding products to cart and order confirmation.	USN-5	As a user, I can view the available products and it to the cart and once the cart is filled I can confirm my order.	2	High	1. K.Jayashree 2. A.Weslin Joel Raj.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Feedback system	USN-6	As a user, I can give the feedback about the product and/or the service and chat with the serviced provider.	2	Medium	1. P.Arun 2. S.Dharaneeswaran
Sprint-2	Registration for retailer	USN-7	As a user, I can register for the application by entering my email, and password, and confirming my password.	2	High	1. P.Arun 2. K.Jayashree 3. A.Weslin Joel Raj 4. S.Dharaneeswaran
Sprint-2	Confirmation	USN-8	As a user, I can receive my confirmation email once I have registered for the application	2	Medium	1. P.Arun 2. K.Jayashree
Sprint-2	Login	USN-9	As a user, I can log in to the authorized account by entering the registered email and password	2	High	1. P.Arun 2. A.Weslin Joel Raj
Sprint-3	Retailers Dashboard	USN-10	As a user, I can view the products that are currently available.	8	High	1. P.Arun 2. K.Jayashree 3. A.Weslin Joel Raj 4. S.Dharaneeswaran
Sprint-4	Stocks update	USN-11	As a user, I can restore the products which are not available in the inventory and restore the inventory details.	2	Medium	1. P.Arun 2. S.Dharaneeswaran
Sprint-4	Stock Alert	USN-12	Alerting the retailer when the stock is low on quantity by using SendGrid.	3	High	1. P.Arun 2. K.Jayashree 3. A.Weslin Joel Raj
Sprint-4	Admin Login	USN-13	As an admin, I can modify and verify the regular and customized users.	2	Medium	1. P.Arun 2. A.Weslin Joel Raj

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Dashboard and updation.	USN-14	As a user, I can change the UI & update features	2	Medium	1. P.Arun 2. K.Jayashree

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	8	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	7	6 Days	14 Nov 2022	19 Nov 2022	9	19 Nov 2022

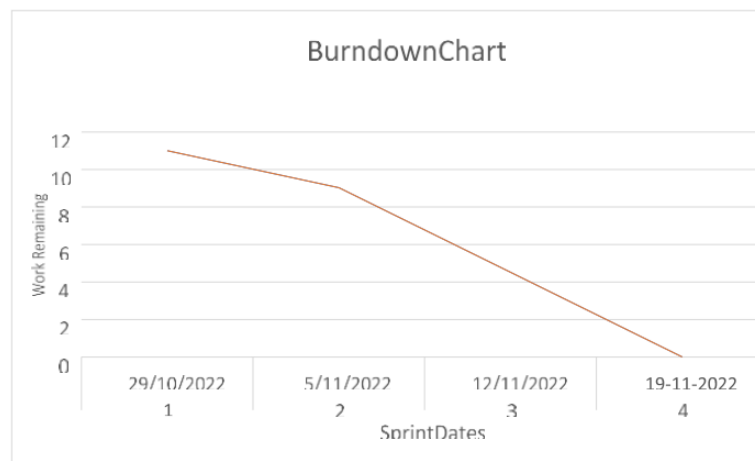
Velocity:

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

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

Our velocity should be:

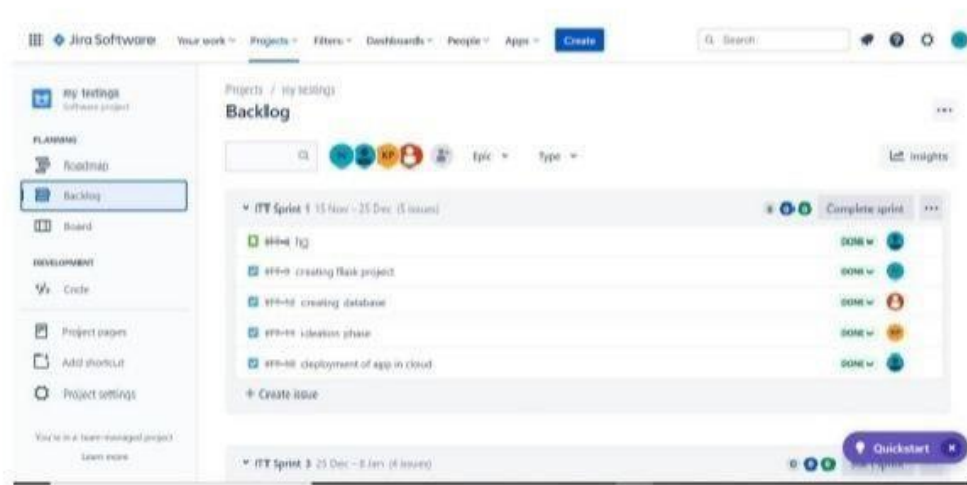
$$AV \frac{(11+10+8+9)}{24} = \frac{38}{24} = 1.58$$



