



INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

IBM PROJECT

Submitted By

B.NAWFAL KHAN	(815119104031)
V.SABAREESON	(815119104037)
R.AJITH KUMAR	(815119104002)
K.SUTHARSAN	(815119104045)
J.FELIX	(815119104014)

*in partial fulfilment for the award of the
degree of*

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

DHANALAKSHMI SRINIVASAN INSTITUTE OF TECHNOLOGY,



TRICHY - 621112

ANNA UNIVERSITY: CHENNAI 600025

NOVEMBER 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**INVENTORY MANAGEMENT SYSTEM FOR RETAILERS**” is bonafide work of **V.SABAREESON (815119104037), R.AJITH KUMAR(815119104002),K.SUTHARSAN(8151191040045),J.FELIX(815119104014) and B.NAWFAL KHAN (815119104031)**, who carried out the project work under my supervision.

SIGNATURE

Mr. S. SASHIKUMAR, M.E.,
HEAD OF THE DEPARTMENT,

Assistant professor,

Department of Computer Science and
Engineering,

Dhanalakshmi Srinivasan Institute of
Technology,

Trichy- 621 112

SIGNATURE

Ms. R. PADMAVATHI, M.E.,
SUPERVISOR,

Assistant professor,

Department of Computer Science and
Engineering,

Dhanalakshmi Srinivasan Institute of
Technology,

Trichy- 621 112

Submitted for the project Viva-Voice held on_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First of all, we thank the Almighty and our parents for blessings without which we couldn't complete the project work within stipulated time.

We are indebted to our respected
Chairman, Shri. A. Srinivasan,
for his noble guidance to accomplish the project successfully.

This project would have been a distant reality if not for the help and encouragement from various people. We take immense pleasure in thanking our **Principal Dr. S. Selvakumar, M.E., Ph.D.,** Dhanalakshmi Srinivasan Institute of Technology for having permitted us to carry out this project.

I would like to express gratitude to Dr. V. Raju, MCA., Ph.D., for his timely guidance and providing all necessary facilities at the right time for completing this project.

We thank our honourable **Head of the Department Mr. S. Sashikumar, M.E.,** Department of Computer Science and Engineering, Dhanalakshmi Srinivasan Institute of Technology for going through the report and giving valuable suggestions.

We would like to take an opportunity to express our sincere thanks to our Institutional Guidance **Ms. R. Padmavathi, M.E.,** Assistant Professor, Dhanalakshmi Srinivasan Institute of Technology for the valuable suggestion given during a project work.

Project Report - Inventory Management System for Retailers

Team ID: PNT2022TMID46138

Team Lead:NAWFAL KHAN B

Team Member 1: SABAREESON V

Team Member 2: AJITH KUMAR R

Team Member 3: SUTHARSAN K

Team Member 4: FELIX J

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

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

In practice, effective retail inventory management results in lower costs and a better understanding of sales patterns. Retail inventory management tools and methods give retailers more information on which to run their businesses. Applications have been developed to help retailers track and manage stocks related to their own products. The System will ask retailers to create their accounts by providing essential details. Retailers can access their accounts by logging into the application.

Once retailers successfully log in to the application they can update their inventory details, also users will be able to add new stock by submitting essential details related to the stock. They can view details of the current inventory. The System will automatically send an email alert to the retailers if there is no stock found in their accounts. So that they can order new stock.

1.2 Purpose

The main purpose of inventory management is to help businesses easily and efficiently manage the ordering, stocking, storing, and using of inventory. By effectively managing your inventory, you'll always know what items are in stock, how many of them there are, and where they are located.

Plus, practicing strong inventory management allows you to understand how you use your inventory—and how demand changes for it—over time. You can zero in on exactly what you need, what's not so important, and what's just a waste of money. That's using inventory management to practice inventory control. By the way, inventory control is the balancing act of always having enough stock to meet demand, while spending as little as possible on ordering and carrying inventory.

2. LITERATURE SURVEY

2.1 Existing problem

Inconsistent Tracking:

Using manual inventory tracking procedures across different software and spreadsheets is time-consuming, redundant and vulnerable to errors. Even small businesses can benefit from a centralized inventory tracking system that includes accounting features.

Warehouse Efficiency:

Inventory management controls at the warehouse is labor-intensive and involves several steps, including receiving and, picking, packing and shipping. The challenge is to perform all these tasks in the most efficient way possible.

Inaccurate Data:

You need to know, at any given moment, exactly what inventory you have. Gone are the days when inventory could be counted once a year with an all-hands-on-deck approach.

Changing Demand:

Customer demand is constantly shifting. Keeping too much could result in obsolete inventory you're unable to sell, while keeping too little could leave you unable to fulfill customer orders. Order strategies for core items, as well as technology to create and execute an inventory plan, can help compensate for changing demand.

Limited Visibility:

When your inventory is hard to identify or locate in the warehouse, it leads to incomplete, inaccurate or delayed shipments. Receiving and finding the right stock is vital to efficient warehouse operations and positive customer experiences.

Manual Documentation:

Managing inventory with paperwork and manual processes is tedious and not secure. And it doesn't easily scale across multiple warehouses with lots of stock.

Problem Stock:

Perishable and fragile stock need specialized plans for care and storage. And high-value inventory needs specific loss-prevention strategies and inventory controls.

Supply Chain Complexity:

Global supply chains shift daily, placing a burden on your inventory planning and management operations. The manufacturers and wholesale distributors that dictate when, where and how your inventory ships require flexibility and offer unpredictable lead times.

Managing Warehouse Space:

Efficiently managing space is an intimidating task. Planning and designing warehouse spaces with inventory management platforms helps you better control the timing of new stock deliveries. It can account for important factors, such as available space. Read more about the differences between warehouse management and inventory management.

INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

LITERATURE SURVEY

SI.NO	TITLE	TECHNIQUES	MERITS	DEMERITS
1.	SORTLY	They pride themselves as being the most simple inventory app with barcode scanner	It also allows team members the ability to see what we have on and at a moments notice	It's missing some core functionality that would be helpful
2.	INFLOW INVENTORY	It can be either deployed on-premise or hosted on the cloud.	Multi-channel management	Iphone app missing some functionality
3.	SOS INVENTORY	Automated order processing	SOS Inventory features is that it allows food traceability and helps manage lots and batches	Slow to implement
4.	INVENTORY NOW	Track the inventory throughout their product cycle	Charge your customers directly from the app	Pay for features And payment issues
5.	TESTSERIES SOFTWARE INVENTORY	Follows FIFO Method(first in first out)	Easy to use and value for money	Late responsive

2.3 Problem Statement Definition

I am	I'm trying to	But	Because	Which makes me feel
Pc assembler	Automatic discovery of computers and other IT assets	Security protection to prevent tampering or unauthorized software	A free trial or a demo package that offers an opportunity for a cost-free assessment	frustrated & tensed

INVENTORY MANAGEMENT SYSTEM FOR RETAILERS PROBLEM STATEMENTS

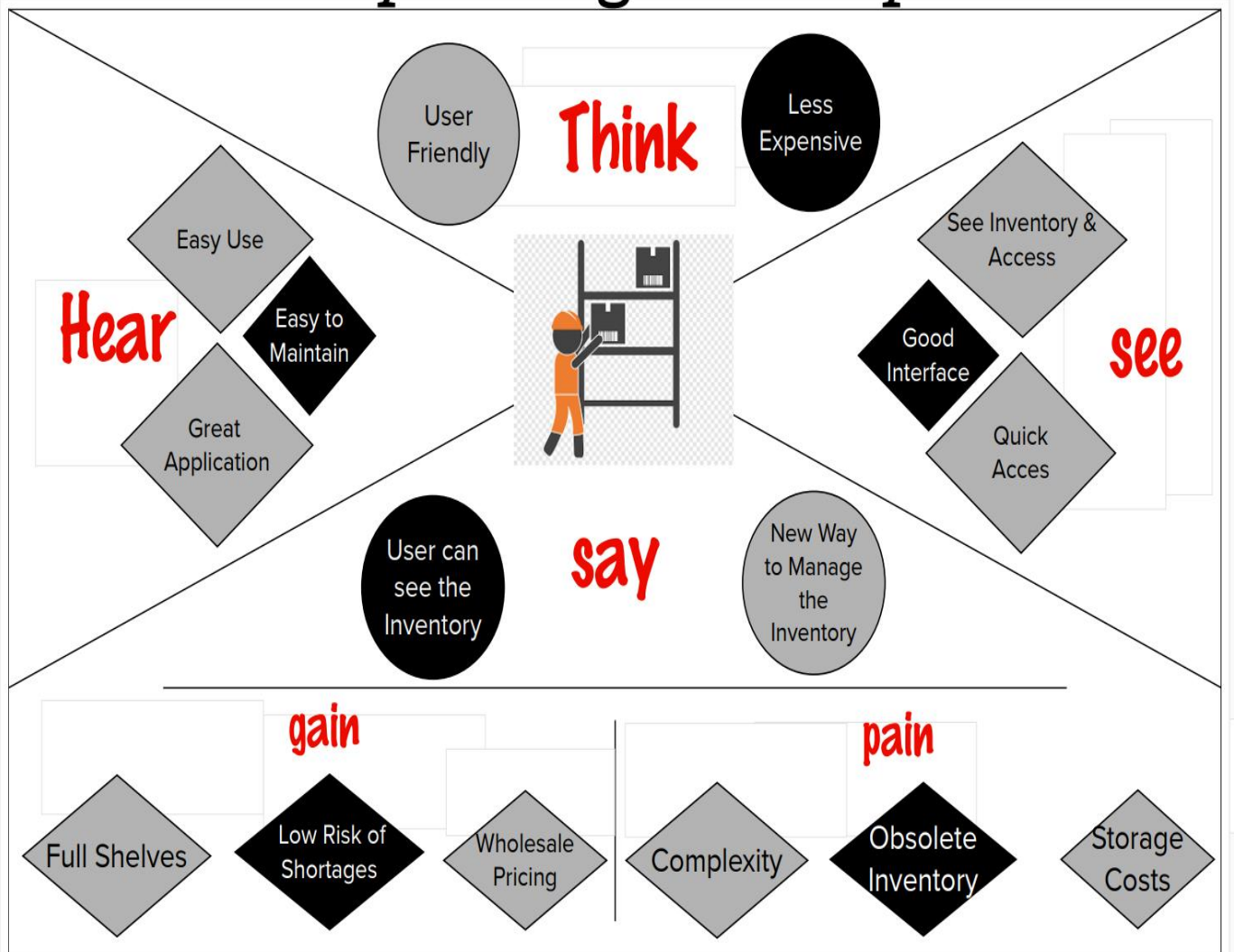
I am	I'm trying to	But	Because	Which makes me feel
Warehouse supervisor	Tracks items in real time as they are received, stored, picked, packed and shipped	Advises on inventory replenishment.	Guides inventory receiving and put-away	Tired & stressed

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Pc assembler	Automatic discovery of computers and other IT assets	Security protection to prevent tampering or unauthorized software	A free trial or a demo package that offers an opportunity for a cost-free assessment	frustrated & tensed
PS-2	Warehouse supervisor	Tracks items in real time as they are received, stored, picked, packed and shipped	Advises on inventory replenishment.	Guides inventory receiving and put-away	Tired & stressed

3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Inventory Management System



3.2 Ideation & Brainstorming

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



Brainstorm & Idea prioritization

Brainstorming is a problem-solving activity where students build on or develop higher order thinking skills. Encourages creative thought. Brainstorming encourages students to think creatively.

PROBLEM STATEMENT

The problem faced by the company is they do not have any systematic system to record and keep their inventory data. It is difficult for the admin to record the inventory data quickly and safely because they only keep it in the logbook and not properly organized.



Key rules of brainstorming

To run an exercise, read and put out the cards.

- Stay in step in. Encourage wild ideas.
- Don't just go out. Listen to others.
- Go for it in it. If possible, be visual.



