Project Report

PROJECT TITLE :

Inventory Management System for Retailers

Team ID	PNT2022TMID21628
Team Lead	Gowtham S
Team Member 1	Hariharan S
Team Member 2	Hemanth C
Team Member 3	Kabilesh K

INDEX

1. INTRODUCTION

1. Project Overview
2. Purpose

2. LITERATURE SURVEY

1. Existing problem
2. References
3. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

4. REQUIREMENT ANALYSIS

1. Functional requirement
2. Non-Functional requirements

5. PROJECT DESIGN

1. Data Flow Diagrams
2. Solution & Technical Architecture
3. User Stories

6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule
3. Reports from JIRA

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

1. Feature 1
2. Feature 2
3. Database Schema (if Applicable)

8. TESTING

1. Test Cases
2. User Acceptance Testing

9. RESULTS

1. Performance Metrics

10. ADVANTAGES & DISADVANTAGES
11. CONCLUSION
12. FUTURE SCOPE
13. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

1.1. Project Overview

The goal of retail inventory management is to maintain the right amount of desired product in stock—neither too little nor too much. Retailers can meet customer demand without running out of inventory by properly managing their inventory.

Effective retail inventory management reduces costs and improves knowledge of sales patterns in practise. Tools and techniques for retail inventory management provide merchants with more data on which to run their businesses. Applications have been created to assist shops in keeping track of and managing the supply of their own products. Retailers will be prompted by the System to register their accounts by providing necessary information. Retailers can log into the application to access their accounts.

Once retailers have successfully logged in to the programme, they can update the information on their inventory. Users can also add new goods by providing the necessary information regarding the item. They have access to the current inventory's specifics. If there is no stock found in the accounts of the retailers, the System will automatically send an email alert to them. in order for them to order new stock.

1.2 Purpose

The main goal of inventory management is to make it simple and effective for businesses to order, stock, store, and use inventory. You'll always be aware of the items you have in stock, their quantity, and location if you manage your inventory well.

Additionally, good inventory management practises help you comprehend how you use your stock and how demand varies over time. You may focus on what you really need, what is unnecessary, and what is just a waste of money. That is the use of inventory management to put inventory control into effect. Inventory control, incidentally, involves striking a balance between keeping enough inventory on hand to meet demand at all times and minimising the cost of ordering and maintaining inventory.

2. LITERATURE SURVEY

2.1 Existing Problem

Inconsistent Tracking:

Using manual inventory monitoring techniques across many programmes and spreadsheets is time-consuming, redundant, and prone to mistakes. An integrated central inventory monitoring system with accounting capabilities might be helpful for even small organisations.

Warehouse Efficiency:

Controlling inventory in a warehouse is a labor-intensive process that includes picking, packing, and shipping among other labor-intensive stages. Performing all of these duties as efficiently as feasible is the difficulty.

Inaccurate Information:

You must always be aware of the exact quantity of your goods. The time when inventory could be counted with all hands on deck once a year is long gone.

Demand Variation:

Consumer demand is continuously changing. Storing too little could prevent you from filling customer orders, while keeping too much might cause you to accumulate outmoded inventory that you can't sell. Using technology to design and carry out an inventory plan, as well as order tactics for essential commodities, can assist offset fluctuating demand.

Limited visibility:

It causes incomplete, inaccurate, or delayed shipments when your inventory is difficult to discover in the warehouse or identify. Effective warehouse operations and satisfying customer experiences depend on receiving and locating the appropriate goods.

Manual Documentation:

Inventory management with paperwork and manual procedures is time-consuming and unsafe. Additionally, scaling across several warehouses with a lot of goods is difficult.

Problem Stock:

Specialized care and storage plans are required for perishable and fragile stock. Additionally, specific loss-prevention techniques and inventory controls are required for high-value inventory.

Complexity of the supply chain:

Your inventory planning and management activities are hampered by the daily shifts in the world's supply networks. The wholesalers and manufacturers who choose when, when, and how your Inventory shipments demand adaptability and have variable lead periods.

Managing Warehouse Space:

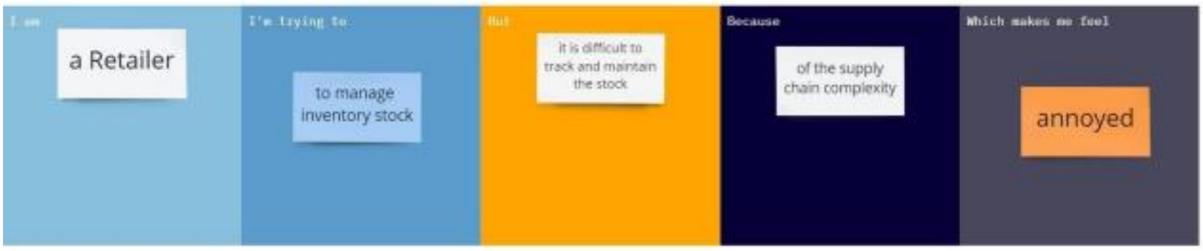
Managing space effectively is a daunting task. Utilizing inventory management solutions to plan and construct warehouse facilities enables you to better manage the timing of new product deliveries. It can take into account crucial elements like space availability. Find out more about the variations between inventory management and warehouse management.

2.2 References

S.No.	Title	How does it relatedto my study	Used methodologies	limitation	Recommendation from this paper	Year
	Agricultur al InventoryManagement System	Support national commerce policies by annually, semi-annually, quarterly and monthly periods	Assets andtheir details arerecorded at village and farmer level.	opportunity cost and handle the investment in inventory are more the funds are blocks up with inventory.	To make proper agricultural inventory management system to make the proper yield of productsto besold at good price.	2015 Fourth InternationalConferenceon Agro- Geoinformatics (Agro- geoinformatics)
	Online inventorymanagement of packagedgases	Using advanced gadgets to managethe product or Itemsto be managed	By sensorsthe gas are monitored and the total amount is measured managed.	The proper functioning of sensors are monitored and uninterruted power supply should be provided	Making using of available moderngadgets for the proper maintenance of the inventories	2010 IEEE Sensors ApplicationsSymposium(SAS)
	A Case Study of InventoryManagement Systemfor an International Lifestyle Product Retailer inBolivia	It is to help introducing the new type of innovations should be carried out using modern techniques, Finally the products recommend by users.	Survey from user in different categories, considering the datacollected.	Tradition alway of placing theitems.	To include modern techniques according to thelatest trend.	April 2021

	A Studyof InventoryManagement SystemCase Study	To make use of holding costs and this is frozen fundthat can be lost ,Is move as a basicinvestment for the products	Analyzingof properfinancial demands	To change only the frozen funds	To make use ofavailable fundsand not by getting any debts.	Sep 2018
	A Study of InventoryManagement System ofLinamar India	The ultimate aim ofthe study is to examine the inventory management process.	Use various alternativemethodologies to maintain the inventory managem ent system	Inaccurate reordering ofproducts andorder Tracking.	To increase the rate of accuracyand proper management trying alternativeways.	June 2018

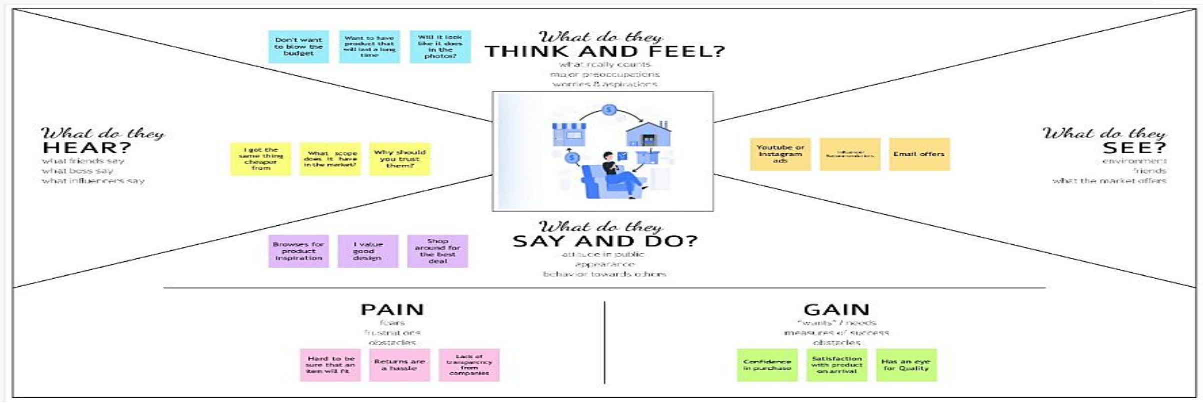
2.3 Problem Statement Definition


I am
a Retailer
I'm trying to
to manage inventory stock
But
It is difficult to track and maintain the stock
Because
of the supply chain complexity
Which makes me feel
annoyed

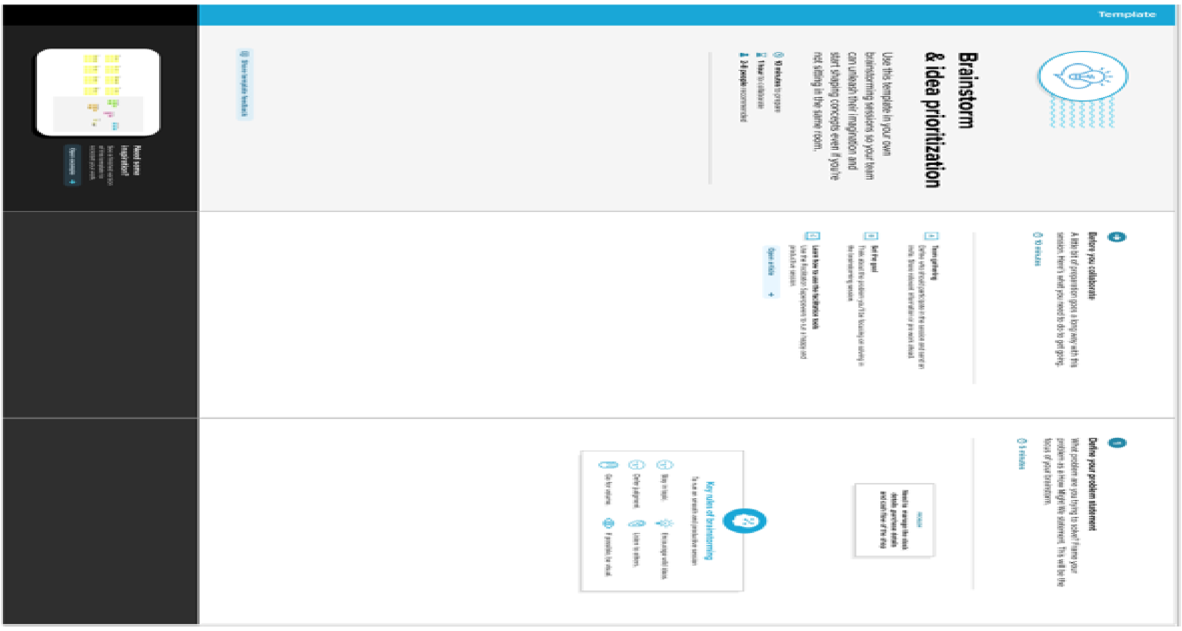
Problem Statement (PS)	I am (Customer)	I'm trying to	Because	Which makes me feel
PS-1	Retailer	Manage the inventory	Due to human error, Lack of interest and consciousness	Frustated

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming


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

Need some inspiration?

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

Team gathering

Set the goal

Learn how to use the facilitation tools

Open article

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.