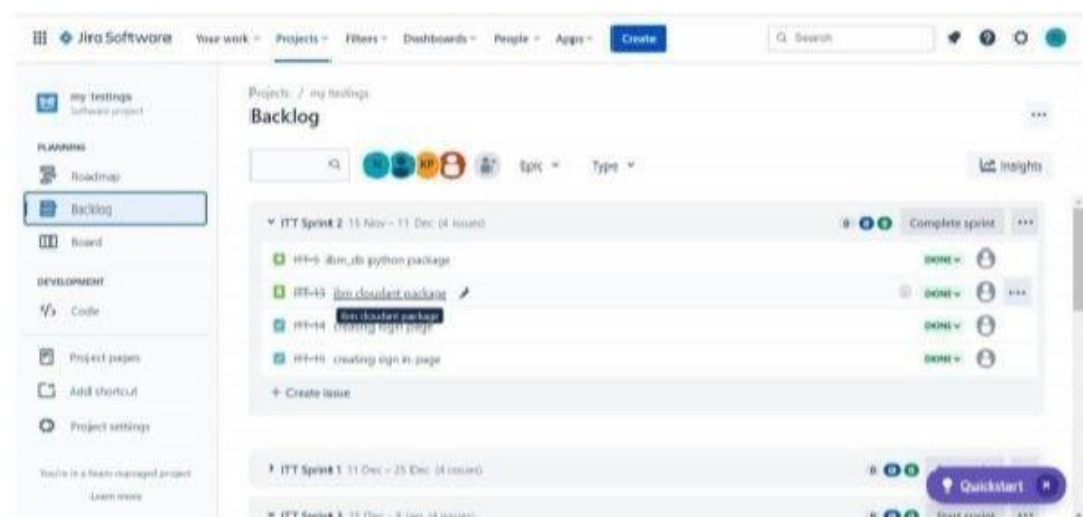
6.3. REPORTS FROM JIRA

IT organizations have the challenge of ensuring system uptime, supporting users, and managing inventory of both hardware and software. IT teams gain significant efficiencies when one tool can support multiple business operations. According to Gartner, mastering the discipline of effective asset management is a huge cost savings for companies.