BRAIN STORM

Nawfal khan

Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming

Felix

Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming

Sabareeson

Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming

Ajith kumar

Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming

Sutharsan

Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming
Brainstorming	Brainstorming	Brainstorming

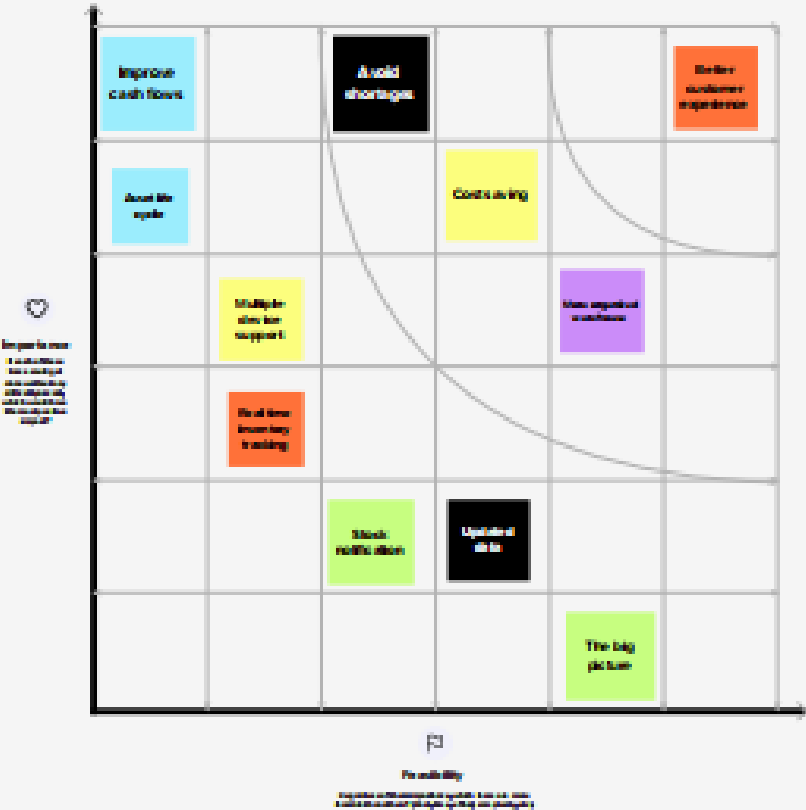
Step-2: Brainstorm, Idea Listing and Grouping

GROUP IDEA



Step-3: Idea Prioritization

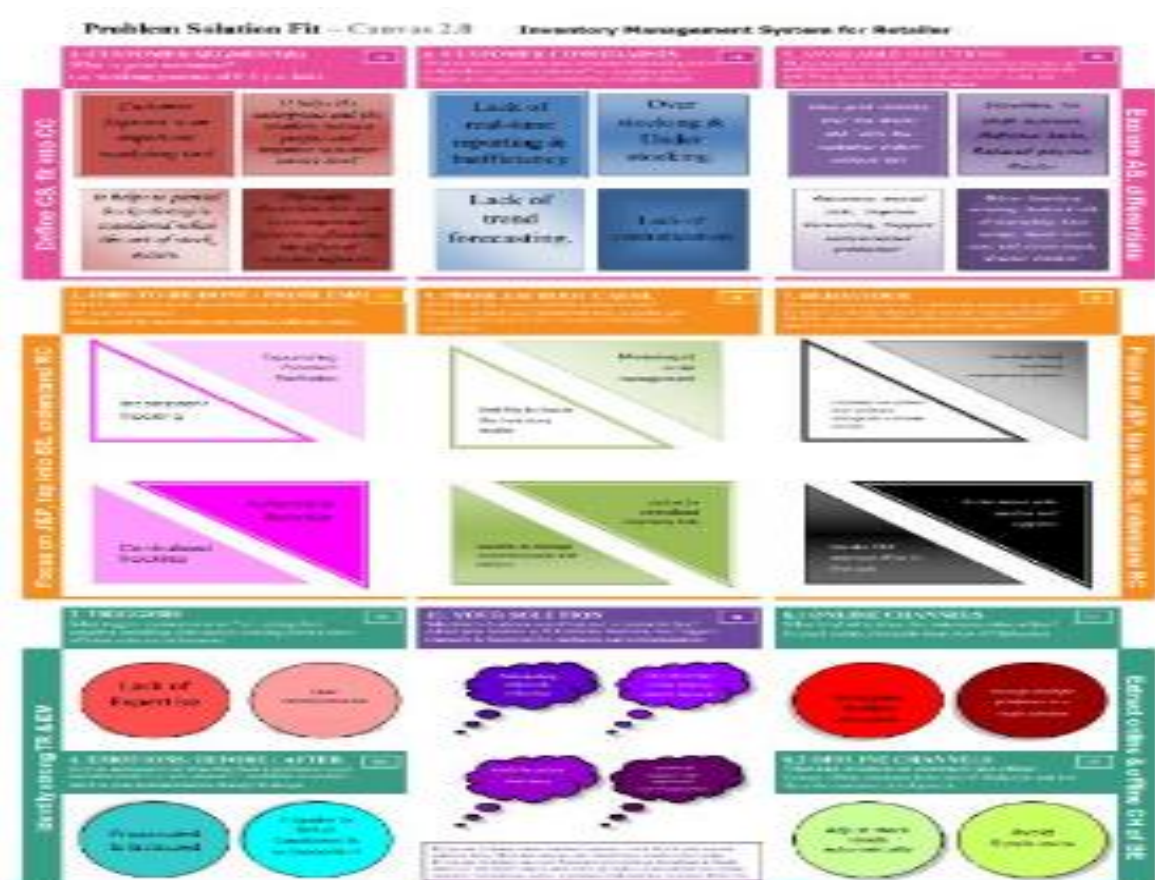
IDEA PRIORITIZATION



3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Retailers do not have any systematic system to record and keep their inventory data.
2.	Idea / Solution description	To develop a cloud web application that will help retailers to manage, track, and control their stocks.
3.	Novelty / Uniqueness	Real time inventory tracking and secured user interactions.
4.	Social Impact / Customer Satisfaction	Better interface to understand the tracking of stocks and better reliability over stock management.
5.	Business Model (Revenue Model)	This cloud web application will get higher usage and acceptance in market and among people of this generation.
6.	Scalability of the Solution	9 out of 10

3.4 Problem Solution fit



4 REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Username/Email-ID Login with Password
FR-4	Admin Login	Login with Username/Email-ID Login with Password
FR-5	Inventory Management	Track quantity of products present in inventory at any instant
FR-6	Tracking of stock	Notifications through Email

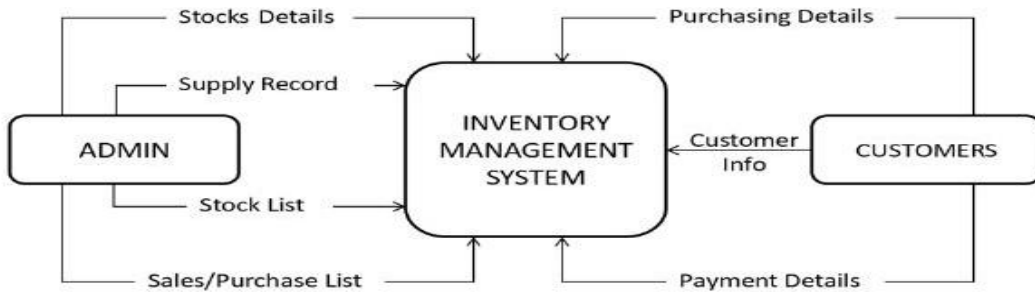
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This cloud web application makes the process of inventory management a lot easier which saves money and time both. This system is highly responsive to both desktop and mobile users.
NFR-2	Security	Inventory security aims to prevent inventory losses – for example, due to incorrect storage, theft, or incorrect incoming goods inspection – so that the correct stock is always available.