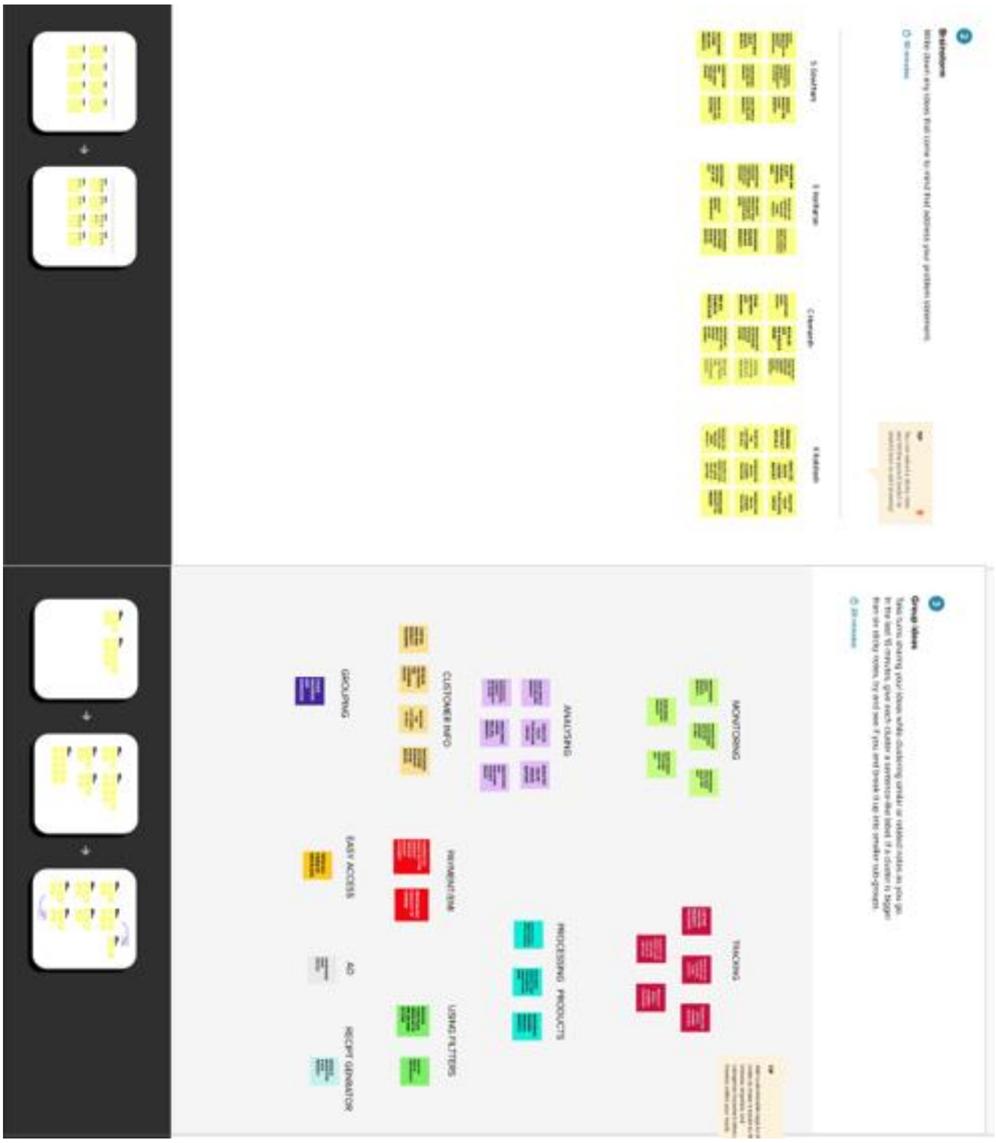
5 minutes

Key rules of brainstorming

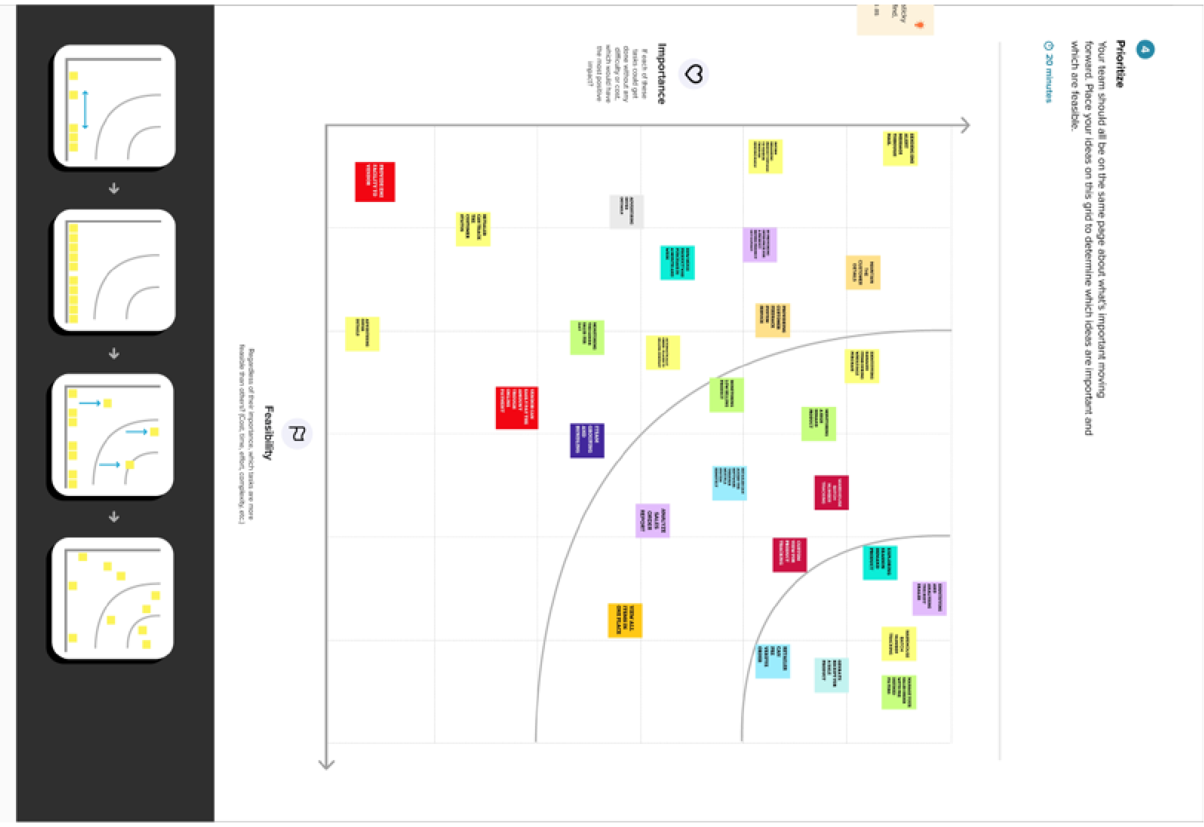
To run an smooth and productive session

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



Step-3: Idea Prioritization



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?

Feasibility

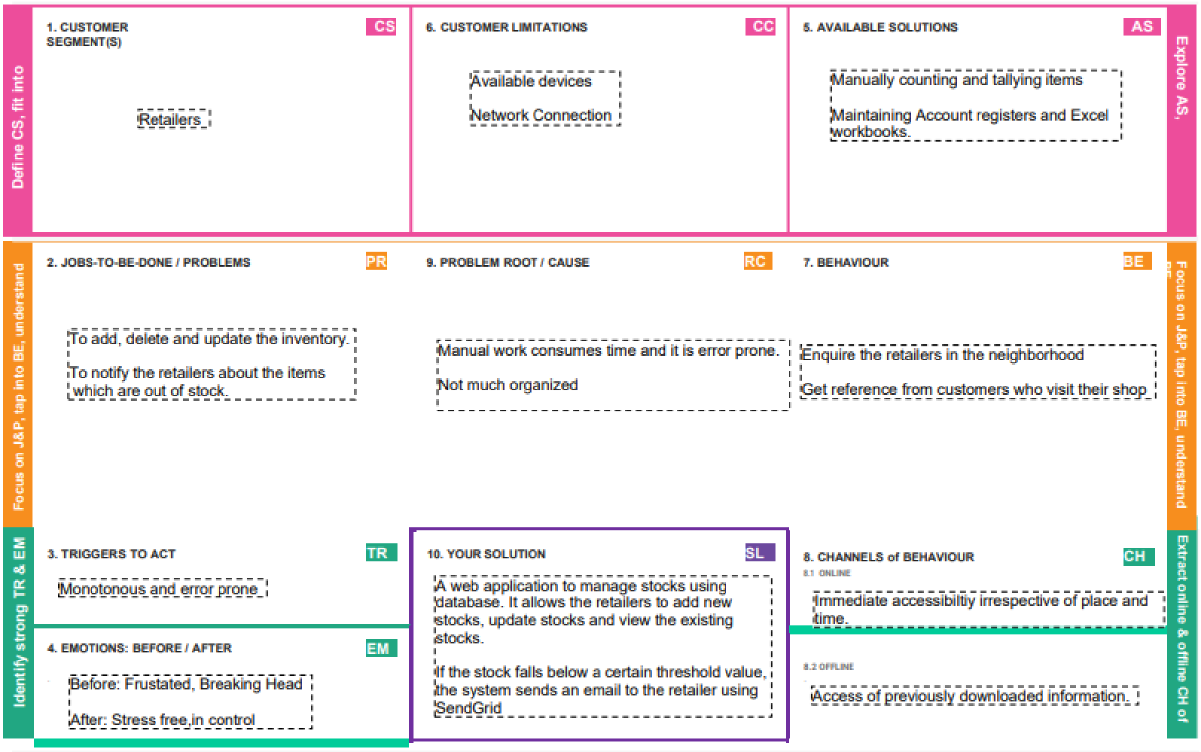
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement(Problem to be solved)	inventory reports, meeting consumer requests, and adjusting to changes in productdemand.
2.	Idea / Solution description	

3.	Novelty / Uniqueness	1. It gives solutions to the unsoldstocks. 2. It produces gooduser experience and also provides suggestions based on trends. 3. It manages the profit and loss of the retail store.
4.	Social Impact / Customer Satisfaction	1. Effortless access: Users on this platform will be able to use this model with justOne-click. 2. Easy to understand the customer's needs 3. It saves time and maintain the dataset of the complex sales and customer detailsefficiently.
5.	Business Model (RevenueModel)	
6.	Scalability of the Solution	This project is basedon retailing. There are huge number of retailstores are running over. So, it willbe on demandas it is helpfulto analyse the complex productsanlaysing.

3.4 Problem Solution Fit



Define CS, fit into
Explore AS,
Focus on J&P, tap into BE, understand
Focus on J&P, tap into BE, understand
Identify strong TR & EM
Extract online & offline CH of

1. CUSTOMER SEGMENT(S) CS

Retailers

6. CUSTOMER LIMITATIONS CC

Available devices

Network Connection

5. AVAILABLE SOLUTIONS AS

Manually counting and tallying items

Maintaining Account registers and Excel workbooks.

2. JOBS-TO-BE-DONE / PROBLEMS PR

To add, delete and update the inventory.

To notify the retailers about the items which are out of stock.

9. PROBLEM ROOT / CAUSE RC

Manual work consumes time and it is error prone.

Not much organized

7. BEHAVIOUR BE

Enquire the retailers in the neighborhood

Get reference from customers who visit their shop

3. TRIGGERS TO ACT TR

Monotonous and error prone

4. EMOTIONS: BEFORE / AFTER EM

Before: Frustated, Breaking Head

After: Stress free,in control

10. YOUR SOLUTION SL

A web application to manage stocks using database. It allows the retailers to add new stocks, update stocks and view the existing stocks.

If the stock falls below a certain threshold value, the system sends an email to the retailer using SendGrid

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Immediate accessibiltiy irrespective of place and time.

8.2 OFFLINE

Access of previously downloaded information.

4. REQUIREMENT ANALYSIS

4.1 Funtional Requirement

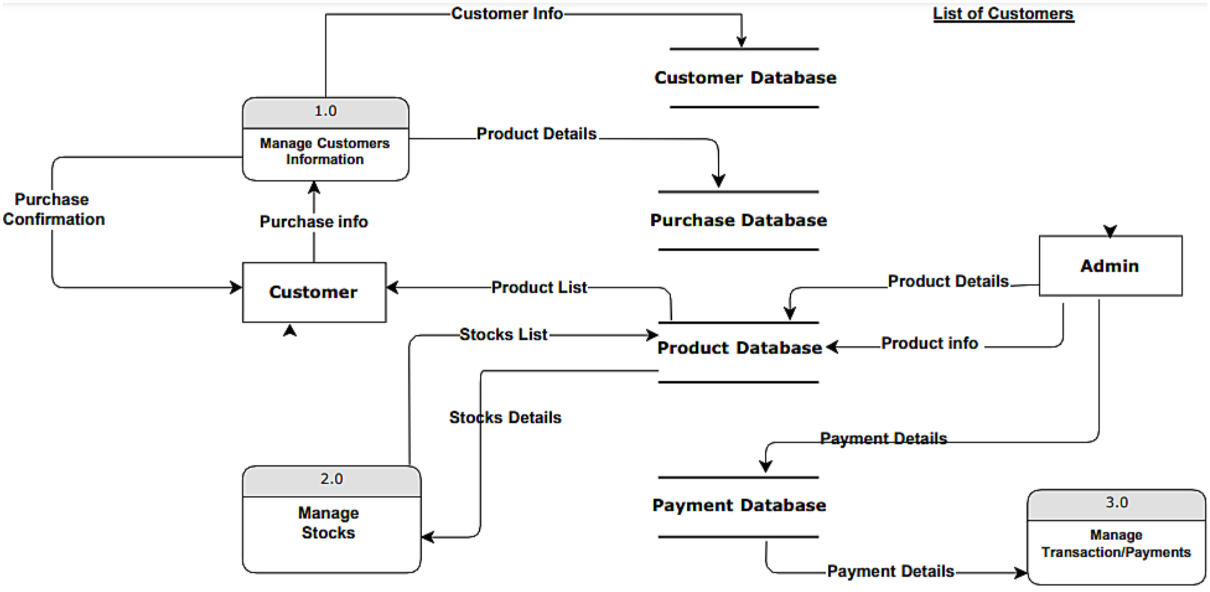
FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Login	Login through Form
FR-3	Addand delete inventory	Getthe details through form and store it in the database
FR-4	Update inventory	Getthe details through form and update it in the database
FR-5	View Details	Display the details according to the query
FR-6	Alert Mail	Notify the retailers through SendGrid
FR-7	Logout	Logs out by clicking the logout button

4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Everyone should be able to use the UI regardless oftheirlanguage preferences. The app's UI and platform should match the device.independent. It should work with a variety of potential rangeof gadgets People with certain disabilities should additionallyable to easily use the application. (Example, Google Assistant shouldbe included so thatblind persons utilise it)
NFR-2	**Security**	Access permissions for the particular system information may only be changed by the system’s data administrator.
NFR-3	**Reliability**	In the event that the server fails due to software orhardware failure, the software should be able to connect to the database.
NFR-4	**Performance**	Theapp's performance should be dependable, with the software running on high-end servers.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



Payment Details

5.2 Solution & Technical Architecture

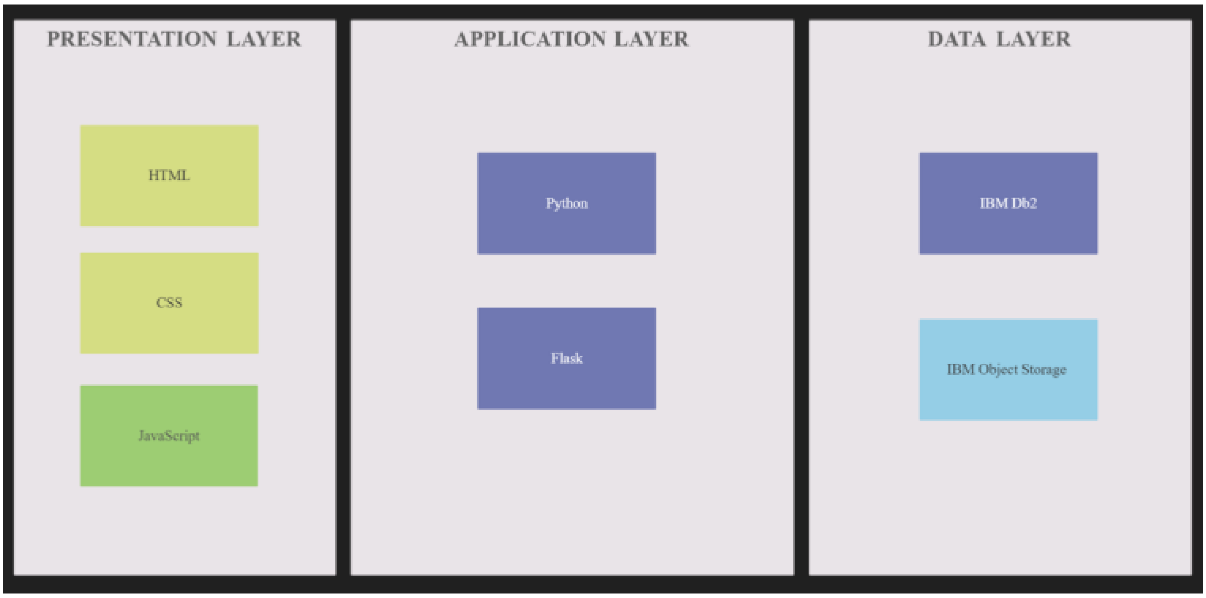


Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.Web UI,Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js /ReactJs etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM WatsonSTT service
4.	Application Logic-3	Logic for a process in the application	IBM WatsonAssistant
5.	Database	Data Type,Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storagerequirements	IBM Block Storage or Other StorageService or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM WeatherAPI, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API,etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal ServerConfiguration: Cloud Server Configuration :	Local, CloudFoundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	e.g. SHA-256, Encryptions, IAMControls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier,Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use ofloadbalancers, distributed servers etc.)	Technology used

5.	Performance	Design consideration for the performance of theapplication (number of requests per sec, use of Cache, use of CDN’s) etc.	Technology used

5.3 User Stories

Table 1 - Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Initial Registration bycustomer	USN-1	As a user, I can register for the applicationby entering my email or number, and password, and confirming my password.	3	High	
Sprint-1	Confirmation forregistering	USN-2	As a user, I can receivemy confirmationthroughemail or number once I have registered for the application.	3	Medium	
Sprint-1	Initial Login	USN-3	As a user, I can log in to the authorizedaccount by entering the registered emailor number and password.	2	High	
Sprint-1	Viewing Dashboard	USN-4	As a user,I can view my accountdetails which I have given duringmy registration process.	3	High	