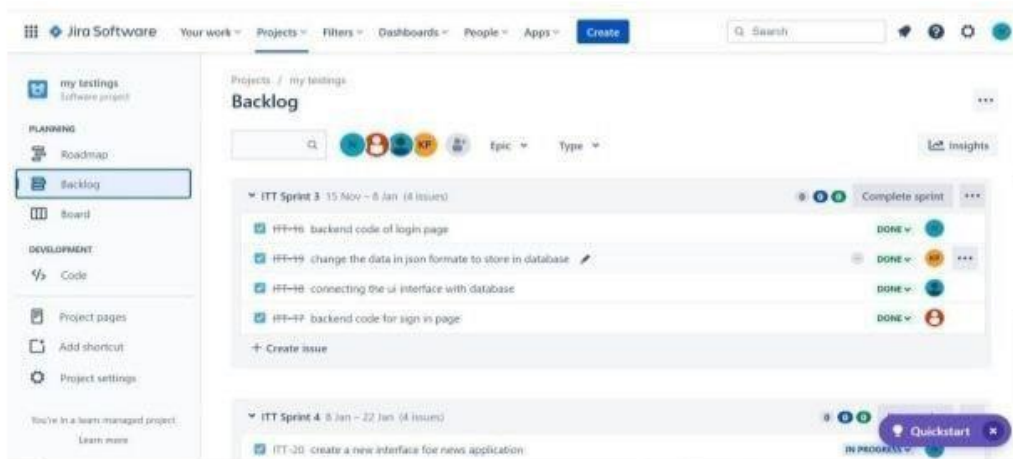
Sprint-1



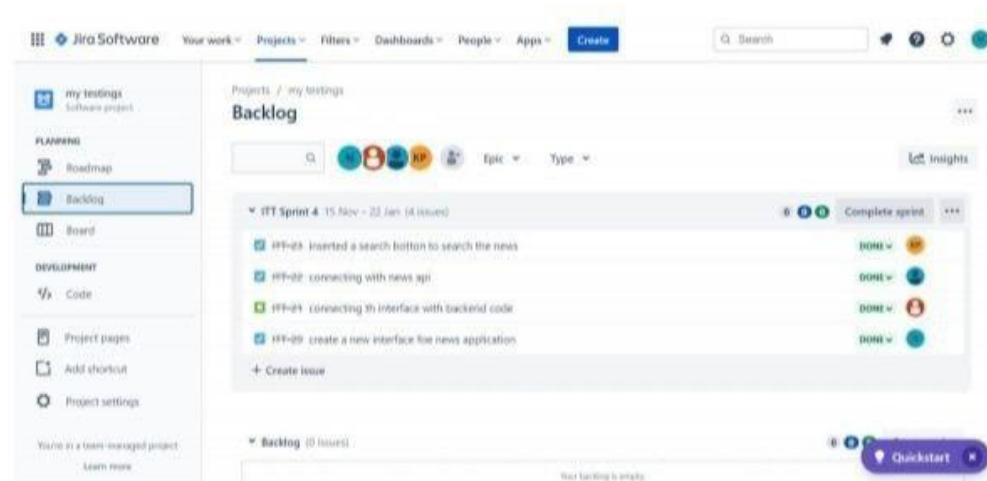
Sprint-2



Sprint-3



Sprint-4



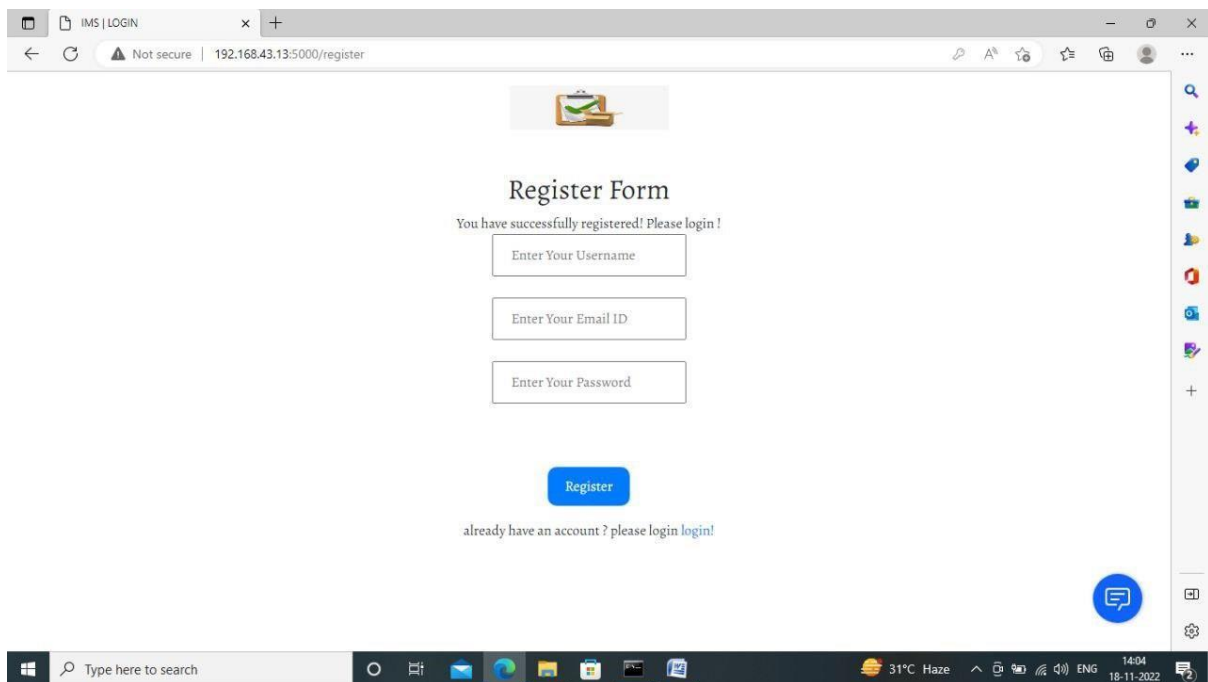
7.CODING & SOLUTIONING

7.1.FEATURE1



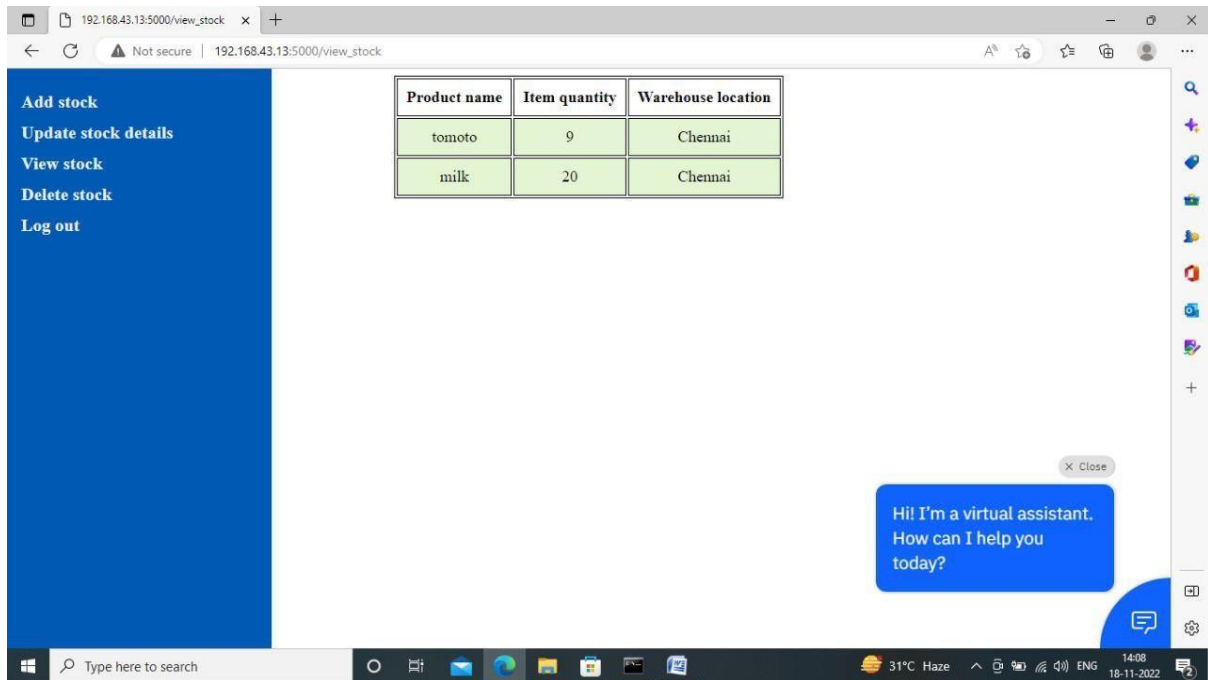
A screenshot of a web browser displaying a login form. At the top is a logo of a clipboard with a green checkmark. Below it is the title "Login Form". There are two input fields: the first contains the text "dharanee" and the second contains a series of dots for a password. Below the fields is a blue button labeled "Login". At the bottom, there is a link that says "Don't have an account yet? Click here to [register!](#)".

7.2. FEATURE 2



A screenshot of a web browser displaying a registration form. The browser's address bar shows "192.168.43.13:5000/register". The form has a logo at the top, followed by the title "Register Form". A message states "You have successfully registered! Please login!". There are three input fields labeled "Enter Your Username", "Enter Your Email ID", and "Enter Your Password". Below these is a blue button labeled "Register". At the bottom, there is a link that says "already have an account ? please [login](#) [login!](#)". The Windows taskbar is visible at the bottom of the screen.

7.3. DATABASE SCHEMA



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

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

Type here to search | 31°C Haze | 14:08 18-11-2022

8. TESTING

8.1.TEST CASE

Features to be tested	Test Description
Login to the system	This tests the login interface of the system.
Adding a Recipe to database	This test is conducted to verify if a recipe is successfully added to the database. This will check if the recipe is added to its header table and also check if the recipe details are added to the recipe details table.
Adding an Ingredient to database	This tests checks if new ingredient is added correctly to the database with the specified details.
Adding a Vendor to the database	This test checks if the newly added vendor is correctly added to the database with the specified details.
Checking the threshold levels	This test is conducted to verify if the ingredients that are below the threshold levels are listed by the function when called by the user. The verification is done by referring to the database.
Updating the sales for the day	This test is conducted to test the sales update in the database. The test checks if the database is updated with the correct ingredient values based on the sales data input to the system.
Updating the order reception to database	This test is conducted to test the correct updating of the database after receiving the order from the vendor.
Create Orders	This test is conducted to check the order creation capability of the system. The list of ingredients that is generated for order must comply with the set conditions of threshold levels

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.