NFR-3	Reliability	The availability of products should be properly updated for customer satisfaction. The out-of-stock information should be notified. The system must give accurate inventory status to the user continuously.
NFR-4	Performance	The companies have to design and operate materials management and product distribution functions effectively. Inventory control systems enable a business to determine and maintain an optimum level of investment in inventory in order to achieve the required operational performance.
NFR-5	Availability	The software will be available only to the administrator of the organization and the product, as well as customer details, will be recorded by him. He can manage the inventory.
NFR-6	Scalability	The System can manage large inventory and provides quick access to the inventory in no time.

5 PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

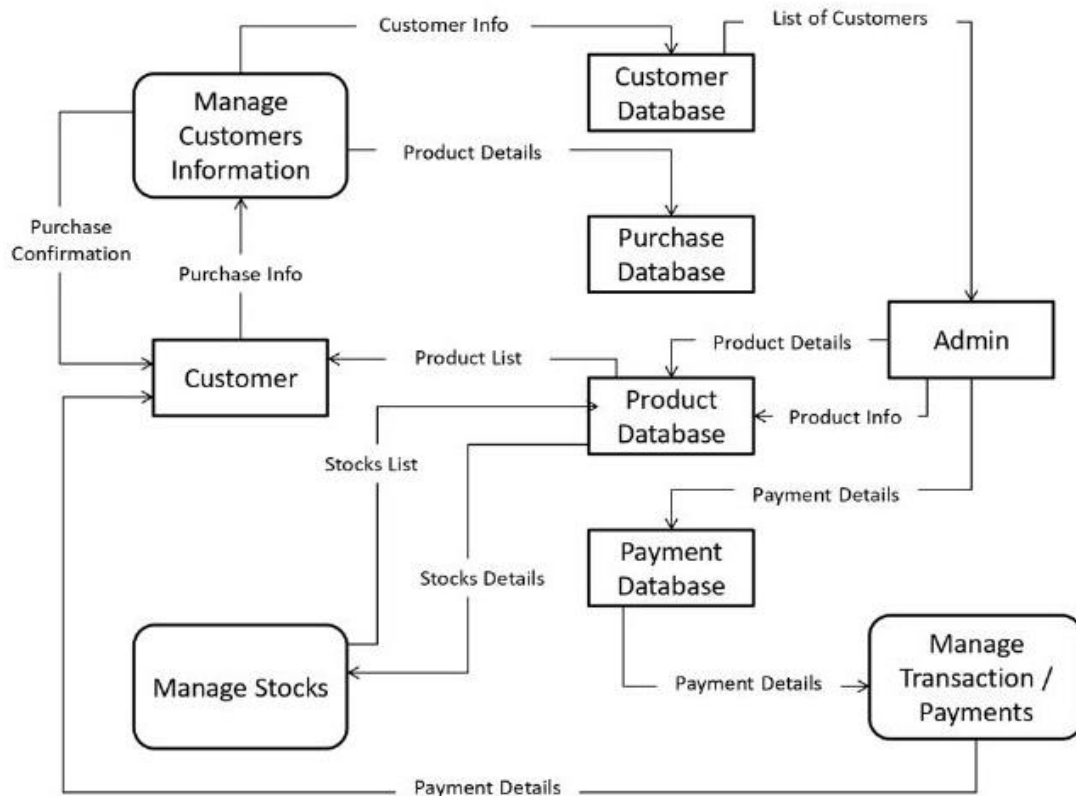


Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	User interacts with the cloud web application.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson Assistant
4.	Database	Data Type, Configurations	MySQL
5.	Cloud Database	Database Service on Cloud	IBM DB2
6.	File Storage	File storage requirements	IBM Object Storage
7.	Infrastructure (Server / Cloud)	Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management	Kubernetes

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Micro web framework, written in Python	Flask
2.	Security Implementations	List all the security/access controls implemented, use of firewalls, etc.	SHA-256
3.	Scalable Architecture	Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management.	Kubernetes
4.	Availability	Docker CLI stores its configuration files in a directory called .docker within your \$HOME directory.	Docker CLI
5.	Performance	To send alerts to users based on their stock	Sendgrid

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application	I can receive a confirmation email & click confirm	Medium	Sprint-1
	Confirmation	USN-3	As a user, I will confirm the registration once I have received the email from the application	I can get a confirmation for my email and password and create an authenticated account.	Medium	Sprint-1
	Login	USN-4	As a user, I can log in to the application through Gmail & Password	I can log onto the application with a verified email and password	High	Sprint-1
	Dashboard	USN-5	As a user, I can view the dashboard of the application by entering my email & password	Once I log on to the application, I can view products to buy.	High	Sprint-2
	Add items to the cart	USN-6	As a user, I can add the products I wish to buy to the carts.	As a user, I can buy any product or add it to my cart for buying later	Medium	Sprint-2
	Stock Update	USN-7	As a user, I can add products that are not available in the dashboard to the stock list.	If any of the products are not available, as a user I can update the inventory and send mail to the owner	Medium	Sprint-3
Customer (Web user)	Registration	USN-8	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-9	As a user, I will receive a confirmation email once I have registered for the application	I can receive a confirmation email & click confirm	Medium	Sprint-1
	Confirmation	USN-10	As a user, I will confirm the registration once I have received the email from the application	I can get a confirmation for my email and password	Medium	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
				and create an authenticated account.		
	Login	USN-11	As a user, I can log in to the application through Gmail & Password	I can log onto the application with a verified email and password	High	Sprint-1
	Dashboard	USN-12	As a user, I can view the dashboard of the application by entering my email & password	Once I log on to the application, I can view products to buy.	High	Sprint-2
	Add items to the cart	USN-13	As a user, I can add the products I wish to buy to the carts.	As a user, I can buy any product or add it to my cart for buying later	Medium	Sprint-2
	Stock Update	USN-14	As a user, I can add products that are not available in the dashboard to the stock list.	If any of the products are not available, as a user I can update the inventory and send mail to the owner	Medium	Sprint-3
Customer Care Executive	Request to Customer Care	USN-15	As a user, I can contact the Customer Care Executive and request any services I want from customer care.	As a user, I can contact Customer Care and get support.	Low	Sprint-4
Administrator	Contact Administrator	USN-16	I can be able to report any difficulties I experience as a report	As a user, and I can give my support in possible ways to the administrator and the administration.	Medium	Sprint-4