Sprint-2	Adding products to cart and orderconfirmation.	USN-5	As a user, I can view the available productsand it to the cart and once the cart is filled Icanconfirm my order.	2	High	Hariharan S Kabilesh K

Sprint	Functional Requirement(Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Feedbacksystem	USN-6	As a user,I can give the feedback about theproduct and/or the service and chat with theserviced provider.	2	Medium	
Sprint-2	Registration forretailer	USN-7	As a user, I can register for the applicationby entering my email, and password, andconfirming my password.	2	High	
Sprint-2	Confirmation	USN-8	As a user, I can receivemy confirmationemail once I have registered for the application	2	Medium	
Sprint-2	Login	USN-9	As a user, I can log in to the authorizedaccount by entering the registered emailand password	2	High	
Sprint-3	Retailers Dashboard	USN-10	As a user, I can viewthe products thatarecurrently available.	8	High	

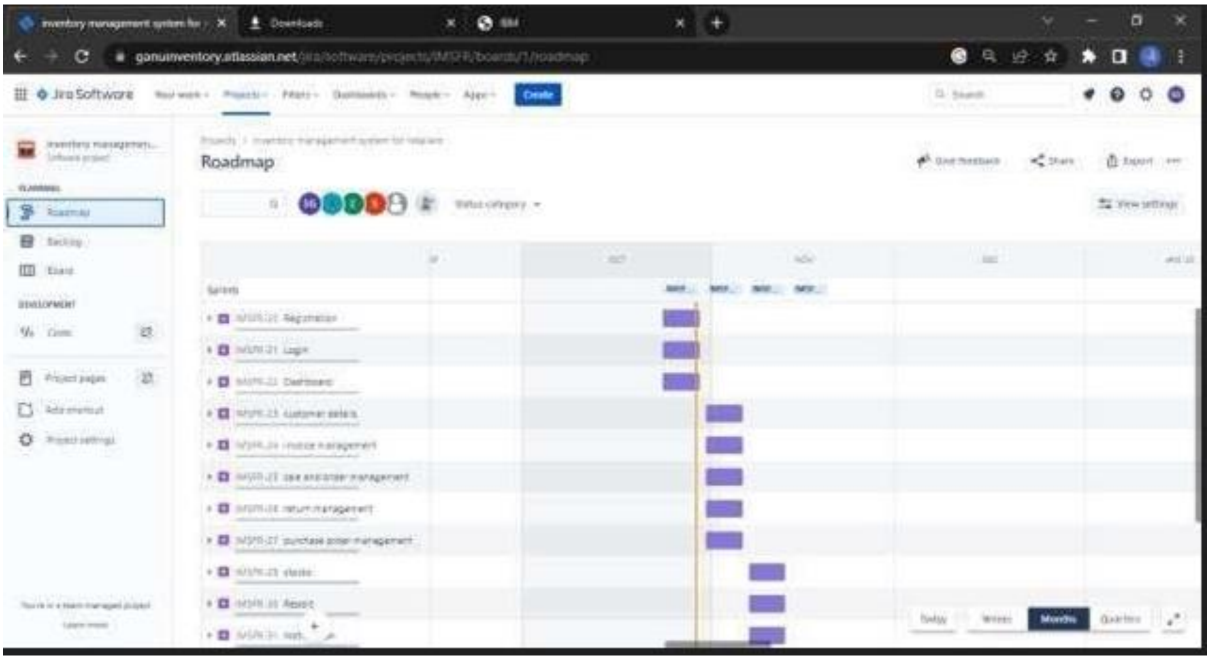
Sprint-4	Stocks update	USN-11	As a user, I can restorethe products whichare not available in the inventory and restore the inventory details.	2	Medium	
Sprint-4	Stock Alert	USN-12	Alerting theretailer when the stock is low onquantity by using SendGrid.	3	High	
Sprint-4	Admin Login	USN-13	As an admin, I can modify and verify theregular and customized users.	2	Medium	

Sprint	Functional Requirement(Epic)	User StoryNumber	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Dashboard and updation.	USN-14	As a user, I can changethe UI & update features	2	Medium	

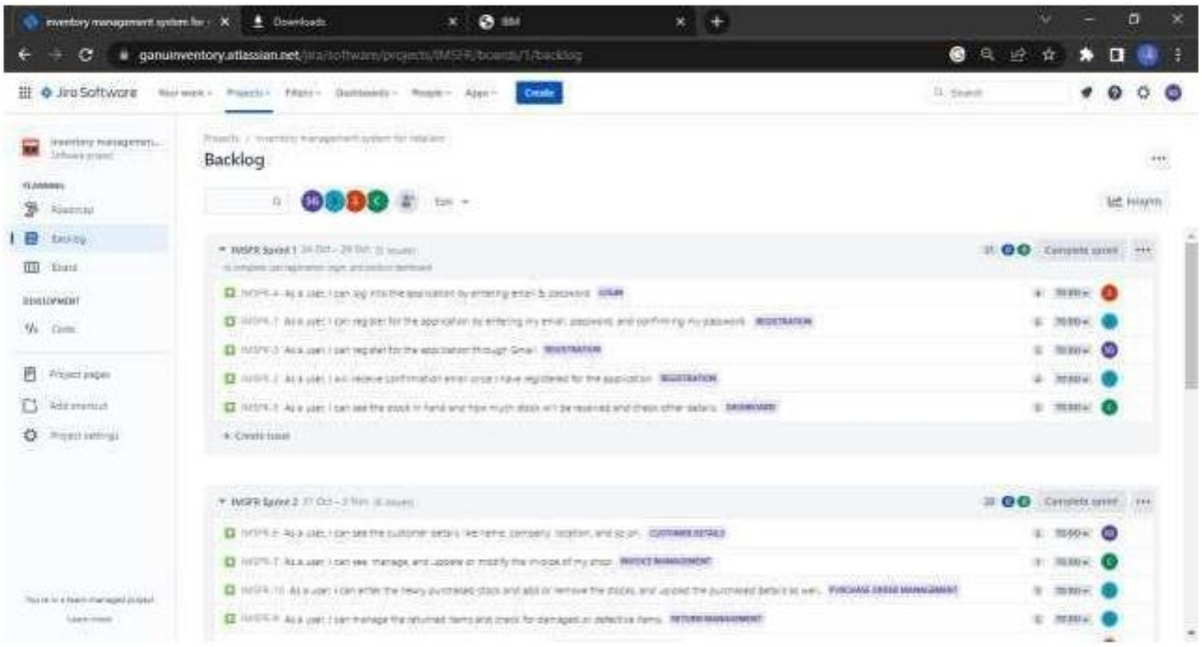
Table 2 -Project Tracker, Velocity& Burndown Chart:

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

6.3 Reports from JIRA



Roadmap



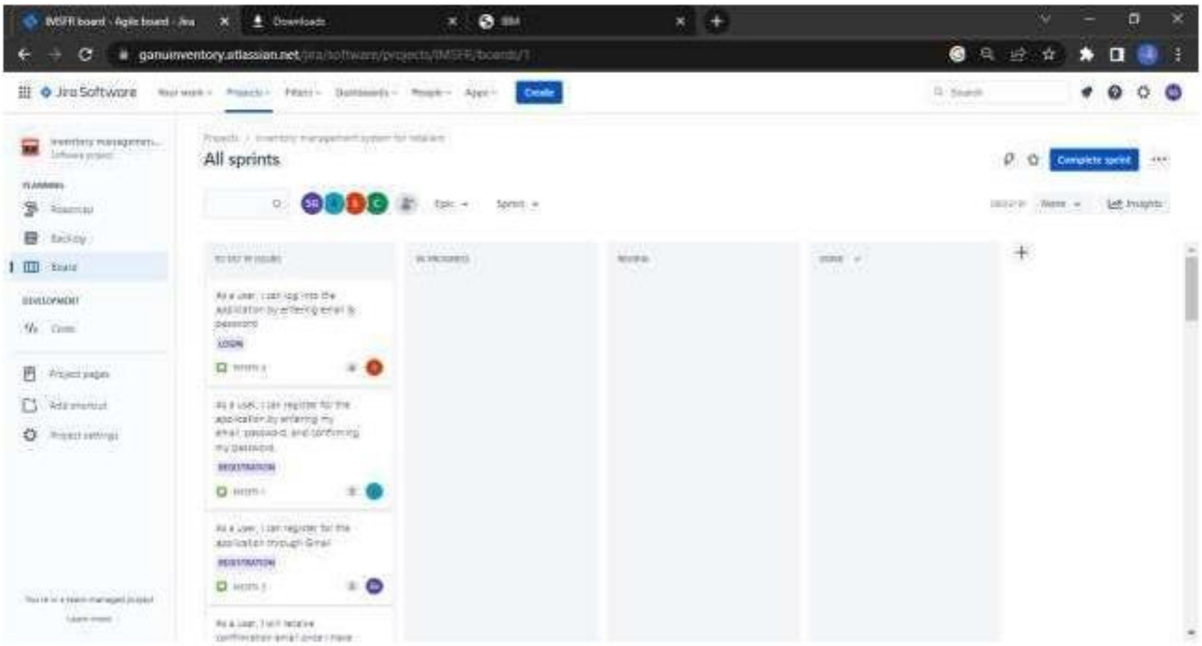
Backlog

IMSFR Sprint 1 24 Oct – 29 Oct (5 issues)

- As a user, I can log into the application by entering email & password LOGIN
- As a user, I can register for the application by entering my email, password, and confirming my password REGISTRATION
- As a user, I can register for the application through Gmail REGISTRATION
- As a user, I will receive confirmation email once I have registered for the application REGISTRATION
- As a user, I can see the stock in hand and how much stock will be received and check other details DASHBOARD

IMSFR Sprint 2 31 Oct – 5 Nov (6 issues)

- As a user, I can see the customer details like name, company, location, and so on. CUSTOMER DETAILS
- As a user, I can see, manage, and update or modify the invoice of my shop. INVOICE MANAGEMENT
- As a user, I can enter the newly purchased stock and add or remove the stocks, and upload the purchased details as well. PURCHASE ORDER MANAGEMENT
- As a user, I can manage the returned items and check for damaged or defective items. RETURN MANAGEMENT



Projects / Inventory management system for retailers

All sprints

TO DO

IN PROGRESS

REVIEW

DONE

As a user, I can log into the application by entering email & password

LOGIN

IMSFR-3

As a user, I can register for the application by entering my email, password, and confirming my password

REGISTRATION

IMSFR-1

As a user, I can register for the application through Gmail

REGISTRATION

IMSFR-2

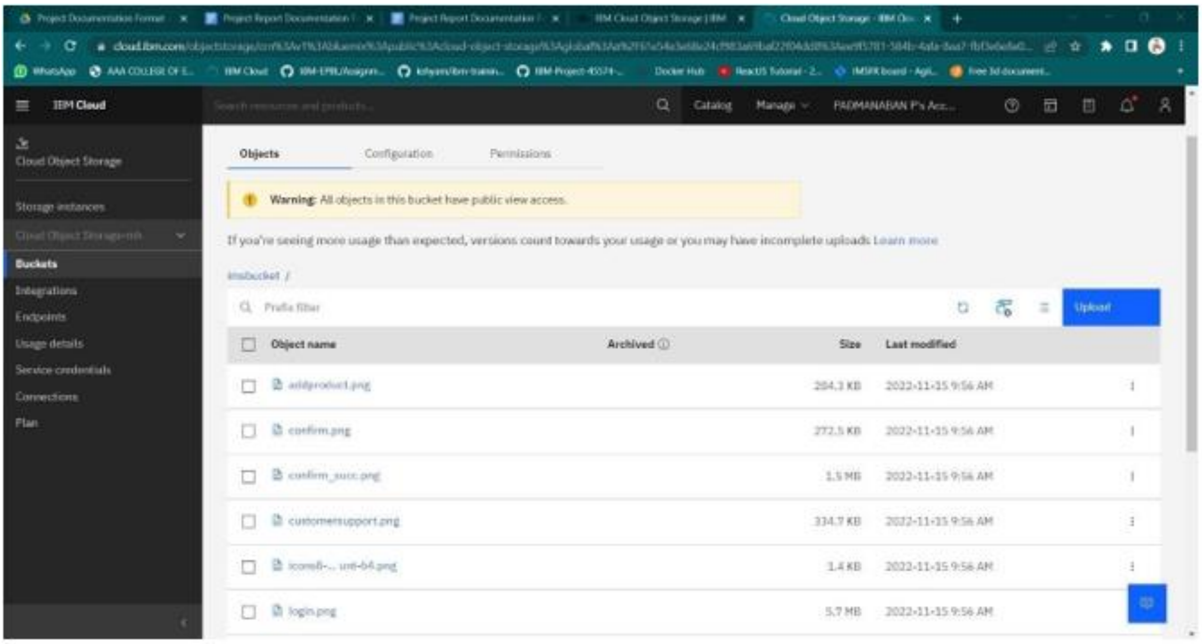
As a user, I will receive confirmation email once I have

7. CODING & SOLUTIONING

7.1 IBM Object Storage Service

IBM Cloud Object Storage, with its global presence and flexible resiliency options, supports exponential data growth for your cloud-native workloads with best-in-class cost optimization,robust data security, and data governance with ease of useBuilt-in data lifecycle operationsalsomake it easy to observeand manage your critical workloads.

Creating a bucketin IBM Object Storage and uploading the images to thebucket:

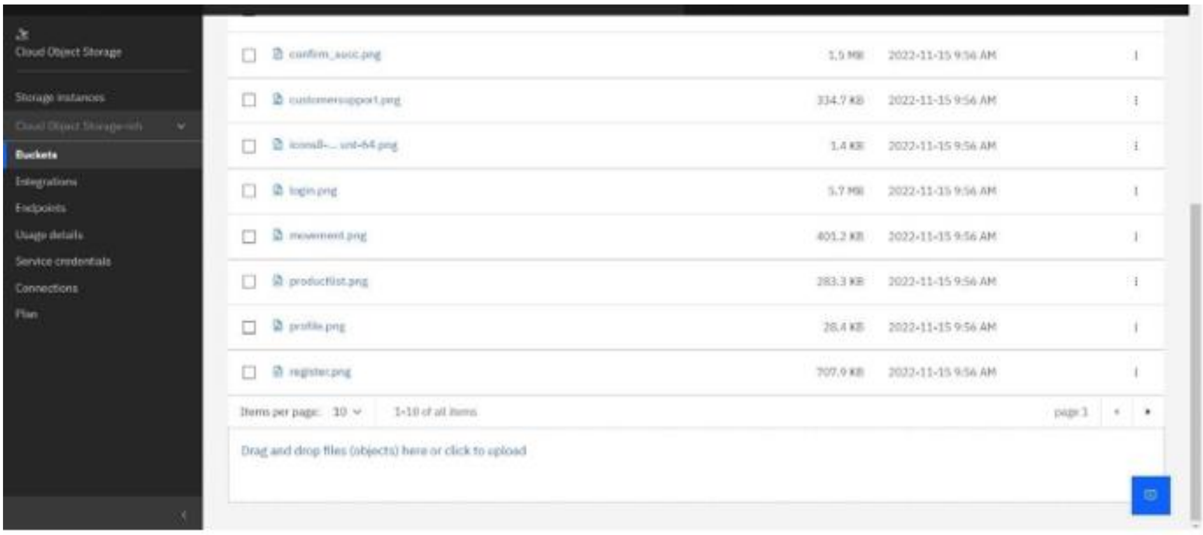


Warning: All objects in this bucket have public view access.