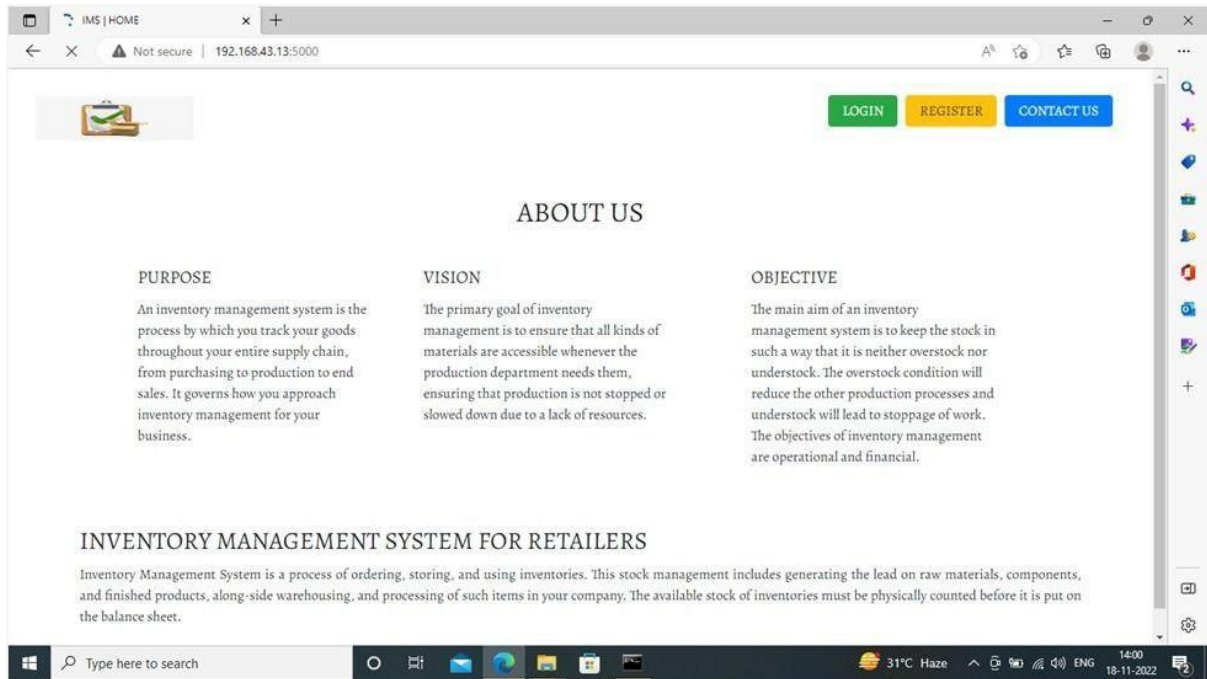
Case 2: Test the quantity text field..

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

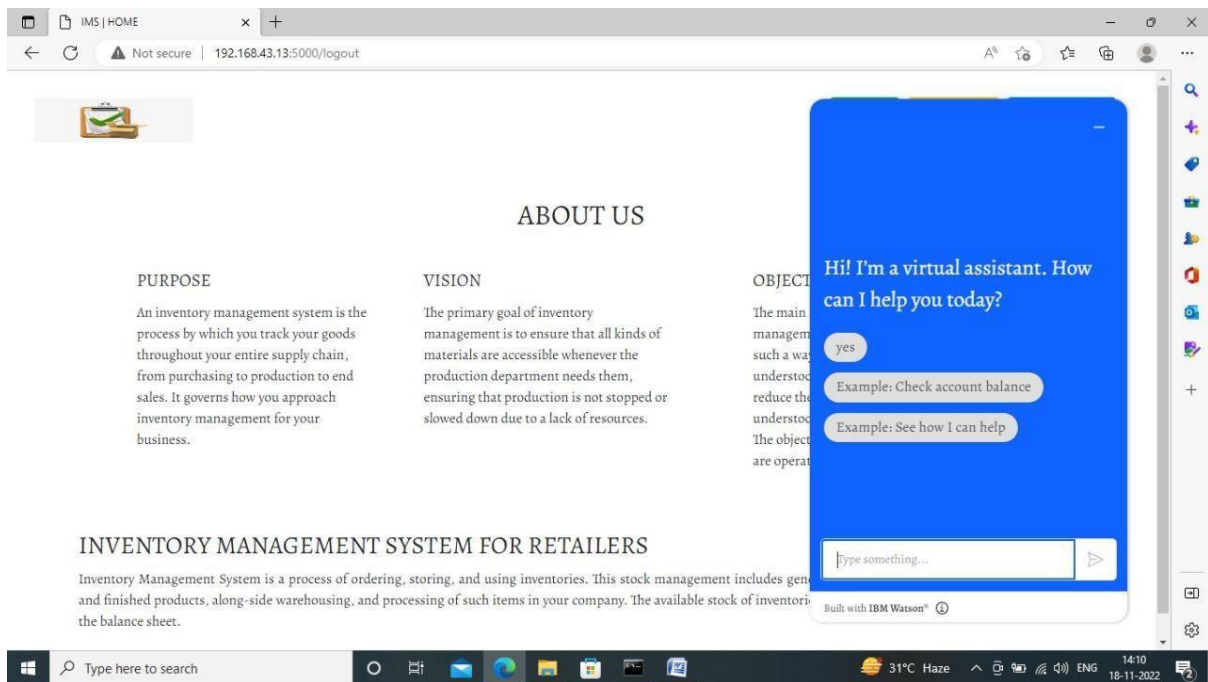
Case 4: Test if the details are updated to the database when requested.