6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

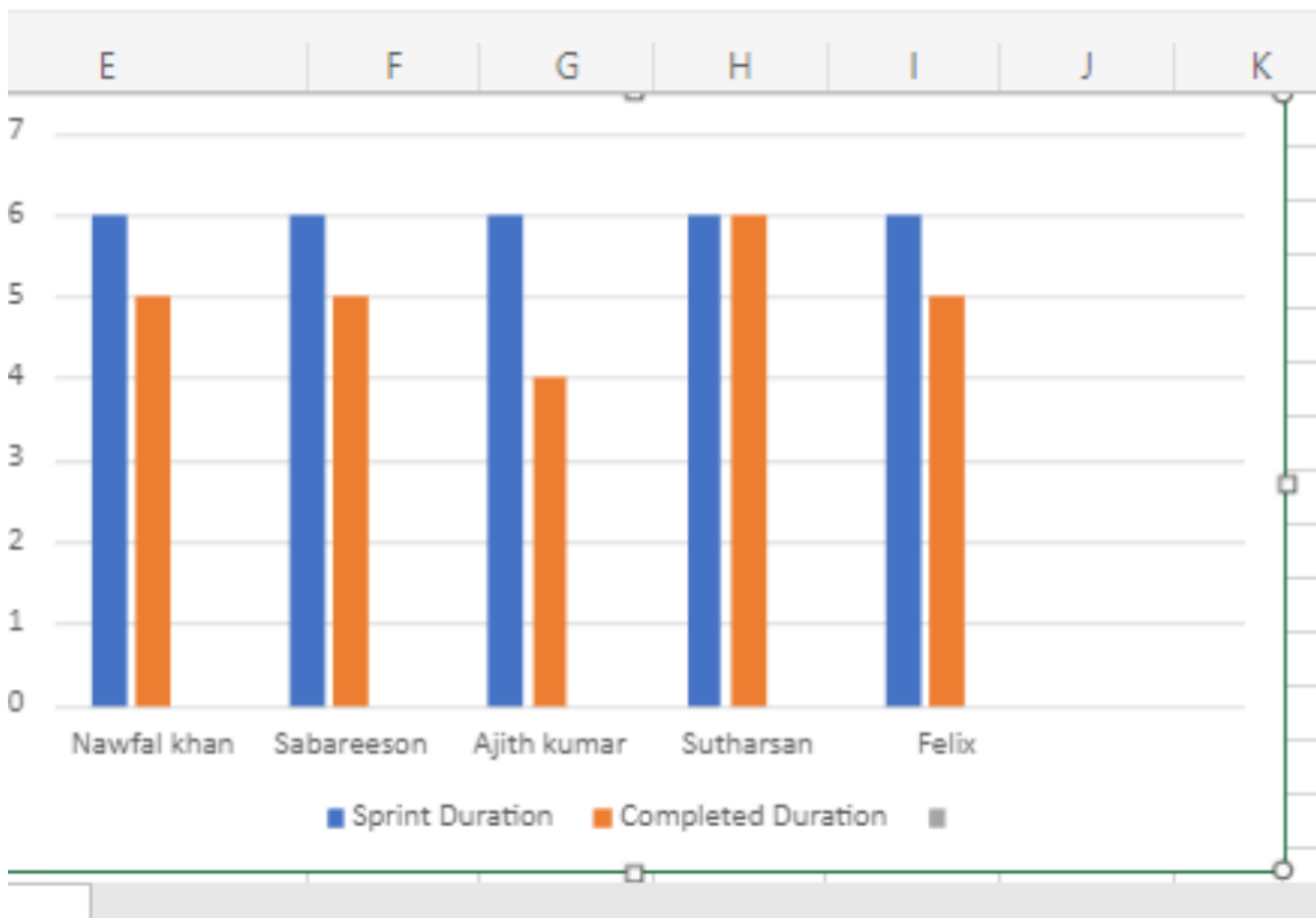
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Nawfal Khan.B
Sprint-1	Registration	USN-2	As a user, I can register for the application through E-mail	1	High	Sabareeson V
Sprint-1	Confirmation	USN-3	As a user, I will confirm the registration once I have received the email from the application	2	Medium	Ajith Kumar R
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	2	High	Felix J & Sutharsan K
Sprint-2	Dashboard	USN-5	As a user, I can view the dashboard of the application by entering my email & password	3	High	Nawfal Khan.B
Sprint-2	ChatBot	USN-6	As a user, I can interact and get help from chatbot	3	Medium	Sabareeson V
Sprint-2	Add items to cart	USN-7	As a user, I can add the products I wish to buy to the carts..	3	Medium	Ajith Kumar R
Sprint-3	Stock Update	USN-8	As a user, I can add products that are not available in the dashboard to the stock list.	5	Medium	Felix J & Sutharsan K
Sprint-4	Request to Customer Care	USN-9	As a user, I can post queries through mail	5	Low	Nawfal Khan.B
Sprint-4	Feedback	USN-10	As a user, I can give my feedback by submitting the form.	5	Low	Sabareeson V