If you're seeing more usage than expected, versions count towards your usage or you may have incomplete uploads Learn more

insbucket /

Object name	Archived	Size	Last modified
addproduct.png		204.3 KB	2022-11-15 9:56 AM
confirm.png		272.5 KB	2022-11-15 9:56 AM
confirm_succ.png		1.5 MB	2022-11-15 9:56 AM
customersupport.png		334.7 KB	2022-11-15 9:56 AM
icons8-... unt-64.png		1.4 KB	2022-11-15 9:56 AM
login.png		5.7 MB	2022-11-15 9:56 AM



Object name	Size	Last modified
confirm_succ.png	1.5 MB	2022-11-15 9:56 AM
customersupport.png	334.7 KB	2022-11-15 9:56 AM
icons8-... unt-64.png	1.4 KB	2022-11-15 9:56 AM
login.png	5.7 MB	2022-11-15 9:56 AM
movement.png	401.2 KB	2022-11-15 9:56 AM
productlist.png	283.3 KB	2022-11-15 9:56 AM
profile.png	28.4 KB	2022-11-15 9:56 AM
register.png	707.9 KB	2022-11-15 9:56 AM

Items per page: 10 1-10 of all items page 1

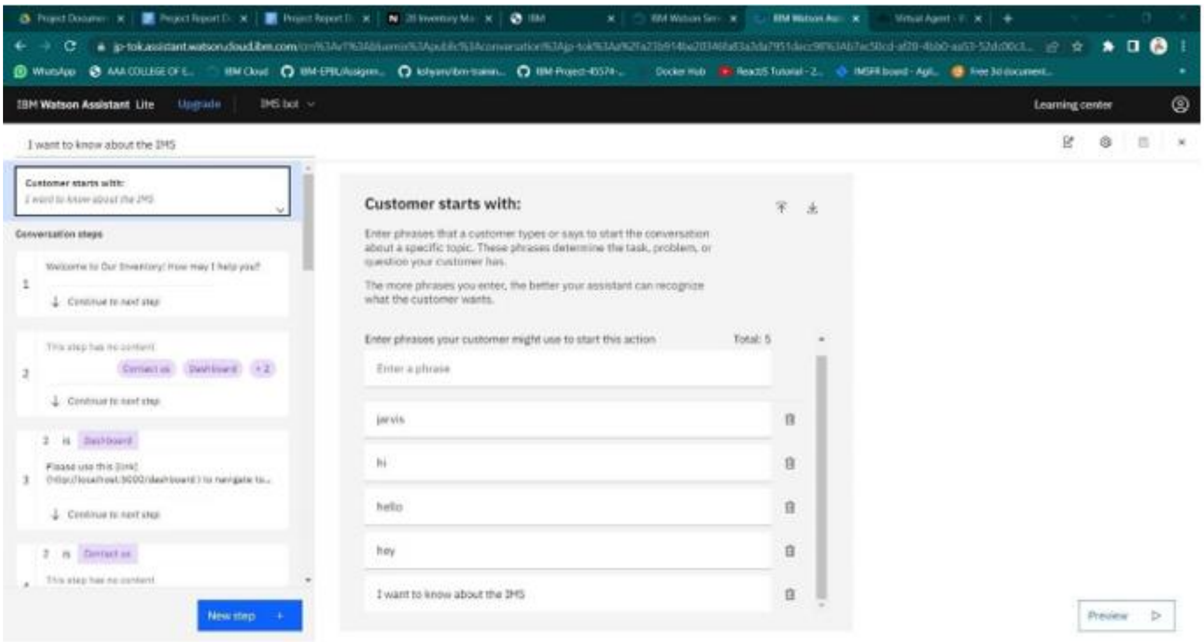
Drag and drop files (objects) here or click to upload

Displaying the images in the application by accessing the images publicly:

7.2 IBM Watson Assistant Service

IBM Watson Assistant uses artificial intelligence that understands customers in context to provide fast, consistent, and accurate answers across any application, device, or channel. Remove the frustration of long wait times, tedious searches, and unhelpful chatbots with theleader in trustworthy AI.

Setting Actions to IBM Watson Assistant:



I want to know about the IMS

Customer starts with:
I want to know about the IMS

Conversation steps

1 Welcome to Our Inventory! How may I help you?
↓ Continue to next step

2 This step has no content
Contact us · Dashboard · +2
↓ Continue to next step

3 2 is Dashboard
Please use this [link](http://localhost:3000/dashboard) to navigate to...
↓ Continue to next step

4 2 is Contact us
This step has no content

Customer starts with:

Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.

The more phrases you enter, the better your assistant can recognize what the customer wants.

Enter phrases your customer might use to start this action   Total: 5

Enter a phrase

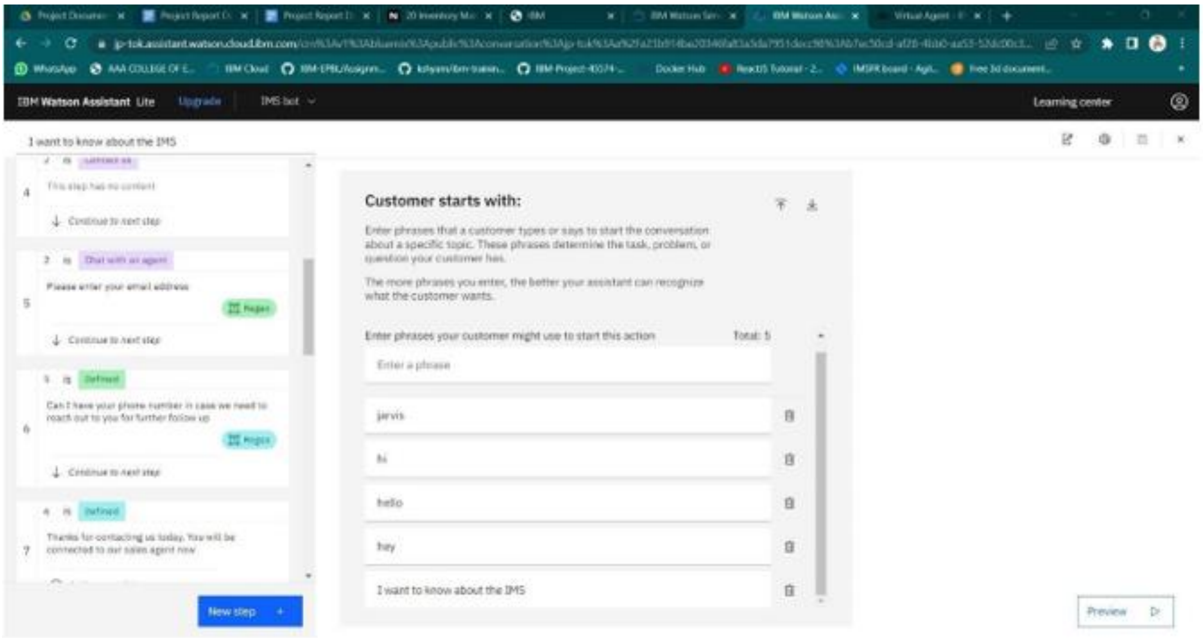
jarvis

hi

hello

hey

I want to know about the IMS



I want to know about the IMS

4 2 is Contact us
This step has no content
↓ Continue to next step

5 2 is Chat with an agent
Please enter your email address
Regex
↓ Continue to next step

6 5 is Defined
Can I have your phone number in case we need to reach out to you for further follow up
Regex
↓ Continue to next step

7 6 is Defined
Thanks for contacting us today. You will be connected to our sales agent now

Customer starts with:

Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.

The more phrases you enter, the better your assistant can recognize what the customer wants.

Enter phrases your customer might use to start this action   Total: 5

Enter a phrase

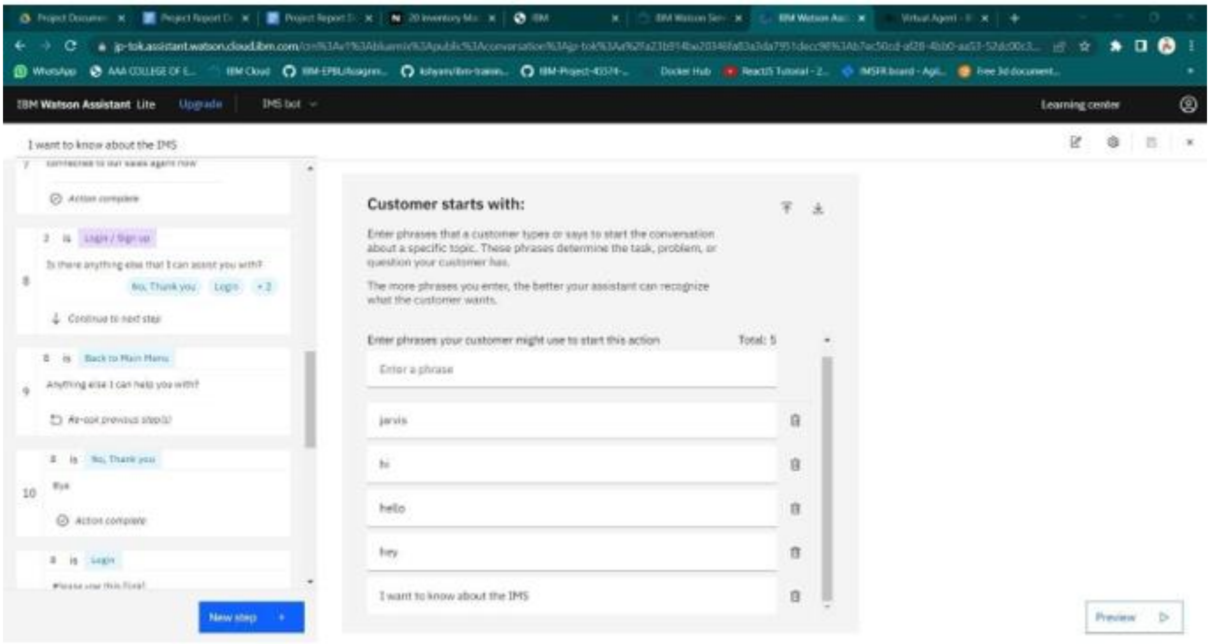
jarvis

hi

hello

hey

I want to know about the IMS



I want to know about the IMS

7 connected to our sales agent now

Action complete

8 — 2 is Login / Sign up

Is there anything else that I can assist you with?

No, Thank you · Login · +2

Continue to next step

9 — 8 is Back to Main Menu

Anything else I can help you with?

Re-ask previous step(s)

10 — 8 is No, Thank you

Bye

Action complete

8 is Login

Please use this [link]

Customer starts with:

Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.

The more phrases you enter, the better your assistant can recognize what the customer wants.

Enter phrases your customer might use to start this action — Total: 5

Enter a phrase

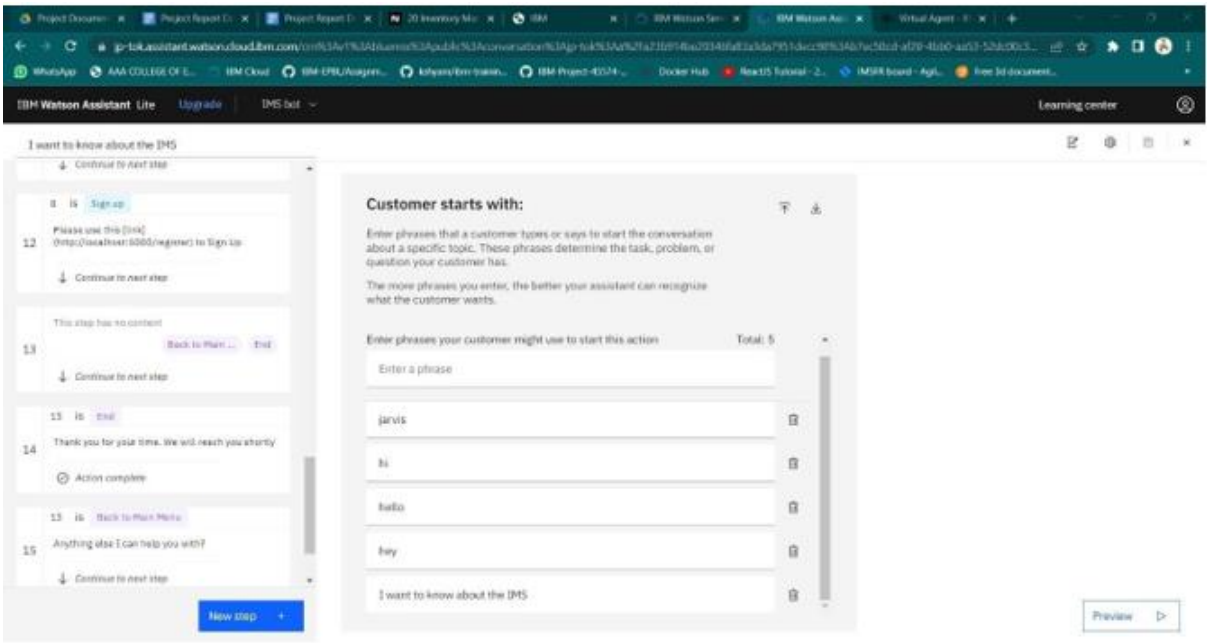
jarvis

hi

hello

hey

I want to know about the IMS



I want to know about the IMS

Continue to next step

12 — 8 is Sign up

Please use this [link](http://localhost:5000/register) to Sign Up

Continue to next step

13 — This step has no content

Back to Main ... · End

Continue to next step

14 — 13 is End

Thank you for your time. We will reach you shortly

Action complete

15 — 13 is Back to Main Menu

Anything else I can help you with?

Continue to next step

Customer starts with:

Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.

The more phrases you enter, the better your assistant can recognize what the customer wants.