9. RESULTS

HOME PAGE



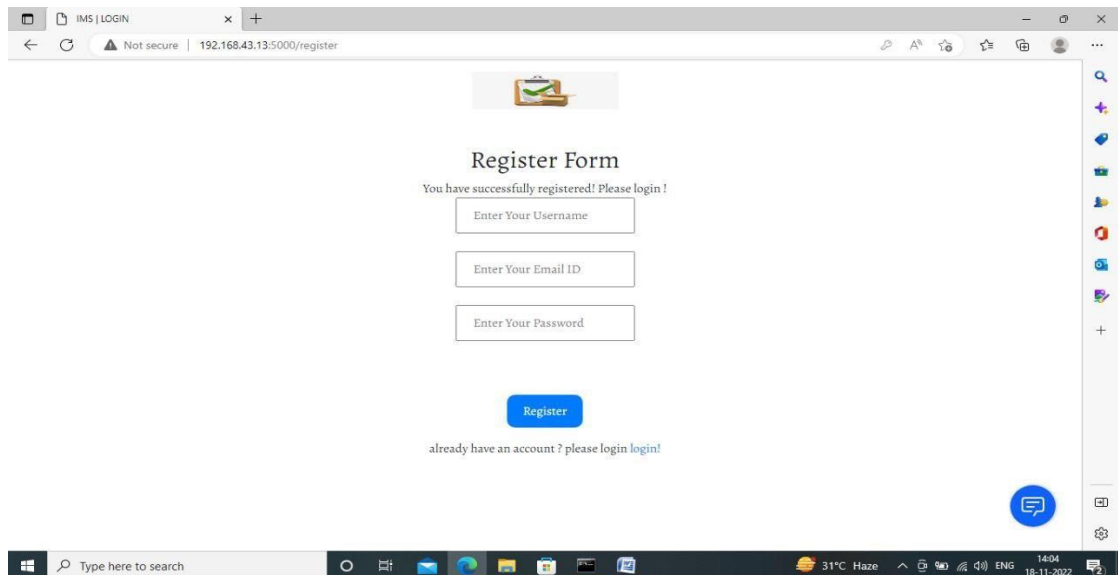
HOMEPAGE WITH VIRTUAL ASSISTANT



REGISTRATION FORM

The registration form is displayed on a white background. It features a logo at the top center, followed by the title "Register Form". Below the title are three input fields: "Enter Your Username", "Enter Your Email ID", and "Enter Your Password". A blue "Register" button is located below the input fields. At the bottom of the form, there is a link that says "already have an account ? please login login!". A blue speech bubble icon is located in the bottom right corner of the page.

REGISTRATION SUCCESSFULLY



The screenshot shows a web browser window with the title "IMS | LOGIN". The address bar displays "192.168.43.13:5000/register". The page content includes a "Register Form" with a message: "You have successfully registered! Please login!". Below this message are three input fields: "Enter Your Username", "Enter Your Email ID", and "Enter Your Password". A blue "Register" button is positioned below the password field. At the bottom of the form, there is a link: "already have an account ? please login login!". The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system tray on the right indicates a temperature of 31°C, Haze weather, and the date 18-11-2022.