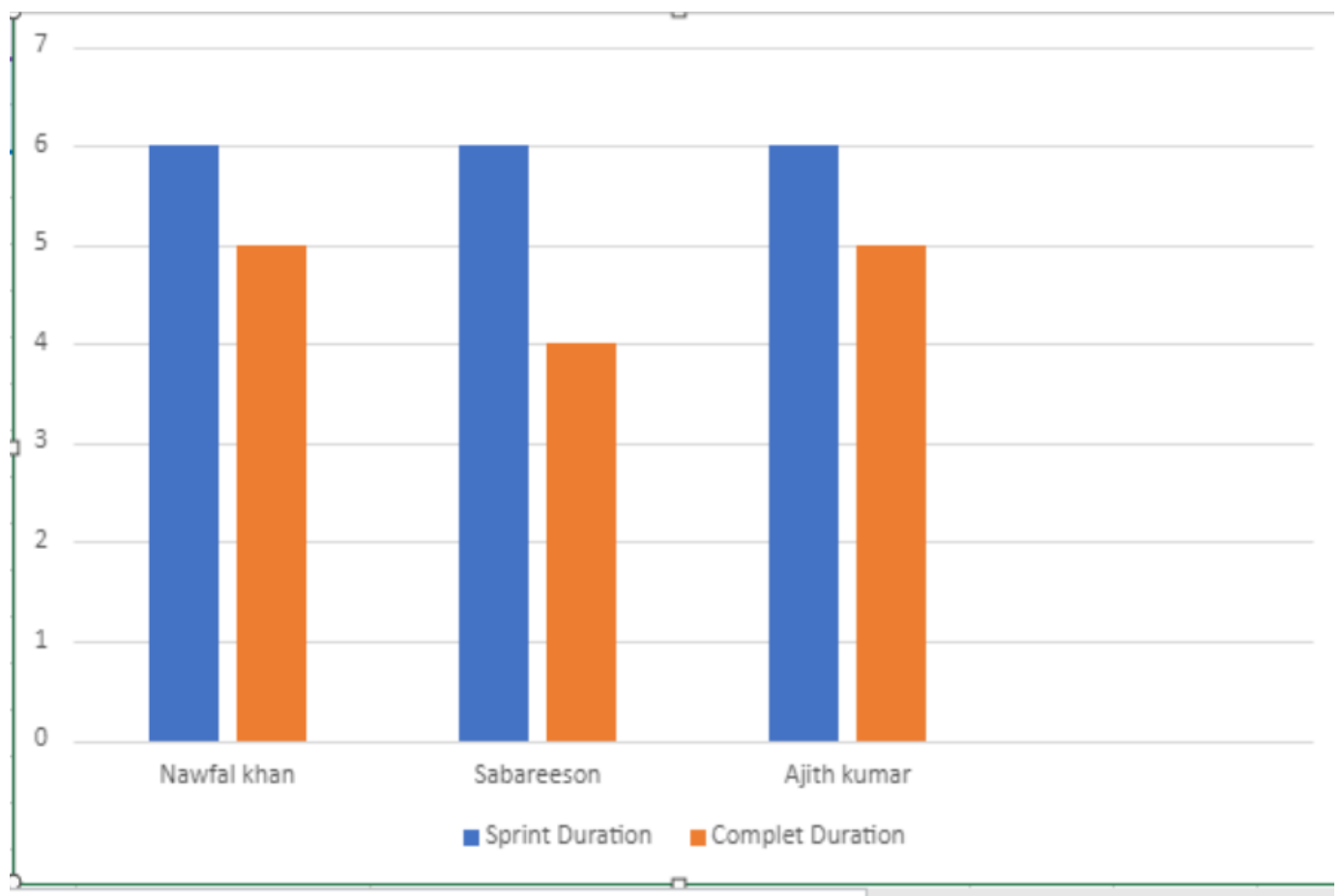
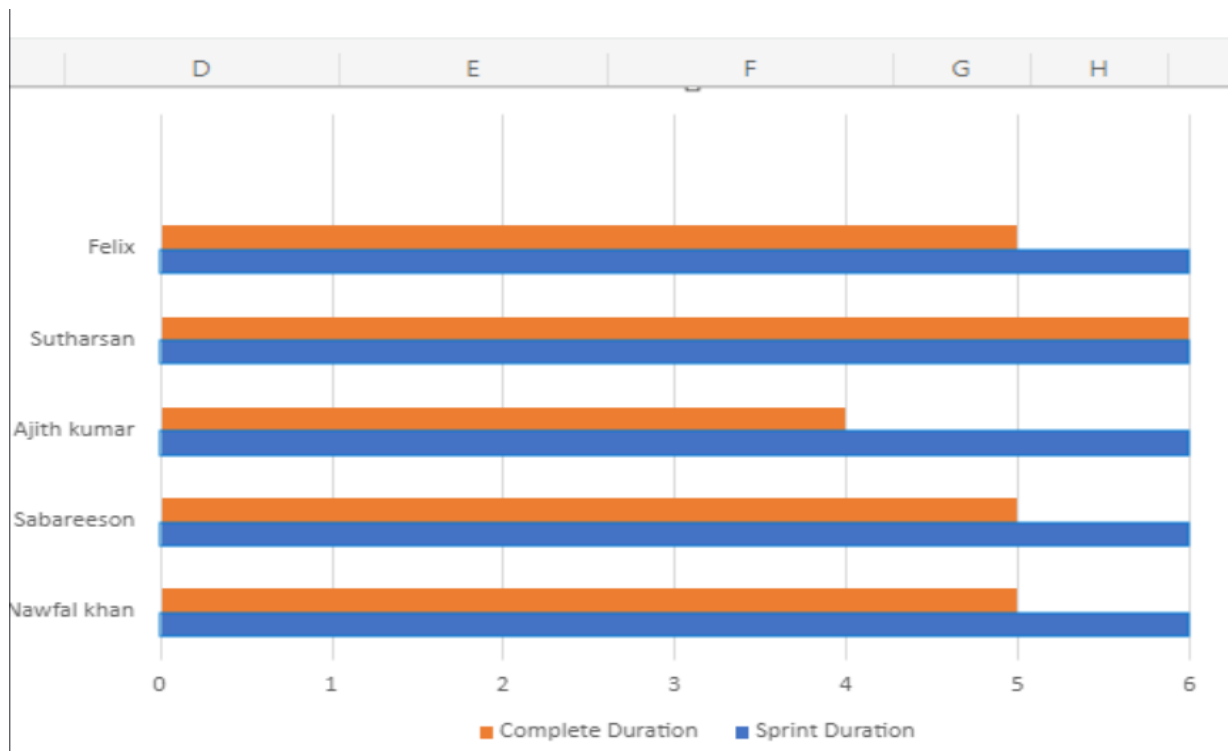
Sprints	Sprint Duration	Velocity	Actual velocity
Sprint-1	6	7	0.85
Sprint-2	6	9	0.66
Sprint-3	6	5	1.2
Sprint-4	6	10	0.6