Enter phrases your customer might use to start this action — Total: 5

Enter a phrase

jarvis

hi

hello

hey

I want to know about the IMS

Code:

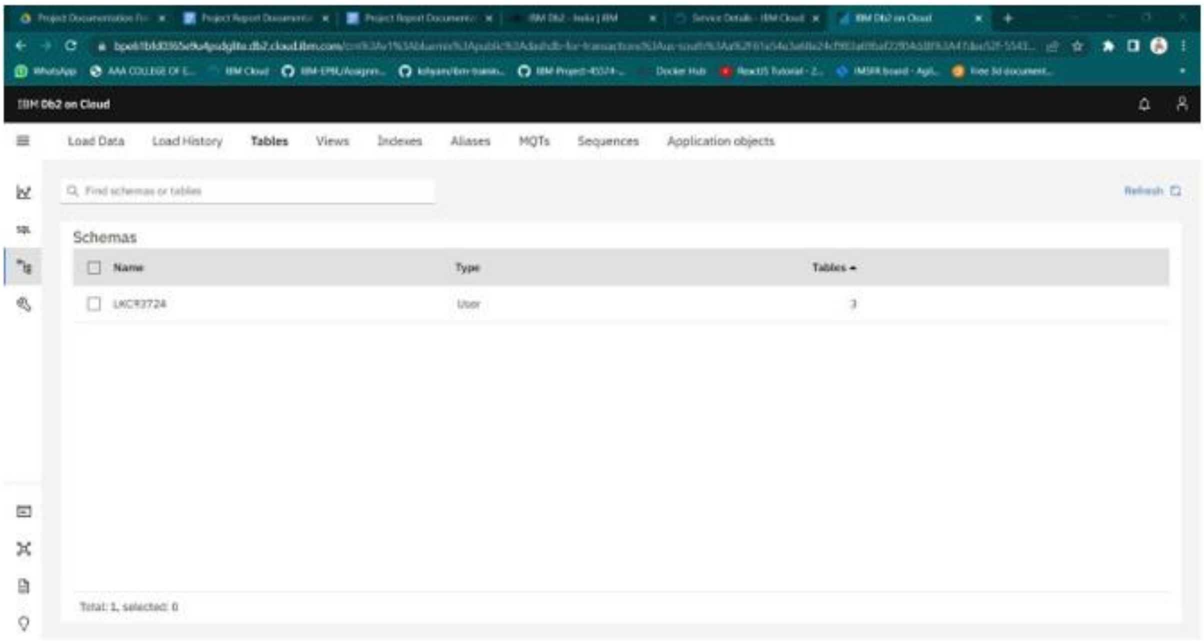
```
<script>
      window.watsonAssistantChatOptions = {
        integrationID: "633fc278-0dda-417b-9c10-bd2f300b411a", // The ID of this
        integration.region: "jp-tok", // The region your integration is hosted in.
        serviceInstanceID: "b7ec50cd-af28-4bb0-aa53-52dc00c34d4e", //
The ID of yourservice 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(t);
      });
    </script>
```

7.3 Database Schema(IBM DB2)

IBM Db2 Database on IBM Cloud combines a proven, AI-infused, enterprise-ready data management system with an integrated data and AI platform built on the security-rich, scalable Red Hat® OpenShift® foundation. Derive insights with machine learning embedded into query processing. Cut costs with the multimodal capability that eliminates the need for data replication and migration. And enhance agility by runningDb2 on any cloud vendor.

Code: (Connecting with IBM Db2 in app.py)

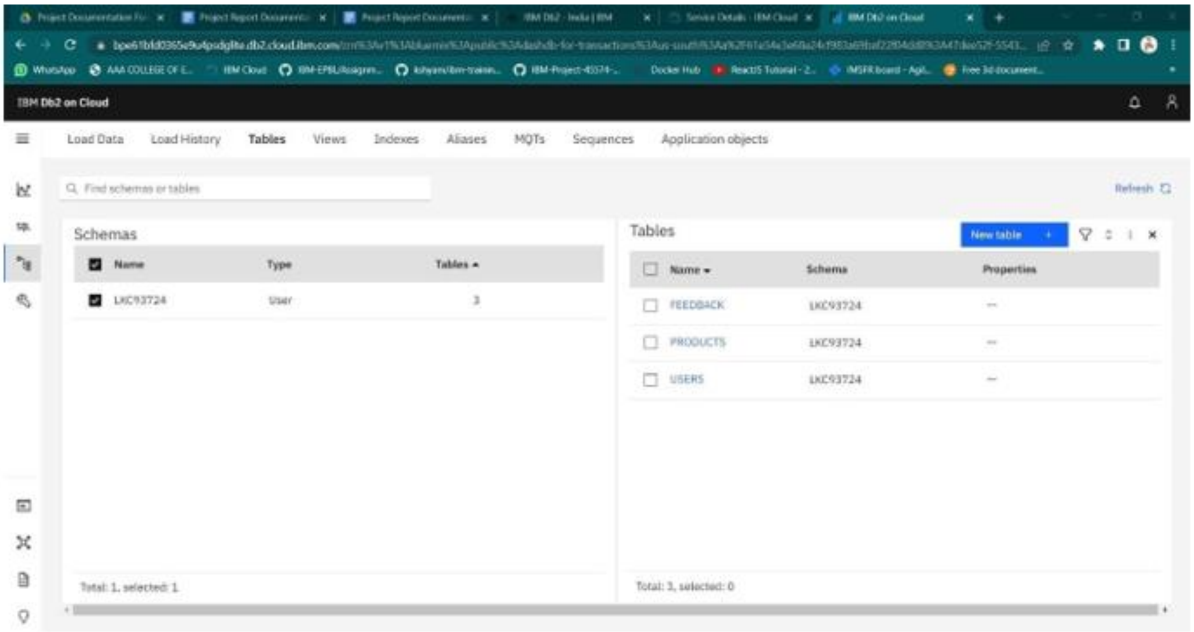
```
import ibm_db

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SECURITY
=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lpd12180;PWD=Rz1lC9DlMXNCH
NTj",'','')
```



Schemas

Name	Type	Tables ▲
□ LKC93724	User	3

Total: 1, selected: 0



Schemas

☑ Name	Type	Tables ▲
☑ LKC93724	User	3

Total: 1, selected: 1

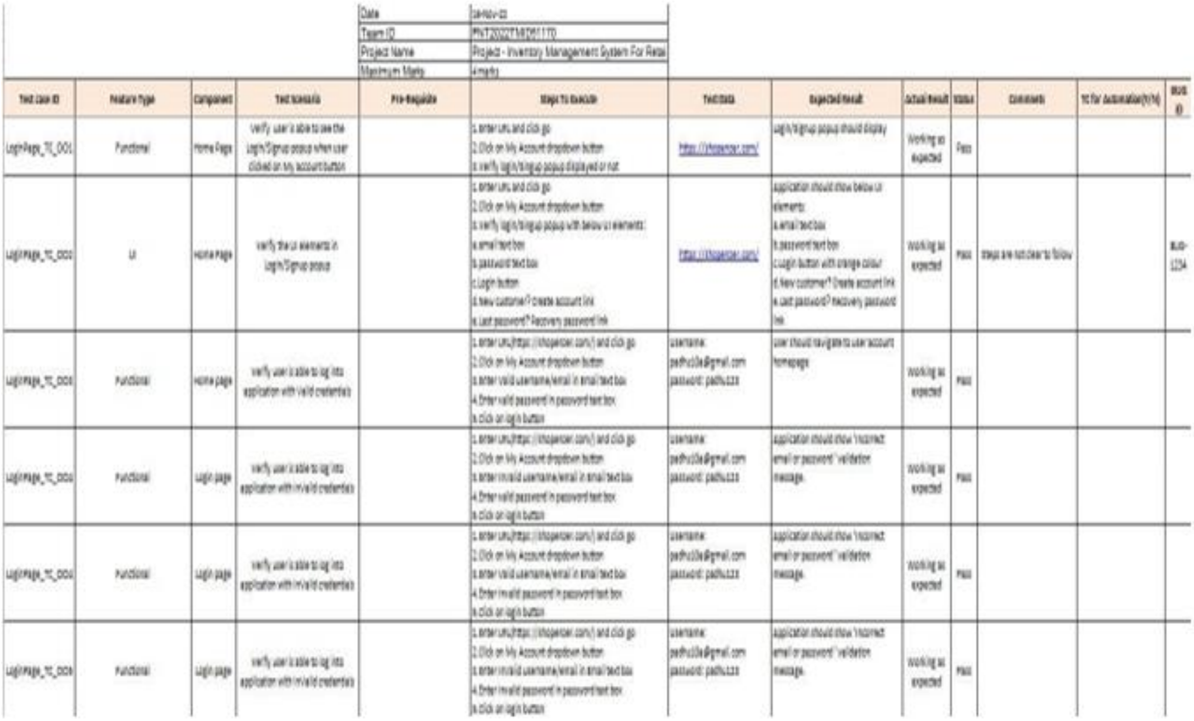
Tables

☐ Name ▼	Schema	Properties
☐ FEEDBACK	LKC93724	—
☐ PRODUCTS	LKC93724	—
☐ USERS	LKC93724	—

Total: 3, selected: 0

8. TESTING

8.1 Test Cases



Date	14-Nov-22
Team ID	PNT2022TMID51170
Project Name	Project - Inventory Management System For Retai
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	https://shopenzer.com/	Login/Signup popup should display	Working as expected	Pass			
LoginPage_TC_OO[illegible]	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	https://shopenzer.com/	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass	Steps are not clear to follow		BUG-1234
LoginPage_TC_OO[illegible]	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	username: padhu[illegible]@gmail.com password: padhu123	User should navigate to user account homepage	Working as expected	Pass			
LoginPage_TC_OO[illegible]	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	username: padhu[illegible]@gmail.com password: padhu123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass			
LoginPage_TC_OO[illegible]	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	username: padhu[illegible]@gmail.com password: padhu123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass			
LoginPage_TC_OO[illegible]	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	username: padhu[illegible]@gmail.com password: padhu123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass			

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the INVENTORY MANAGEMENT SYSTEM project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Seveíity 1	Seveíity 2	Seveíity 3	Seveíity4	Subtotal
By Design	10	4	2	3	19
Duplicate	1	0	3	0	4
Exteínal	2	3	0	1	6
Fixed	11	2	4	20	37
Not Repíoduced	0	0	1	0	1
Skipped	0	0	1	1	2

Won't Fix	0	5	2	1	8
I’otals	24	14	13	26	77

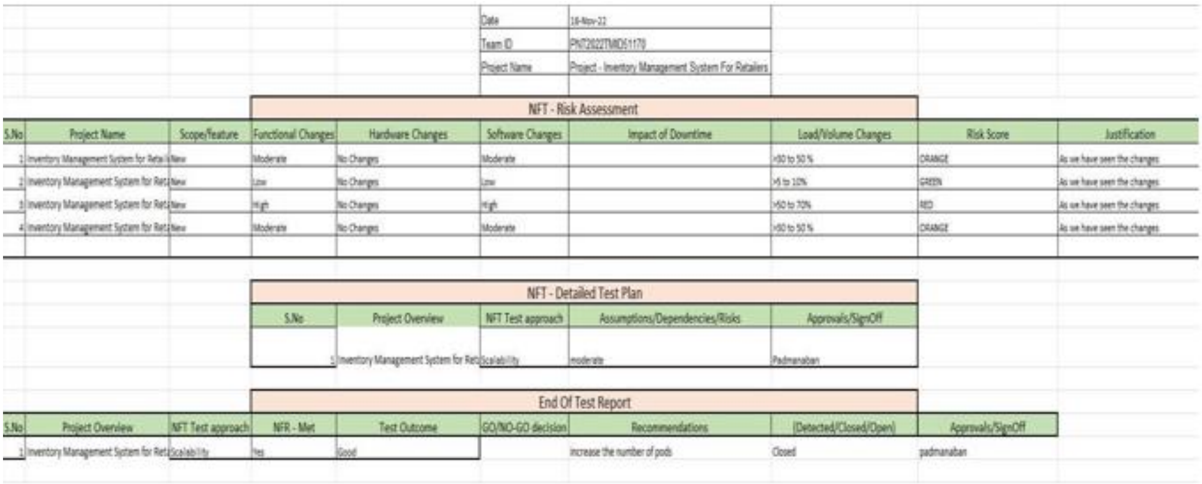
3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	I’otal Cases	Not I’ested	Ïail	Pass
Píint Engine	6	0	0	6
Client Application	25	0	0	20
Secuíity	2	0	0	2
Outsouíce Shipping	3	0	0	3
Exception Repoíting	7	0	0	7
Final Repoít Output	4	0	0	4
Veísion Contíol	2	0	0	2

9. RESULTS

9.1 Performance Metrics



Date	18-Nov-22
Team ID	PNT2022TMID51170
Project Name	Project - Inventory Management System For Retailers

NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Inventory Management System for Retai	New	Moderate	No Changes	Moderate		>30 to 50 %	ORANGE	As we have seen the changes
2	Inventory Management System for Ret	New	Low	No Changes	Low		>5 to 10%	GREEN	As we have seen the changes
3	Inventory Management System for Ret	New	High	No Changes	High		>50 to 70%	RED	As we have seen the changes
4	Inventory Management System for Ret	New	Moderate	No Changes	Moderate		>30 to 50 %	ORANGE	As we have seen the changes

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Inventory Management System for Ret	Scalability	moderate	Padmanaban

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	(Detected/Closed/Open)	Approvals/SignOff
1	Inventory Management System for Ret	Scalability	Yes	Good		increase the number of pods	Closed	padmanaban

10 ADVANTAGES & DISADVANTAGES

10.1 Advantages:

- **Better Inventory Accuracy:**
  With solid inventory management, you know what’s in stock and order only the amount of inventory you need to meet demand.

- **Reduced Risk of Overselling:**
  Inventory management helps track what’s in stock and what’s on backorder, so you don’t oversell products.

- **Cost Savings:**
  Stock costs money until it sells. Carrying costs include storage handling and transportation fees, insurance and employee salaries. Inventory is also at risk of theft, loss from natural disasters or obsolescence.

- **Avoiding Stockouts and Excess Stock:**
  Better planning and management helps a business minimize the number of days, if any, that an item is out of stock and avoid carrying too much inventory.

- **Greater Insights:**
  With inventory tracking and stock control, you can also easily spot sales trends or track recalled products or expiry dates.

- **More Productivity:**
  Good inventory management solutions save time that could be spent on other activities.

10.2 Disadvantages:

- **Expensive for Small Businesses:**
  The cost of inventory management software can seem daunting to a small business, but the investment often pays for itself in increased profits and improved customer loyalty. Additionally, cloud-based systems have made software that was once the domain of large enterprises available to smaller businesses.
- **Complex to Learn:**

Business software is sometimes tricky to learn.

However, managers can help by investing in online training to quickly bring users up to speed. Risk of System Crashes: Software does crash. However, you can remove the risk of data and productivity loss by using cloud-based platforms.
Malicious Hacks:
Malicious hacks are a risk to all businesses. The Internet of Things (IoT) adds even more complexity. Cloud-based software typically has greater security than a single company would offer on its own because of the risk a breach would have on the vendor.
Reduced Physical Audits:
When you automate some warehouse operations, it’s easy to skip a physical inventory check. Solve this by instituting regular audits.

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. As you have read above, there are different techniques that businesses can utilize to simplify and optimize stock management processes and control systems.

12. FUTURE SCOPE

- **Stock control for omnichannel retailing**
  Stores doing omnichannel retailing are at the top of their game; they attract the 90% of consumers who switch between at least three applications per day to complete specific tasks.

- **Inventories that power experiential retail**
  Experiential retail is a trend that's catching fire — especially in the past few months. In fact, they keep popping up in the news section of Google search results.
- **Advanced sales forecasting**
  Inventory management is big on having products on the shelf ready for shoppers when they need it. It curbs stockouts and fosters better personalization.
- **Season-based product recommendations**
  Speaking of Artificial Intelligence, AI-powered recommendation engines that adjust inventory based on real-time weather conditions and forecast are making their way to retail stores.