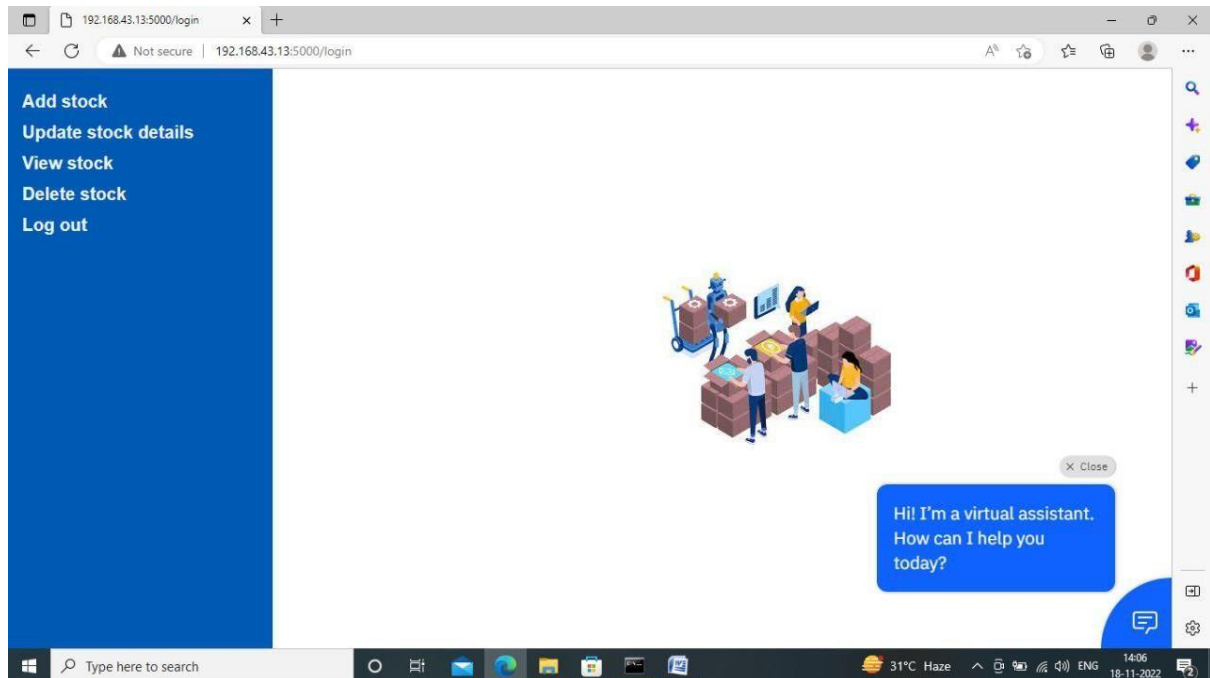
LOGIN FORM



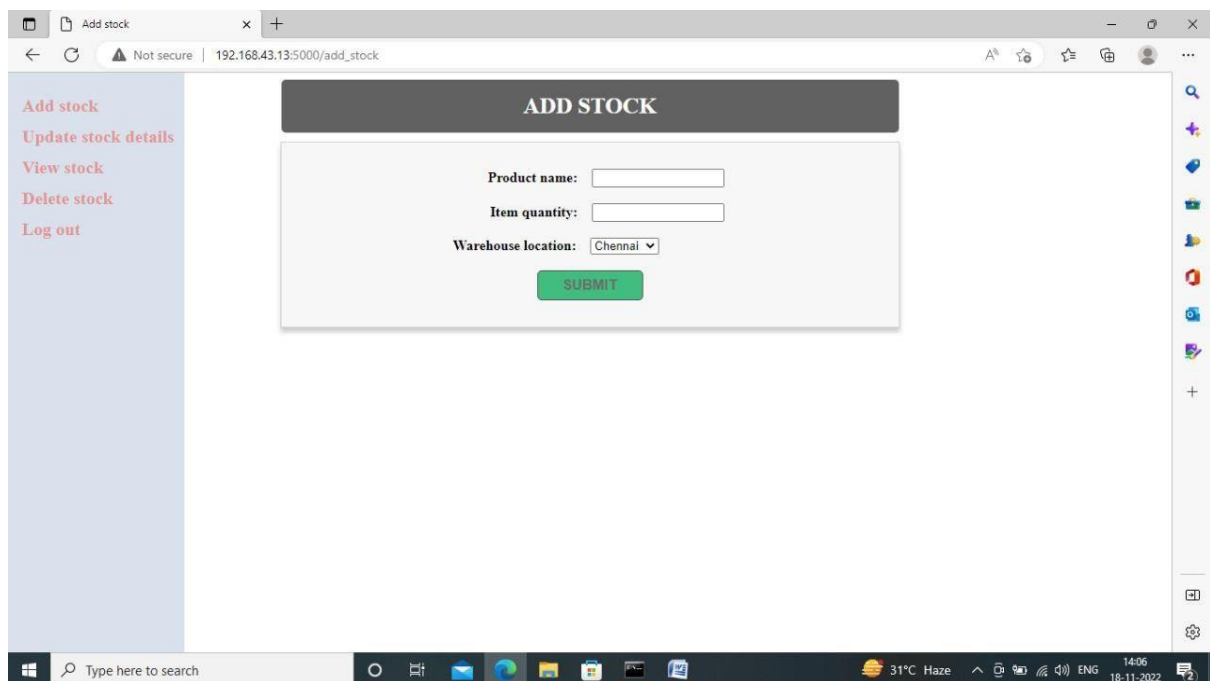
The screenshot shows a login form with the title "Login Form". It contains two input fields: the first is labeled "dharanee" and the second is a password field with masked characters. A blue "Login" button is located below the password field. At the bottom of the form, there is a link: "Don't have an account yet? Click here to register!".