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	9	6 Days	31 Oct 2022	05 Nov 2022	9	29 Oct 2022
Sprint-3	5	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

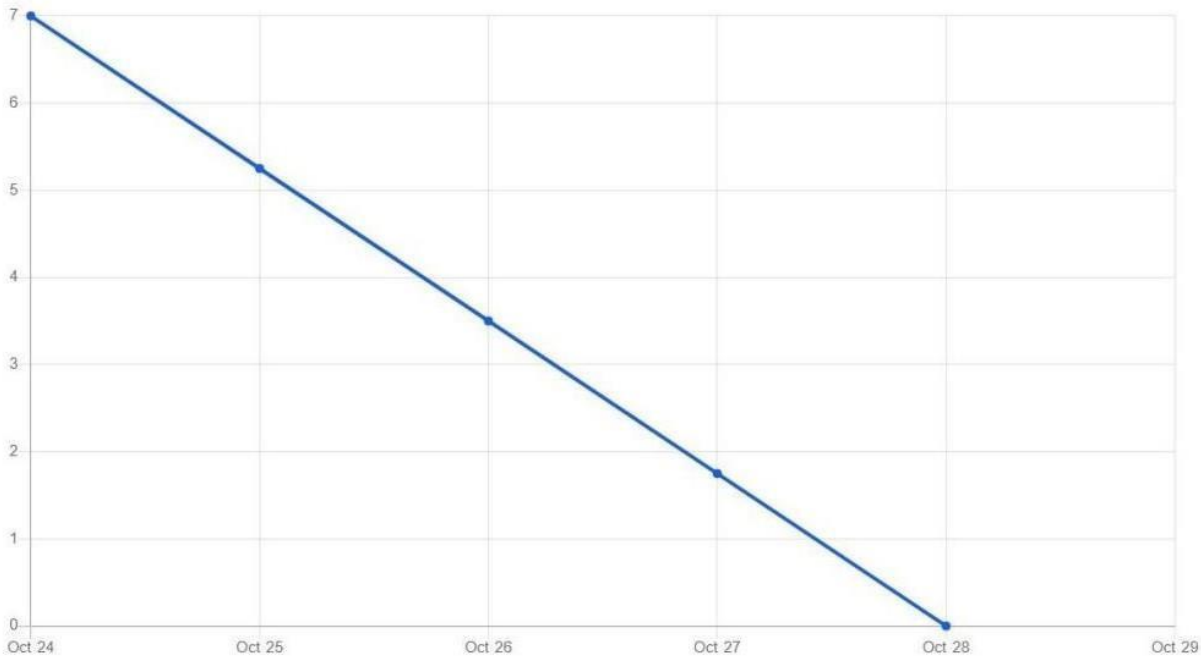
6.3 Reports from JIRA