13 APPENDIX

Source Code

App.py

```
from turtle import st
from flask import Flask, render_template, request, redirect, url_for, session
from markupsafe import escape

import ibm_db
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SECURITY
=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lpd12180;PWD=Rz1lC9DlMXNCH
NTj","","")


import smtplib, ssl
## email.mime subclasses
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
## The pandas library is only for generating the current date, which is not necessary for
sending emails
import pandas as pd

from datetime import datetime

from flask import Flask


app = Flask(__name__)

var_list = []


app.secret_key = 'your secret key'

@app.route('/')
def home():
  if not session.get("name"):
      return render_template('home.html')
  return render_template('home.html', session = session)

@app.route('/register')
def new_student():
  return render_template('Register.html')


@app.route('/addrec',methods = ['POST', 'GET'])
def addrec():
  msg=''
  if request.method == 'POST':
    fname = request.form['fname']
    lname = request.form['lname']
    cname = request.form['cname']
    state = request.form['state']
    city = request.form['city']
    mobileno = request.form['mobileno']
    emailid = request.form['emailid']
    password = request.form['password']
    pincode = request.form['pincode']

    sql = "SELECT * FROM Users WHERE EMAILID =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,emailid)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)

    if account:
      msg="You are already a member, please login using your details"
      return render_template('Register.html', msg = msg)
    else:

      var_list.append(fname)
      var_list.append(lname)
      var_list.append(cname)
      var_list.append(state)
      var_list.append(city)
      var_list.append(mobileno)
      var_list.append(emailid)
      var_list.append(password)
      var_list.append(pincode)

      html= '''<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional //EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:o="urn:schemas-microsoft-com:office:office"><head><!–[if gte mso 9]><xml>
  <o:OfficeDocumentSettings>
    <o:AllowPNG/>
    <o:PixelsPerInch>96</o:PixelsPerInch>
  </o:OfficeDocumentSettings>
</xml>
<![endif]–>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="x-apple-disable-message-reformatting">
  <!–[if !mso]><!–>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <!–<![endif]–>
  <title></title>

  <style type="text/css">
    @media only screen and (min-width: 620px) {
      .u-row {
```

```
    width: 600px !important;
  }
  .u-row .u-col {
    vertical-align: top;
  }
  .u-row .u-col-100 {
    width: 600px !important;
  }
}

@media (max-width: 620px) {
  .u-row-container {
    max-width: 100% !important;
    padding-left: 0px !important;
    padding-right: 0px !important;
  }
  .u-row .u-col {
    min-width: 320px !important;
    max-width: 100% !important;
    display: block !important;
  }
  .u-row {
    width: calc(100% - 40px) !important;
  }
  .u-col {
    width: 100% !important;
  }
  .u-col>div {
    margin: 0 auto;
  }
}

body {
  margin: 0;
  padding: 0;
}

table,
tr,
td {
  vertical-align: top;
  border-collapse: collapse;
}

p {
  margin: 0;
}

.ie-container table,
.mso-container table {
  table-layout: fixed;
}
```

```
    * {
      line-height: inherit;
    }

    a[x-apple-data-detectors='true'] {
      color: inherit !important;
      text-decoration: none !important;
    }

    table,
    td {
      color: #000000;
    }

    #u_body a {
      color: #0000ee;
      text-decoration: underline;
    }
  </style>



  <!–[if !mso]><!–>
  <link href="https://fonts.googleapis.com/css?family=Cabin:400,700" rel="stylesheet" type="text/css">
  <!–<![endif]–>

</head>

<body class="clean-body u_body" style="margin: 0;padding: 0;-webkit-text-size-adjust: 100%;background-color: #f9f9f9;color: #000000">
  <!–[if IE]><div class="ie-container"><![endif]–>
  <!–[if mso]><div class="mso-container"><![endif]–>
  <table id="u_body" style="border-collapse: collapse;table-layout: fixed;border-spacing: 0;mso-table-lspace: 0pt;mso-table-rspace: 0pt;vertical-align: top;min-width: 320px;Margin: 0 auto;background-color: #f9f9f9;width:100%" cellpadding="0" cellspacing="0">
    <tbody>
      <tr style="vertical-align: top">
        <td style="word-break: break-word;border-collapse: collapse !important;vertical-align: top">
          <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0" border="0"><tr><td align="center" style="background-color: #f9f9f9;"><![endif]–>


          <div class="u-row-container" style="padding: 0px;background-color: transparent">
            <div class="u-row" style="Margin: 0 auto;min-width: 320px;max-width: 600px;overflow-wrap: break-word;word-wrap: break-word;word-break: break-word;background-color: transparent;">
              <div style="border-collapse: collapse;display: table;width: 100%;height: 100%;background-color: transparent;">
        <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0" border="0"><tr><td style="padding: 0px;background-color: transparent;" align="center"><table cellpadding="0" cellspacing="0" border="0" style="width:600px;"><tr style="background-color: transparent;"><![endif]–>

        <!–[if (mso)|(IE)]><td align="center" width="600" style="width: 600px;padding: 0px;border-top: 0px solid transparent;border-left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid transparent;" valign="top"><![endif]–>
        <div class="u-col u-col-100" style="max-width: 320px;min-width: 600px;display: table-cell;vertical-align: top;">
         <div style="height: 100%;width: 100% !important;">
          <!–[if (!mso)&(!IE)]><!–>
          <div style="height: 100%; padding: 0px;border-top: 0px solid transparent;border-left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid transparent;">
           <!–<![endif]–>

           <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0" cellspacing="0" width="100%" border="0">
            <tbody>
             <tr>
              <td style="overflow-wrap:break-word;word-break:break-word;padding:10px;font-family:'Cabin',sans-serif;" align="left">

               <div style="color: #afb0c7; line-height: 170%; text-align: center; word-wrap: break-word;">
                <p style="font-size: 14px; line-height: 170%;"><span style="font-size: 14px; line-height: 23.8px;">View Email in Browser</span></p>
               </div>

              </td>
             </tr>
            </tbody>
           </table>

           <!–[if (!mso)&(!IE)]><!–>
          </div>
          <!–<![endif]–>
         </div>
        </div>
        <!–[if (mso)|(IE)]></td><![endif]–>
        <!–[if (mso)|(IE)]></tr></table></td></tr></table><![endif]–>
       </div>
      </div>
     </div>



     <div class="u-row-container" style="padding: 0px;background-color: transparent">
      <div class="u-row" style="Margin: 0 auto;min-width: 320px;max-width: 600px;overflow-wrap: break-word;word-wrap: break-word;word-break: break-word;background-color: #ffffff;">
```

```
        <div style="border-collapse: collapse;display: table;width: 100%;height:
100%;background-color: transparent;">
         <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0"
border="0"><tr><td style="padding: 0px;background-color: transparent;" align="center"><table
cellpadding="0" cellspacing="0" border="0" style="width:600px;"><tr style="background-color:
#ffffff;"><![endif]–>

         <!–[if (mso)|(IE)]><td align="center" width="600" style="width: 600px;padding:
0px;border-top: 0px solid transparent;border-left: 0px solid transparent;border-right: 0px solid
transparent;border-bottom: 0px solid transparent;" valign="top"><![endif]–>
         <div class="u-col u-col-100" style="max-width: 320px;min-width: 600px;display:
table-cell;vertical-align: top;">
          <div style="height: 100%;width: 100% !important;">
           <!–[if (!mso)&(!IE)]><!–>
           <div style="height: 100%; padding: 0px;border-top: 0px solid transparent;border-
left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid
transparent;">
            <!–<![endif]–>

            <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
             <tbody>
              <tr>
               <td style="overflow-wrap:break-word;word-break:break-
word;padding:20px;font-family:'Cabin',sans-serif;" align="left">

                <table width="100%" cellpadding="0" cellspacing="0" border="0">
                 <tr>
                  <td style="padding-right: 0px;padding-left: 0px;" align="center">

                   <img align="center" border="0"
src="https://assets.unlayer.com/projects/111476/1667819721582-Secure%20login-
rafiki.png" alt="Image" title="Image" style="outline: none;text-decoration: none;-ms-
interpolation-mode: bicubic;clear: both;display: inline-block !important;border: none;height:
auto;float: none;width: 100%;max-width: 560px;"
                    width="560" />

                  </td>
                 </tr>
                </table>

               </td>
              </tr>
             </tbody>
            </table>

            <!–[if (!mso)&(!IE)]><!–>
           </div>
           <!–<![endif]–>
          </div>
         </div>
         <!–[if (mso)|(IE)]></td><![endif]–>
        <!–[if (mso)|(IE)]></tr></table></td></tr></table><![endif]–>
       </div>
      </div>
     </div>



     <div class="u-row-container" style="padding: 0px;background-color: transparent">
      <div class="u-row" style="Margin: 0 auto;min-width: 320px;max-width: 600px;overflow-
wrap: break-word;word-wrap: break-word;word-break: break-word;background-color:
#003399;">
       <div style="border-collapse: collapse;display: table;width: 100%;height:
100%;background-color: transparent;">
        <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0"
border="0"><tr><td style="padding: 0px;background-color: transparent;" align="center"><table
cellpadding="0" cellspacing="0" border="0" style="width:600px;"><tr style="background-color:
#003399;"><![endif]–>

        <!–[if (mso)|(IE)]><td align="center" width="600" style="width: 600px;padding:
0px;border-top: 0px solid transparent;border-left: 0px solid transparent;border-right: 0px solid
transparent;border-bottom: 0px solid transparent;" valign="top"><![endif]–>
        <div class="u-col u-col-100" style="max-width: 320px;min-width: 600px;display:
table-cell;vertical-align: top;">
         <div style="height: 100%;width: 100% !important;">
          <!–[if (!mso)&(!IE)]><!–>
          <div style="height: 100%; padding: 0px;border-top: 0px solid transparent;border-
left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid
transparent;">
           <!–<![endif]–>

           <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
            <tbody>
             <tr>
              <td style="overflow-wrap:break-word;word-break:break-word;padding:40px
10px 10px;font-family:'Cabin',sans-serif;" align="left">

               <table width="100%" cellpadding="0" cellspacing="0" border="0">
                <tr>
                 <td style="padding-right: 0px;padding-left: 0px;" align="center">

                  <img align="center" border="0"
src="https://cdn.templates.unlayer.com/assets/1597218650916-xxxxc.png" alt="Image"
title="Image" style="outline: none;text-decoration: none;-ms-interpolation-mode:
bicubic;clear: both;display: inline-block !important;border: none;height: auto;float: none;width:
26%;max-width: 150.8px;"
                   width="150.8" />

                 </td>
                </tr>
               </table>
              </td>
             </tr>
            </tbody>
           </table>

           <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
            <tbody>
             <tr>
              <td style="overflow-wrap:break-word;word-break:break-
word;padding:10px;font-family:'Cabin',sans-serif;" align="left">

               <div style="color: #e5eaf5; line-height: 140%; text-align: center; word-wrap:
break-word;">
                <p style="font-size: 14px; line-height: 140%;"><strong>T H A N K S   F O
R   S I G N I N G   U P !</strong></p>
               </div>

              </td>
             </tr>
            </tbody>
           </table>

           <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
            <tbody>
             <tr>
              <td style="overflow-wrap:break-word;word-break:break-word;padding:0px
10px 31px;font-family:'Cabin',sans-serif;" align="left">

               <div style="color: #e5eaf5; line-height: 140%; text-align: center; word-wrap:
break-word;">
                <p style="font-size: 14px; line-height: 140%;"><span style="font-size: 28px;
line-height: 39.2px;"><strong><span style="line-height: 39.2px; font-size: 28px;">Confirm Your
E-mail Address </span></strong>
                 </span>
                </p>
               </div>

              </td>
             </tr>
            </tbody>
           </table>

           <!–[if (!mso)&(!IE)]><!–>
          </div>
          <!–<![endif]–>
         </div>
        </div>
        <!–[if (mso)|(IE)]></td><![endif]–>
        <!–[if (mso)|(IE)]></tr></table></td></tr></table><![endif]–>
       </div>
```

```
      </div>
     </div>



     <div class="u-row-container" style="padding: 0px;background-color: transparent">
      <div class="u-row" style="Margin: 0 auto;min-width: 320px;max-width: 600px;overflow-wrap: break-word;word-wrap: break-word;word-break: break-word;background-color: #ffffff;">
       <div style="border-collapse: collapse;display: table;width: 100%;height: 100%;background-color: transparent;">
        <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0" border="0"><tr><td style="padding: 0px;background-color: transparent;" align="center"><table cellpadding="0" cellspacing="0" border="0" style="width:600px;"><tr style="background-color: #ffffff;"><![endif]–>

        <!–[if (mso)|(IE)]><td align="center" width="600" style="width: 600px;padding: 0px;border-top: 0px solid transparent;border-left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid transparent;" valign="top"><![endif]–>
        <div class="u-col u-col-100" style="max-width: 320px;min-width: 600px;display: table-cell;vertical-align: top;">
         <div style="height: 100%;width: 100% !important;">
          <!–[if (!mso)&(!IE)]><!–>
          <div style="height: 100%; padding: 0px;border-top: 0px solid transparent;border-left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid transparent;">
           <!–<![endif]–>

           <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0" cellspacing="0" width="100%" border="0">
            <tbody>
             <tr>
              <td style="overflow-wrap:break-word;word-break:break-word;padding:33px 55px;font-family:'Cabin',sans-serif;" align="left">

               <div style="line-height: 160%; text-align: center; word-wrap: break-word;">
                <p style="font-size: 14px; line-height: 160%;"><span style="font-size: 22px; line-height: 35.2px;">Hi, </span></p>
                <p style="font-size: 14px; line-height: 160%;"><span style="font-size: 18px; line-height: 28.8px;">You're almost ready to get started. Please click on the button below to confirm your email address and experience the awesome Inventory Management Service!</span></p>
               </div>

              </td>
             </tr>
            </tbody>
           </table>

           <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0" cellspacing="0" width="100%" border="0">
            <tbody>
             <tr>
```

```
                            <td style="overflow-wrap:break-word;word-break:break-
word;padding:10px;font-family:'Cabin',sans-serif;" align="left">

                              <!–[if mso]><style>.v-button {background: transparent
!important;}</style><![endif]–>
                              <div align="center">
                                <!–[if mso]><v:roundrect xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:w="urn:schemas-microsoft-com:office:word" href="http://localhost:5000/confirm"
style="height:46px; v-text-anchor:middle; width:157px;" arcsize="8.5%"  stroke="f"
fillcolor="#9ecbe4"><w:anchorlock/><center style="color:#103c55;font-family:'Cabin',sans-
serif;"><![endif]–>