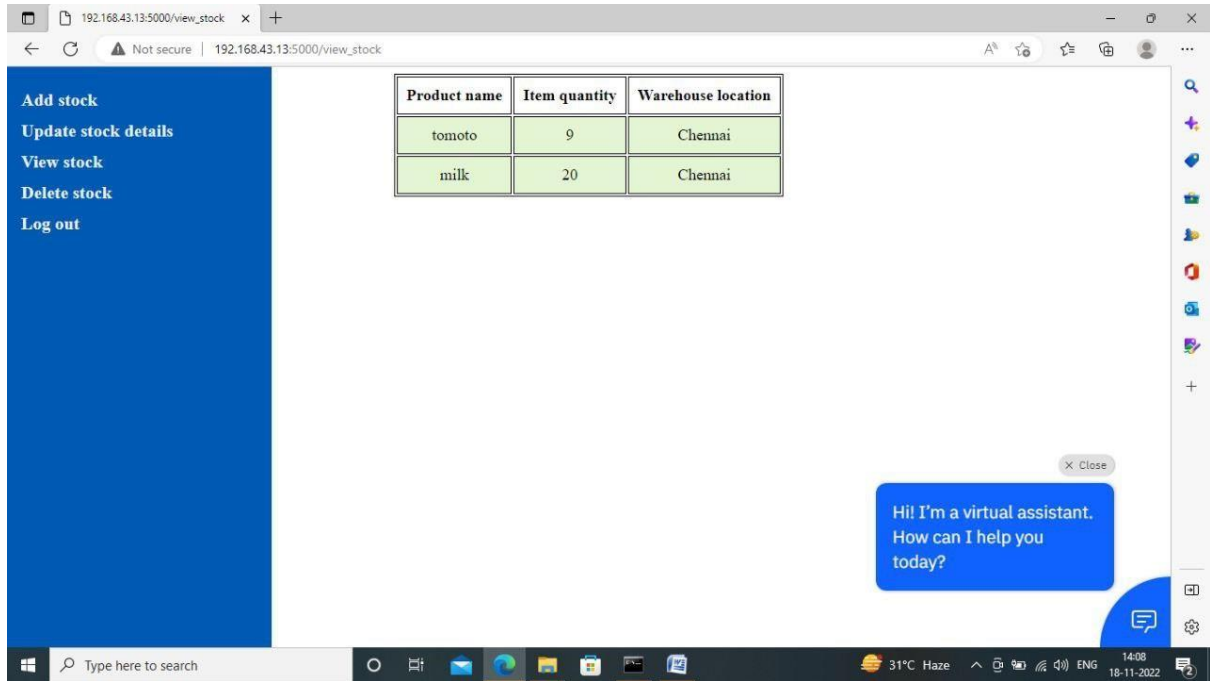
DASHBOARD



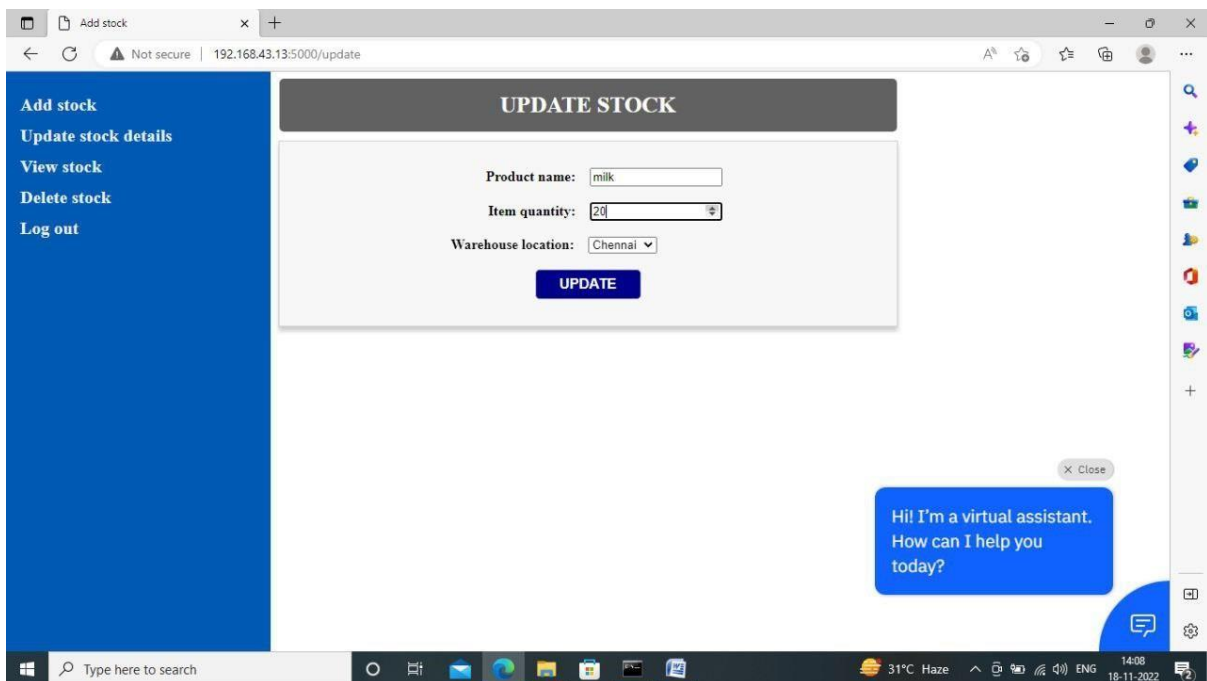
ADDSTOCK



VIEW STOCK



UPDATE STOCK



UPDATESTOCKDETAILS

The screenshot shows a web browser window with the address bar displaying '192.168.43.13:5000/view_stock'. The page has a blue sidebar on the left with the following menu items: 'Add stock', 'Update stock details', 'View stock', 'Delete stock', and 'Log out'. The main content area displays a table with the following data:

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

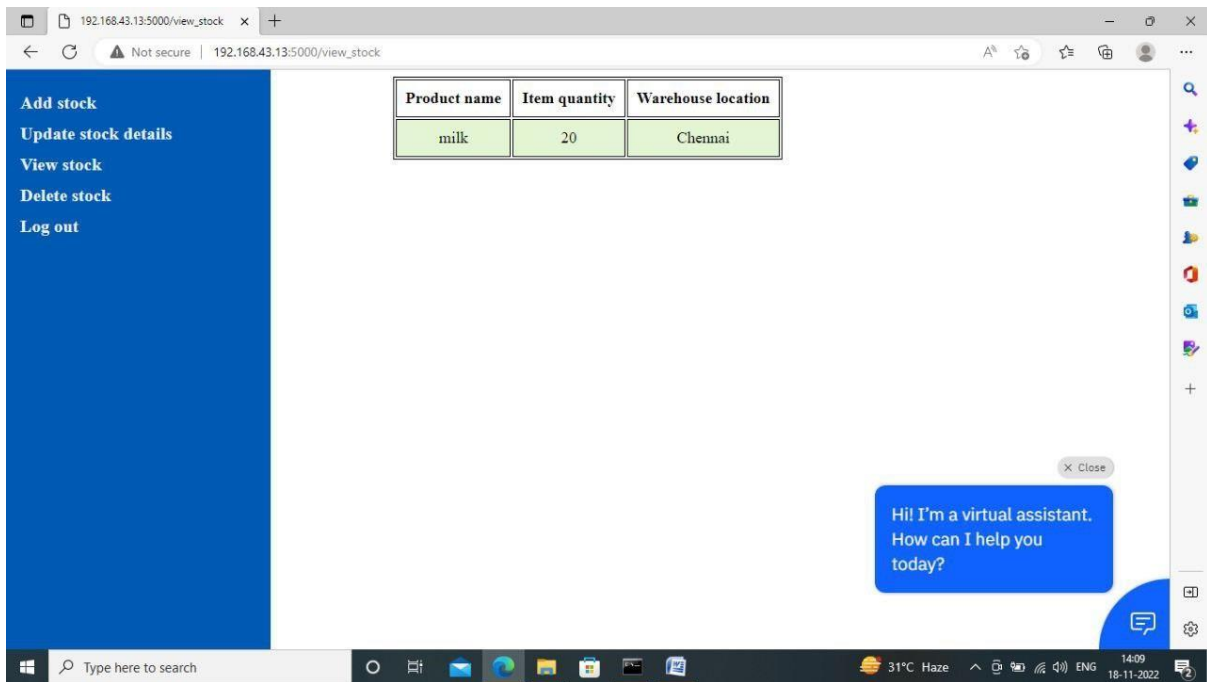
In the bottom right corner, there is a blue chat bubble with the text: 'Hi! I'm a virtual assistant. How can I help you today?'. The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates a temperature of 31°C, 'Haze' weather, and the date '18-11-2022'.

DELETE STOCK

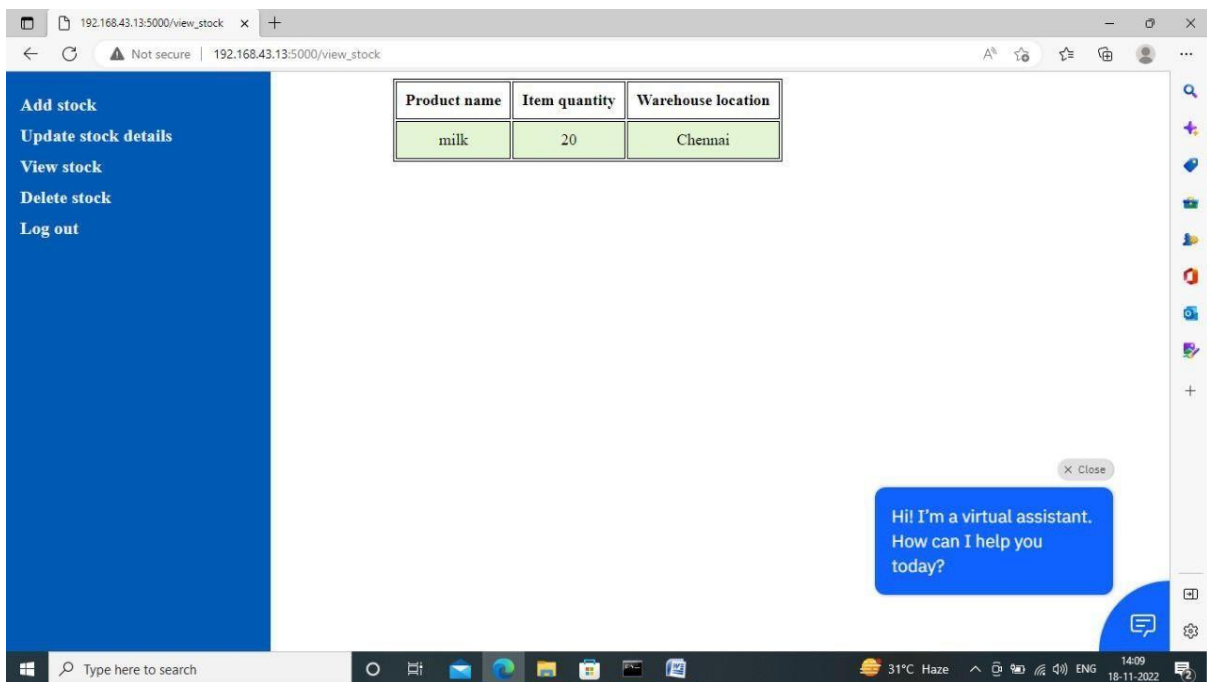
The screenshot shows a web browser window with the address bar displaying '192.168.43.13:5000/delete'. The page has a blue sidebar on the left with the following menu items: 'Add stock', 'Update stock details', 'View stock', 'Delete stock', and 'Log out'. The main content area features a 'Delete stock' form with the following fields:

Product name:

In the bottom right corner, there is a blue chat bubble with the text: 'Hi! I'm a virtual assistant. How can I help you today?'. The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates a temperature of 31°C, 'Haze' weather, and the date '18-11-2022'.



LOGOUT



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

ADVANTAGES

1. **It helps to maintain the right amount of stocks:** contrary to the belief that is held by some people, inventory management does not seek to reduce the amount of inventory that you have in stock, however, it seeks to maintain an equilibrium point where your inventory is working at a maximum efficiency and you do not have to have many stocks or too few stocks at hand at any particular point in time. The goal is to find that zone where you are never losing money in your inventory in either direction. With the aid of an efficient inventory management strategy, it is easy to improve the accuracy of inventory order.
2. **It leads to a more organized warehouse:** with the aid of a good inventory management system, you can easily organize your warehouse. If your warehouse

is not organized, you will find it very difficult to manage your inventory. A lot of businesses choose to optimize their warehouse by putting the items that have the highest sales together in a place that is easy to access in the warehouse. This ultimately helps to speed up order fulfilment and keeps clients happy.

DISADVANTAGES

1. Bureaucracy: even though inventory management allows employees at every level of the company to read and manipulate company stock and product inventory, the infrastructure required to build such a system adds a layer of bureaucracy to the whole process and the business in general. In instances where inventory control is in-house, this includes the number of new hires that are not present to regulate the warehouse and facilitate transactions. In instances where the inventory management is in the hands of a third party, the cost is a subscription price and a dependence on another separate company to manage its infrastructure. No matter the choice you go for, it translates to a higher overhead cost and more layers of management between the owner and the customer. From the view point of the customer, a problem that requires senior management to handle will take a longer period of time before it will be trashed out.

2. Impersonal touch: another disadvantage of inventory management is a lack of personal touch. Large supply chain management systems make products more accessible across the globe and most provide customer service support in case of difficulty, but the increase in infrastructure can often mean a decrease in the personal touch that helps a company to stand out above the rest. For instance, the sales manager of a small manufacturing company that sells plumbing supplies to local plumbers can throw in an extra box of washers or elbows at no charge to the customer without raising any alarms. This is done for the sake of customer

relations and often makes the customer feel like he is special. While free materials can also be provided under inventory management, processing time and paper work make obtaining the material feel more like a chore for the customer or even an entitlement.

11.CONCLUSION

The project “Inventory Management System for Retailers” mainly as the name suggests deals with the calculation of the available and processed resources for an accurate inventory control and process management for a domain specific client who are related to the subject of food chains/outlets. This enables the inventory to be applied at every level in the hierarchy of the products and its complex combinations of recipes. A system that accurately calculates the atomic ingredients used for making a recipe then automatically performs the back end operation pertaining to a database of many relational tables onto which the changes are being made with each and every operation performed on the front end and which also shows up if at the time of retrieval. The most important part of Inventory controlling is its ability to check for threshold levels and alert the manager to replenish the stock before it reaches a danger zone. So as when an ingredient level goes below the threshold level then it routes an alert to the manager. Then if needed accordingly an automated order form is produced so as to each specific vendor along with the quantities needed for replenishment. As a part of the standard maintaining a drill of risk management is done in order to sustain during the days of special occasion or holidays when the demand reaches to rather more different scale as compared to other days. These occasions call on for special inclusions into the menu which reflects on the recipes and in turn reflects the ingredients being used up eventually. Thus was provided the liberty

of adding special recipe to the menu for some special occasion and is regarded as a key feature.

.

12.FUTURE SCOPE

- The Fourth Industrial Revolution will continue to drive technological change that will impact the way that we manage inventories.
- Successful companies will view inventory as a strategic asset, rather than an aggravating expense or an evil to be tolerated.
- Collaboration with supply chain partners, coupled with a holistic approach to supply chain management, will be key to effective inventory management.

13.APPENDIX

SOURCE CODE :

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(prepare_stmt, 1, username)
ibm_db.bind_param(prepare_stmt, 2, email)
ibm_db.bind_param(prepare_stmt, 3, password)

```

```

ibm_db.execute(prepare_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_location']
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 exists!!'      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','POST'])
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','POST']) 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_location']
        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_stock') 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('/update') 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 ID of 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-42439-1660662734>

DEMOLINK:

<https://drive.google.com/file/d/1TQ9-S3DH1oAsYUACHP8-4sBWADPIFjoe/view?usp=drivesdk>