                                <a href="http://159.122.175.228:30496/confirm" target="_blank" class="v-
button" style="box-sizing: border-box;display: inline-block;font-family:'Cabin',sans-serif;text-
decoration: none;-webkit-text-size-adjust: none;text-align: center;color: #103c55;
background-color: #9ecbe4; border-radius: 4px;-webkit-border-radius: 4px; -moz-border-
radius: 4px; width:auto; max-width:100%; overflow-wrap: break-word; word-break: break-
word; word-wrap:break-word; mso-border-alt: none;">
                                  <span style="display:block;padding:14px 44px 13px;line-
height:120%;"><span style="font-size: 16px; line-height: 19.2px;"><strong><span style="line-
height: 19.2px; font-size: 16px;">CONFIRM</span></strong>
                                  </span>
                                  </span>
                                </a>
                                <!–[if mso]></center></v:roundrect><![endif]–>
                              </div>

                            </td>
                          </tr>
                        </tbody>
                      </table>

                      <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
                        <tbody>
                          <tr>
                            <td style="overflow-wrap:break-word;word-break:break-word;padding:33px
55px 60px;font-family:'Cabin',sans-serif;" align="left">

                              <div style="color: #3598db; line-height: 160%; text-align: center; word-wrap:
break-word;">
                                <p style="line-height: 160%; font-size: 14px;"><span style="font-size: 18px;
line-height: 28.8px;">Once again, Thanks for signing up with us!</span></p>
                              </div>

                            </td>
                          </tr>
                        </tbody>
                      </table>

                      <!–[if (!mso)&(!IE)]><!–>
                    </div>
                    <!–<![endif]–>
```

```
         </div>
        </div>
        <!–[if (mso)|(IE)]></td><![endif]–>
        <!–[if (mso)|(IE)]></tr></table></td></tr></table><![endif]–>
       </div>
      </div>
     </div>


     <!–[if (mso)|(IE)]></td></tr></table><![endif]–>
    </td>
   </tr>
  </tbody>
 </table>
 <!–[if mso]></div><![endif]–>
 <!–[if IE]></div><![endif]–>
</body>

</html>'''


    # Set up the email addresses and password. Please replace below with your email address and password
    email_from = 'hariharanibm810@gmail.com'
    epassword = 'Ibmfrom@810'
    email_to = emailid

    # Generate today's date to be included in the email Subject
    date_str = pd.Timestamp.today().strftime('%Y-%m-%d')

    # Create a MIMEMultipart class, and set up the From, To, Subject fields
    email_message = MIMEMultipart()
    email_message['From'] = email_from
    email_message['To'] = email_to
    email_message['Subject'] = f'Report email - {date_str}'

    # Attach the html doc defined earlier, as a MIMEText html content type to the MIME message
    email_message.attach(MIMEText(html, "html"))
    # Convert it as a string
    email_string = email_message.as_string()

    # Connect to the Gmail SMTP server and Send Email
    context = ssl.create_default_context()
    with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
        server.login(email_from, epassword)
        server.sendmail(email_from, email_to, email_string)
    return render_template('notify.html')



@app.route('/confirm')
```

```
def confirnation():
   insert_sql = "INSERT INTO Users (FIRSTNAME, LASTNAME, COMPANYNAME, STATE, CITY,
MOBILENO, EMAILID, PASSWORD, PINCODE)  VALUES (?,?,?,?,?,?,?,?,?)"
   prep_stmt = ibm_db.prepare(conn, insert_sql)
   ibm_db.bind_param(prep_stmt, 1, var_list[0])
   ibm_db.bind_param(prep_stmt, 2, var_list[1])
   ibm_db.bind_param(prep_stmt, 3, var_list[2])
   ibm_db.bind_param(prep_stmt, 4, var_list[3])
   ibm_db.bind_param(prep_stmt, 5, var_list[4])
   ibm_db.bind_param(prep_stmt, 6, var_list[5])
   ibm_db.bind_param(prep_stmt, 7, var_list[6])
   ibm_db.bind_param(prep_stmt, 8, var_list[7])
   ibm_db.bind_param(prep_stmt, 9, var_list[8])
   ibm_db.execute(prep_stmt)
   var_list.clear()
   return render_template('confirm.html')


@app.route('/login')
def login():
  return render_template('Login.html')

@app.route('/loginrec', methods =['POST', 'GET'])
def loginrec():
    msg = ''
    if request.method == 'POST' and 'email' in request.form and 'password' in request.form:
        email = request.form['email']
        password = request.form['password']

        sql = "SELECT * FROM Users WHERE EMAILID =? AND PASSWORD =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            session['loggedin'] = True
            session['id'] = account['ID']
            session['email'] = account['EMAILID']
            session['name'] = account['FIRSTNAME']

            return render_template('dashboard/dashboard.html')
        else:
            msg = 'Incorrect email / password !'
    return render_template('login.html', msg = msg)

@app.route('/dashboard')
def dashboard():
  if session['loggedin'] == True:
     return render_template('dashboard/dashboard.html')
  else:
    return redirect(url_for('home'))

@app.route('/addproduct')
def addproduct():
  if session['loggedin'] == True:
     return render_template('dashboard/addproduct.html')
  else:
    return redirect(url_for('home'))

@app.route('/movement')
def movement():
  if session['loggedin'] == True:
    products = []
    sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, session['name'])
    ibm_db.execute(prep_stmt)
    dictionary = ibm_db.fetch_both(prep_stmt)
    while dictionary != False:
      # print ("The Name is : ",  dictionary)
      products.append(dictionary)
      dictionary = ibm_db.fetch_both(prep_stmt)

    if products:
      return render_template("dashboard/movement.html", products = products , session =
session)
    else:
      return render_template("dashboard/movement.html")
  else:
    return redirect(url_for('home'))

@app.route('/moveproc',methods = ['POST', 'GET'])
def moveproc():
  if request.method == 'POST':
        pname = request.form['pname']
        quantityout = request.form['quantityout']
        tow = request.form['to']

        insert_sql = "UPDATE products SET QUANTITYOUT = ?, TO = ? WHERE PRODUCTNAME
= ? AND HOLDERNAME = ?;"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1,quantityout)
        ibm_db.bind_param(prep_stmt, 2, tow)
        ibm_db.bind_param(prep_stmt, 3, pname)
        ibm_db.bind_param(prep_stmt, 4, session['name'])
        ibm_db.execute(prep_stmt)
        select_sql="SELECT QUANTITYIN from PRODUCTS WHERE PRODUCTNAME = ? AND
HOLDERNAME = ?;"
        pre_stmt = ibm_db.prepare(conn, select_sql)
        ibm_db.bind_param(pre_stmt, 1, pname)
        ibm_db.bind_param(pre_stmt, 2, session['name'])
        ibm_db.execute(pre_stmt)
    outofstock = ibm_db.fetch_both(pre_stmt)
    if outofstock['QUANTITYIN'] <= int(quantityout):

      html= ''' <!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional //EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:o="urn:schemas-microsoft-com:office:office">

<head>
 <!–[if gte mso 9]>
<xml>
 <o:OfficeDocumentSettings>
  <o:AllowPNG/>
  <o:PixelsPerInch>96</o:PixelsPerInch>
 </o:OfficeDocumentSettings>
</xml>
<![endif]–>
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <meta name="x-apple-disable-message-reformatting">
 <!–[if !mso]><!–>
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <!–<![endif]–>
 <title></title>

 <style type="text/css">
  @media only screen and (min-width: 620px) {
   .u-row {
    width: 600px !important;
   }
   .u-row .u-col {
    vertical-align: top;
   }
   .u-row .u-col-100 {
    width: 600px !important;
   }
  }

  @media (max-width: 620px) {
   .u-row-container {
    max-width: 100% !important;
    padding-left: 0px !important;
    padding-right: 0px !important;
   }
   .u-row .u-col {
    min-width: 320px !important;
    max-width: 100% !important;
    display: block !important;
   }
   .u-row {
    width: calc(100% - 40px) !important;
   }
```

```
    .u-col {
      width: 100% !important;
    }
    .u-col>div {
      margin: 0 auto;
    }
  }

  body {
    margin: 0;
    padding: 0;
  }

  table,
  tr,
  td {
    vertical-align: top;
    border-collapse: collapse;
  }

  p {
    margin: 0;
  }

  .ie-container table,
  .mso-container table {
    table-layout: fixed;
  }

  * {
    line-height: inherit;
  }

  a[x-apple-data-detectors='true'] {
    color: inherit !important;
    text-decoration: none !important;
  }

  table,
  td {
    color: #000000;
  }

  @media (max-width: 480px) {
    #u_column_3 .v-col-background-color {
      background-color: #3598db !important;
    }
  }
</style>
```

```
  <!–[if !mso]><!–>
  <link href="https://fonts.googleapis.com/css?family=Cabin:400,700" rel="stylesheet"
type="text/css">
  <!–<![endif]–>

</head>

<body class="clean-body u_body" style="margin: 0;padding: 0;-webkit-text-size-adjust:
100%;background-color: #f9f9f9;color: #000000">
  <!–[if IE]><div class="ie-container"><![endif]–>
  <!–[if mso]><div class="mso-container"><![endif]–>
  <table style="border-collapse: collapse;table-layout: fixed;border-spacing: 0;mso-table-
lspace: 0pt;mso-table-rspace: 0pt;vertical-align: top;min-width: 320px;Margin: 0
auto;background-color: #f9f9f9;width:100%" cellpadding="0" cellspacing="0">
    <tbody>
      <tr style="vertical-align: top">
        <td style="word-break: break-word;border-collapse: collapse !important;vertical-align:
top">
          <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0"
border="0"><tr><td align="center" style="background-color: #f9f9f9;"><![endif]–>


          <div class="u-row-container" style="padding: 0px;background-color: transparent">
            <div class="u-row" style="Margin: 0 auto;min-width: 320px;max-width: 600px;overflow-
wrap: break-word;word-wrap: break-word;word-break: break-word;background-color: #ffffff;">
              <div style="border-collapse: collapse;display: table;width: 100%;height:
100%;background-color: transparent;">
                <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0"
border="0"><tr><td style="padding: 0px;background-color: transparent;" align="center"><table
cellpadding="0" cellspacing="0" border="0" style="width:600px;"><tr style="background-color:
#ffffff;"><![endif]–>

                <!–[if (mso)|(IE)]><td align="center" width="600" class="v-col-background-color"
style="width: 600px;padding: 0px;border-top: 0px solid transparent;border-left: 0px solid
transparent;border-right: 0px solid transparent;border-bottom: 0px solid transparent;"
valign="top"><![endif]–>
                <div class="u-col u-col-100" style="max-width: 320px;min-width: 600px;display:
table-cell;vertical-align: top;">
                  <div class="v-col-background-color" style="height: 100%;width: 100% !important;">
                    <!–[if (!mso)&(!IE)]><!–>
                    <div style="height: 100%; padding: 0px;border-top: 0px solid transparent;border-
left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid
transparent;">
                      <!–<![endif]–>

                      <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
                        <tbody>
                          <tr>
                            <td style="overflow-wrap:break-word;word-break:break-
word;padding:0px;font-family:'Cabin',sans-serif;" align="left">
```

```
               <table width="100%" cellpadding="0" cellspacing="0" border="0">
                <tr>
                 <td style="padding-right: 0px;padding-left: 0px;" align="right">

                  <img align="right" border="0"
src="https://assets.unlayer.com/projects/111476/1668447888843-Warning-cuate.png"
alt="Image" title="Image" style="outline: none;text-decoration: none;-ms-interpolation-mode:
bicubic;clear: both;display: inline-block !important;border: none;height: auto;float: none;width:
100%;max-width: 600px;"
                   width="600" />

                 </td>
                </tr>
               </table>

              </td>
             </tr>
            </tbody>
           </table>

           <!–[if (!mso)&(!IE)]><!–>
          </div>
          <!–<![endif]–>
         </div>
        </div>
        <!–[if (mso)|(IE)]></td><![endif]–>
        <!–[if (mso)|(IE)]></tr></table></td></tr></table><![endif]–>
       </div>
      </div>
     </div>



     <div class="u-row-container" style="padding: 0px;background-color: transparent">
      <div class="u-row" style="Margin: 0 auto;min-width: 320px;max-width: 600px;overflow-
wrap: break-word;word-wrap: break-word;word-break: break-word;background-color:
#003399;">
       <div style="border-collapse: collapse;display: table;width: 100%;height:
100%;background-color: transparent;">
        <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0"
border="0"><tr><td style="padding: 0px;background-color: transparent;" align="center"><table
cellpadding="0" cellspacing="0" border="0" style="width:600px;"><tr style="background-color:
#003399;"><![endif]–>

        <!–[if (mso)|(IE)]><td align="center" width="600" class="v-col-background-color"
style="background-color: #3598db;width: 600px;padding: 0px;border-top: 0px solid
transparent;border-left: 0px solid transparent;border-right: 0px solid transparent;border-
bottom: 0px solid transparent;" valign="top"><![endif]–>
        <div id="u_column_3" class="u-col u-col-100" style="max-width: 320px;min-width:
600px;display: table-cell;vertical-align: top;">
         <div class="v-col-background-color" style="background-color: #3598db;height:
100%;width: 100% !important;">
```

```
          <!–[if (!mso)&(!IE)]><!–>
          <div style="height: 100%; padding: 0px;border-top: 0px solid transparent;border-
left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid
transparent;">
           <!–<![endif]–>

           <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
            <tbody>
             <tr>
              <td style="overflow-wrap:break-word;word-break:break-word;padding:40px
10px 10px;font-family:'Cabin',sans-serif;" align="left">

               <table width="100%" cellpadding="0" cellspacing="0" border="0">
                <tr>
                 <td style="padding-right: 0px;padding-left: 0px;" align="center">

                  <img align="center" border="0"
src="https://assets.unlayer.com/projects/111476/1668448081560-vecteezy_warning-icon-
png-transparent_9663941_592.png" alt="Image" title="Image" style="outline: none;text-
decoration: none;-ms-interpolation-mode: bicubic;clear: both;display: inline-block
!important;border: none;height: auto;float: none;width: 30%;max-width: 174px;"
                   width="174" />

                 </td>
                </tr>
               </table>

              </td>
             </tr>
            </tbody>
           </table>

           <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
            <tbody>
             <tr>
              <td style="overflow-wrap:break-word;word-break:break-
word;padding:10px;font-family:'Cabin',sans-serif;" align="left">

               <div style="color: #e5eaf5; line-height: 140%; text-align: center; word-wrap:
break-word;">
                <p style="font-size: 14px; line-height: 140%;"><span style="color: #000000;
font-size: 20px; line-height: 28px;">WARNING</span></p>
               </div>

              </td>
             </tr>
            </tbody>
           </table>
```

```
                      <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
                        <tbody>
                          <tr>
                            <td style="overflow-wrap:break-word;word-break:break-word;padding:0px
10px 31px;font-family:'Cabin',sans-serif;" align="left">

                              <div style="color: #e5eaf5; line-height: 140%; text-align: center; word-wrap:
break-word;">
                                <p style="font-size: 14px; line-height: 140%;"><span style="font-size: 24px;
line-height: 33.6px;"><strong>You are <span style="color: #000000; font-size: 24px; line-
height: 33.6px;">Out of Stock !</span></strong>
                                  </span>
                                </p>
                              </div>

                            </td>
                          </tr>
                        </tbody>
                      </table>

                      <!–[if (!mso)&(!IE)]><!–>
                    </div>
                    <!–<![endif]–>
                  </div>
                </div>
                <!–[if (mso)|(IE)]></td><![endif]–>
                <!–[if (mso)|(IE)]></tr></table></td></tr></table><![endif]–>
              </div>
            </div>
          </div



          <div class="u-row-container" style="padding: 0px;background-color: transparent">
            <div class="u-row" style="Margin: 0 auto;min-width: 320px;max-width: 600px;overflow-
wrap: break-word;word-wrap: break-word;word-break: break-word;background-color: #ffffff;">
              <div style="border-collapse: collapse;display: table;width: 100%;height:
100%;background-color: transparent;">
                <!–[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0"
border="0"><tr><td style="padding: 0px;background-color: transparent;" align="center"><table
cellpadding="0" cellspacing="0" border="0" style="width:600px;"><tr style="background-color:
#ffffff;"><![endif]–>

                <!–[if (mso)|(IE)]><td align="center" width="600" class="v-col-background-color"
style="background-color: #000000;width: 600px;padding: 0px;border-top: 0px solid
transparent;border-left: 0px solid transparent;border-right: 0px solid transparent;border-
bottom: 0px solid transparent;" valign="top"><![endif]–>
                <div class="u-col u-col-100" style="max-width: 320px;min-width: 600px;display:
table-cell;vertical-align: top;">
                  <div class="v-col-background-color" style="background-color: #000000;height:
100%;width: 100% !important;">
```

```
                <!–[if (!mso)&(!IE)]><!–>
                <div style="height: 100%; padding: 0px;border-top: 0px solid transparent;border-
left: 0px solid transparent;border-right: 0px solid transparent;border-bottom: 0px solid
transparent;">
                 <!–<![endif]–>

                 <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
                  <tbody>
                   <tr>
                    <td style="overflow-wrap:break-word;word-break:break-word;padding:33px
55px;font-family:'Cabin',sans-serif;" align="left">

                     <div style="color: #000000; line-height: 160%; text-align: center; word-wrap:
break-word;">
                      <p style="font-size: 14px; line-height: 160%;"><span style="font-size: 18px;
line-height: 28.8px; color: #ecf0f1;">Please order <span style="color: #3598db; font-size:
18px; line-height: 28.8px;">new stocks </span>to get rid of the <span style="color: #3598db;
font-size: 18px; line-height: 28.8px;">out-of-stock</span>.</span>
                      </p>
                     </div>

                    </td>
                   </tr>
                  </tbody>
                 </table>

                 <table style="font-family:'Cabin',sans-serif;" role="presentation" cellpadding="0"
cellspacing="0" width="100%" border="0">
                  <tbody>
                   <tr>
                    <td style="overflow-wrap:break-word;word-break:break-word;padding:33px
55px 60px;font-family:'Cabin',sans-serif;" align="left">

                     <div style="color: #3598db; line-height: 160%; text-align: center; word-wrap:
break-word;">
                      <p style="line-height: 160%; font-size: 14px; text-align: center;"><span
style="font-size: 18px; line-height: 28.8px; color: #3598db;">Post queries in the Contact
Support for further clearance!</span></p>
                      <p style="line-height: 160%; font-size: 14px; text-align: center;"> </p>
                      <p style="line-height: 160%; font-size: 14px; text-align: center;"><span
style="font-size: 18px; line-height: 28.8px; color: #3598db;">Thank you!</span></p>
                     </div>

                    </td>
                   </tr>
                  </tbody>
                 </table>

                 <!–[if (!mso)&(!IE)]><!–>
                </div>
                <!–<![endif]–>
```

```
              </div>
            </div>
            <!–[if (mso)|(IE)]></td><![endif]–>
            <!–[if (mso)|(IE)]></tr></table></td></tr></table><![endif]–>
          </div>
        </div>
      </div>


      <!–[if (mso)|(IE)]></td></tr></table><![endif]–>
    </td>
  </tr>
  </tbody>
 </table>
 <!–[if mso]></div><![endif]–>
 <!–[if IE]></div><![endif]–>
</body>

</html>

        '''



        # Set up the email addresses and password. Please replace below with your email
address and password
        email_from = 'hariharanibm810@gmail.com'
        epassword = 'Ibmfrom@810'
        email_to = session['email']

        # Generate today's date to be included in the email Subject
        date_str = pd.Timestamp.today().strftime('%d-%m-%Y')

        # Create a MIMEMultipart class, and set up the From, To, Subject fields
        email_message = MIMEMultipart()
        email_message['From'] = email_from
        email_message['To'] = email_to
        email_message['Subject'] = f'Warning!!! {pname} - Out Of Stock - {date_str}'

        # Attach the html doc defined earlier, as a MIMEText html content type to the MIME
message
        email_message.attach(MIMEText(html, "html"))
        # Convert it as a string
        email_string = email_message.as_string()

        # Connect to the Gmail SMTP server and Send Email
        context = ssl.create_default_context()
        with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
            server.login(email_from, epassword)
            server.sendmail(email_from, email_to, email_string)
        products = []
        sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, session['name'])
        ibm_db.execute(prep_stmt)
        dictionary = ibm_db.fetch_both(prep_stmt)
        while dictionary != False:
            # print ("The Name is : ",  dictionary)
            products.append(dictionary)
            dictionary = ibm_db.fetch_both(prep_stmt)
        return render_template('dashboard/movement.html', msg = "Product movement
noted!", products = products)
    else:
        products = []
        sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, session['name'])
        ibm_db.execute(prep_stmt)
        dictionary = ibm_db.fetch_both(prep_stmt)
        while dictionary != False:
            # print ("The Name is : ",  dictionary)
            products.append(dictionary)
            dictionary = ibm_db.fetch_both(prep_stmt)
        return render_template('dashboard/movement.html', msg = "Product movement
noted!", products = products)



@app.route('/report')
def report():
 if session['loggedin'] == True:
  products = []
  stockonhand = []
  sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
  prep_stmt = ibm_db.prepare(conn, sql)
  ibm_db.bind_param(prep_stmt, 1, session['name'])
  ibm_db.execute(prep_stmt)
  dictionary = ibm_db.fetch_both(prep_stmt)
  while dictionary != False:
  # print ("The Name is : ",  dictionary)
  products.append(dictionary)
  dictionary = ibm_db.fetch_both(prep_stmt)

  for i in products:
    calc = int((i['QUANTITYIN'])) - int(i['QUANTITYOUT'])
    stockonhand.append(str(calc))

  return render_template('dashboard/report.html', row_row1 =zip(products,stockonhand))
 else:
  return redirect(url_for('home'))

@app.route('/stockupdate')
def stock():
 if session['loggedin'] == True:
  products = []
```

```
    sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, session['name'])
    ibm_db.execute(prep_stmt)
    dictionary = ibm_db.fetch_both(prep_stmt)
    while dictionary != False:
      # print ("The Name is : ",  dictionary)
      products.append(dictionary)
      dictionary = ibm_db.fetch_both(prep_stmt)

    if products:
      return render_template("dashboard/stockupdate.html", products = products , session =
session)
    else:
      return render_template("dashboard/stockupdate.html")
  else:
    return redirect(url_for('home'))


@app.route('/proc_delete', methods = ['POST', 'GET'])
def proc_delete():
        id = request.args.get('pid')
        delete_sql = "DELETE FROM products WHERE ID = ? AND HOLDERNAME = ?;"
        prep_stmt = ibm_db.prepare(conn, delete_sql)
        ibm_db.bind_param(prep_stmt, 1, id)
        ibm_db.bind_param(prep_stmt, 2, session['name'])
        ibm_db.execute(prep_stmt)


        products = []
        sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, session['name'])
        ibm_db.execute(prep_stmt)
        dictionary = ibm_db.fetch_both(prep_stmt)
        while dictionary != False:
          # print ("The Name is : ",  dictionary)
          products.append(dictionary)
          dictionary = ibm_db.fetch_both(prep_stmt)
        return render_template('dashboard/stockupdate.html', msg='Product successfully
deleted!' , products = products)

@app.route('/proc_update', methods = ['POST', 'GET'])
def proc_update():
        if request.method == 'POST':
          pname = request.form['pname']
          quantityin = request.form['quantityin']
          pid = request.form['pid']
        update_sql = "UPDATE products SET PRODUCTNAME = ?, QUANTITYIN = ? WHERE ID
= ? AND HOLDERNAME = ?;"
        prep_stmt = ibm_db.prepare(conn, update_sql)
        ibm_db.bind_param(prep_stmt, 1, pname)
        ibm_db.bind_param(prep_stmt, 2, quantityin)
        ibm_db.bind_param(prep_stmt, 3, pid)
        ibm_db.bind_param(prep_stmt, 4, session['name'])
        ibm_db.execute(prep_stmt)


        products = []
        sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, session['name'])
        ibm_db.execute(prep_stmt)
        dictionary = ibm_db.fetch_both(prep_stmt)
        while dictionary != False:
          # print ("The Name is : ",  dictionary)
          products.append(dictionary)
          dictionary = ibm_db.fetch_both(prep_stmt)
        return render_template('dashboard/stockupdate.html', msg='Product successfully
updated!' , products = products)

@app.route('/addproc',methods = ['POST', 'GET'])
def addproc():
    if request.method == 'POST':
      pname = request.form['pname']
      quantity = request.form['quantity']
      the_time = datetime.now()
      the_time = the_time.replace(second=0, microsecond=0)

      sql = "SELECT * FROM Products WHERE HOLDERNAME =?"
      stmt = ibm_db.prepare(conn, sql)
      ibm_db.bind_param(stmt,1,session['name'])
      ibm_db.execute(stmt)
      product = ibm_db.fetch_assoc(stmt)
      if product:

       if product['PRODUCTNAME']==pname:

        return render_template('dashboard/addproduct.html', msg="Product already added!
Add a new product.")
       else:
        sql ="INSERT INTO Products
(PRODUCTNAME,QUANTITYIN,QUANTITYOUT,TO,DATE,HOLDERNAME) VALUES (?,?,?,?,?,?);"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, pname)
        ibm_db.bind_param(prep_stmt, 2, quantity)
        ibm_db.bind_param(prep_stmt, 3, '0')
        ibm_db.bind_param(prep_stmt, 4, '')
        ibm_db.bind_param(prep_stmt, 5, str(the_time))
        ibm_db.bind_param(prep_stmt, 6, session['name'])
        ibm_db.execute(prep_stmt)
        return render_template('dashboard/addproduct.html', msg="Product added")
      else:
        sql ="INSERT INTO Products
(PRODUCTNAME,QUANTITYIN,QUANTITYOUT,TO,DATE,HOLDERNAME) VALUES (?,?,?,?,?,?);"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, pname)
        ibm_db.bind_param(prep_stmt, 2, quantity)
        ibm_db.bind_param(prep_stmt, 3, '0')
        ibm_db.bind_param(prep_stmt, 4, '')
        ibm_db.bind_param(prep_stmt, 5, str(the_time))
        ibm_db.bind_param(prep_stmt, 6, session['name'])
        ibm_db.execute(prep_stmt)
        return render_template('dashboard/addproduct.html', msg="Product added")


@app.route('/productlist')
def productlist():
  if session['loggedin'] == True:
    products = []
    sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, session['name'])
    ibm_db.execute(prep_stmt)
    dictionary = ibm_db.fetch_both(prep_stmt)
    while dictionary != False:
      # print ("The Name is : ",  dictionary)
      products.append(dictionary)
      dictionary = ibm_db.fetch_both(prep_stmt)

    if products:
      return render_template("dashboard/productlist.html", products = products , session =
session)
    else:
      return render_template("dashboard/productlist.html")
  else:
    return redirect(url_for('home'))

@app.route('/contactsupport')
def contactsupport():
  if session['loggedin'] == True:
    return render_template('dashboard/contactsupport.html')
  else:
    return redirect(url_for('home'))


@app.route('/contactsup', methods = ['POST','GET'])
def contactsup():
  if request.method == 'POST':
      name = request.form['name']
      mobileno = request.form['mobileno']
      emailid = request.form['emailid']
      query = request.form['query']
      html = "<h1>Query from, </h1><br/><b>Name: </b>"+name+"<br/><b>Email ID:
</b>"+emailid+"<br/><b>Contact no: </b>"+mobileno+"<br/><b>Query:
</b><b>"+query+"</b>"

     # Set up the email addresses and password. Please replace below with your email
address and password
      email_from = 'hariharanibm810@gmail.com'
      epassword = 'Ibmfrom@810'
      email_to = 'imsapp0810@gmail.com'

     # Generate today's date to be included in the email Subject
      date_str = pd.Timestamp.today().strftime('%Y-%m-%d')

     # Create a MIMEMultipart class, and set up the From, To, Subject fields
      email_message = MIMEMultipart()
      email_message['From'] = email_from
      email_message['To'] = email_to
      email_message['Subject'] = f'Query email - {date_str}'

     # Attach the html doc defined earlier, as a MIMEText html content type to the MIME
message
      email_message.attach(MIMEText(html, "html"))
     # Convert it as a string
      email_string = email_message.as_string()

     # Connect to the Gmail SMTP server and Send Email
      context = ssl.create_default_context()
      with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
        server.login(email_from, epassword)
        server.sendmail(email_from, email_to, email_string)
      return render_template('dashboard/contactsupport.html', msg = "We have mailed your
query to our Support team! Soon they will reach you.")

@app.route('/feedback')
def feedback():
  if session['loggedin'] == True:
      return render_template('dashboard/feedback.html')
  else:
    return redirect(url_for('home'))

@app.route('/feedbackadd', methods = ['POST','GET'])
def feedbackadd():
  if request.method == 'POST':
      interface = request.form['interface']
      availability = request.form['availability']
      userfriendly = request.form['userfriendly']
      chatbot = request.form['chatbot']
      suggest = request.form['suggest']

      sql = "SELECT * FROM Feedback WHERE NAME =?"
      stmt = ibm_db.prepare(conn, sql)
      ibm_db.bind_param(stmt,1,session['name'])
```

```
      ibm_db.execute(stmt)
      account = ibm_db.fetch_assoc(stmt)

      if account:
        return render_template('dashboard/feedback.html', msg = "Your feedback was
submitted already.")
      else:
        ins_sql = "INSERT INTO Feedback
(interface,availability,userfriendly,chatbot,suggest,name) VALUES (?,?,?,?,?,?);"
        prep_stmt = ibm_db.prepare(conn, ins_sql)
        ibm_db.bind_param(prep_stmt, 1, interface)
        ibm_db.bind_param(prep_stmt, 2, availability)
        ibm_db.bind_param(prep_stmt, 3, userfriendly)
        ibm_db.bind_param(prep_stmt, 4, chatbot)
        ibm_db.bind_param(prep_stmt, 5, suggest)
        ibm_db.bind_param(prep_stmt, 6, session['name'])
        ibm_db.execute(prep_stmt)

        return render_template('dashboard/feedback.html', msg = "Your feedback was
submitted.")

@app.route('/logout')
def logout():
    session['loggedin'] = False
    session.pop('id', None)
    session.pop('email', None)
    session.pop('name', None)
    return redirect(url_for('home'))

if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=5000)
```

Dockerfile

```
FROM python:3.10.6
WORKDIR /app
COPY requirements.txt ./
RUN pip install -r requirements.txt
COPY . .
EXPOSE 5000
CMD ["python","./app.py"]
```

requirement.txt

```
flask
ibm_db
pandas
```

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-25852-1659975112

View the deployed ims-final flask application by clicking the below link:

http://169.51.207.113:30496/