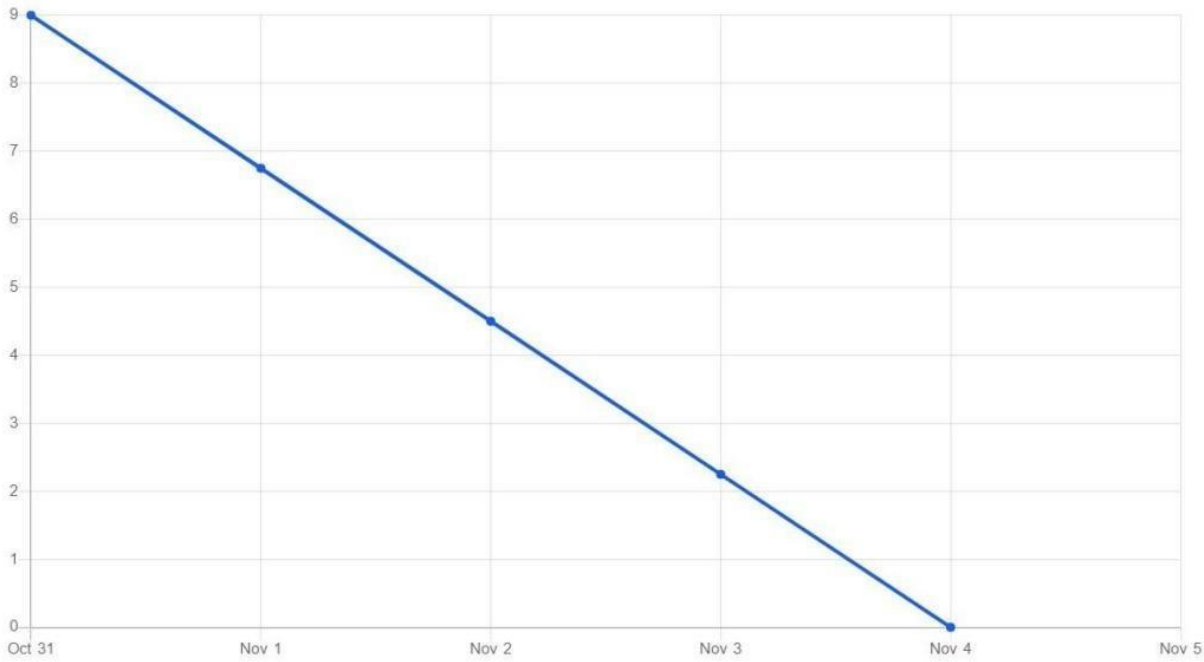
SPRINT-1

Burndown Chart



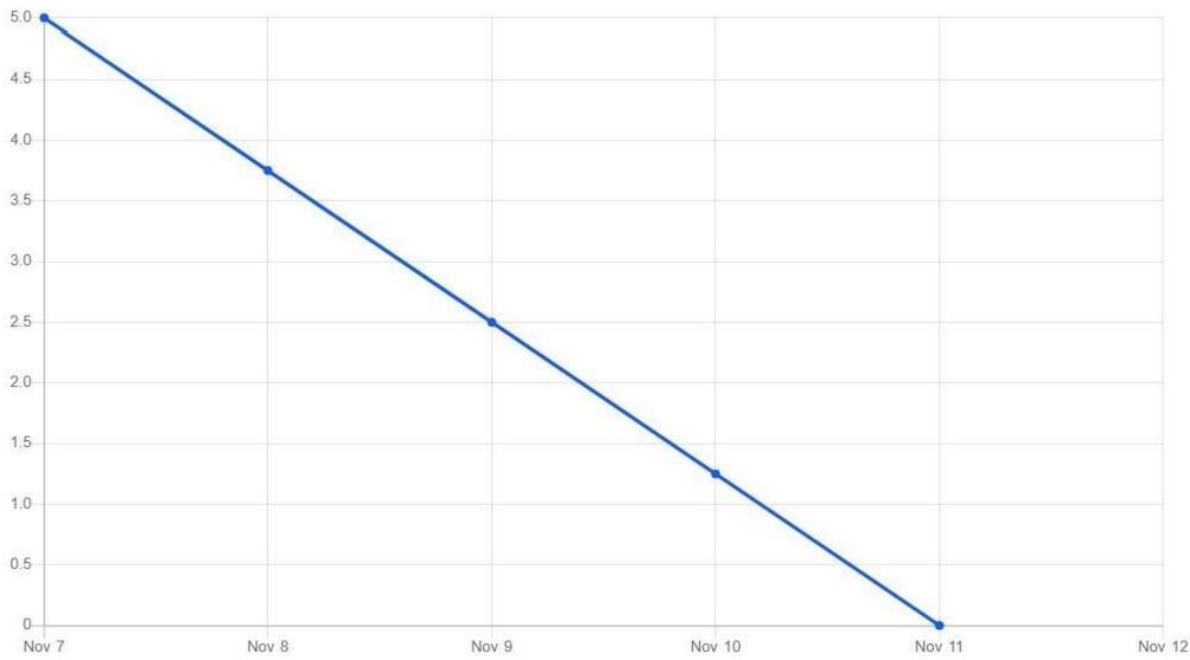
SPRINT-2

Burndown Chart



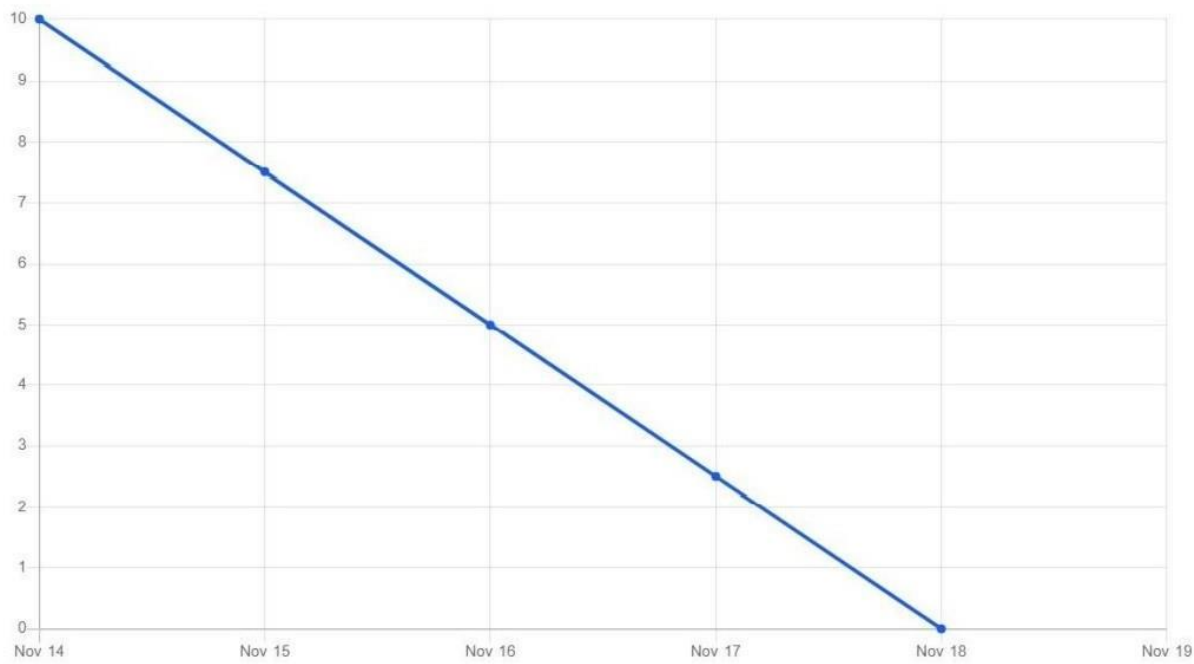
SPRINT-3

Burndown Chart



SPRINT-4

Burndown Chart

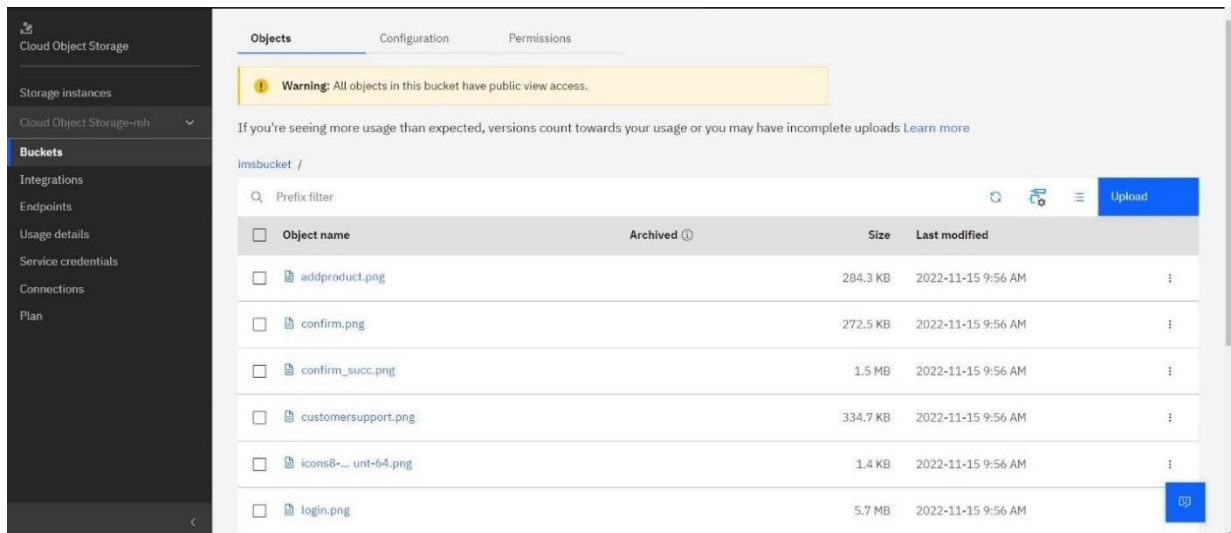


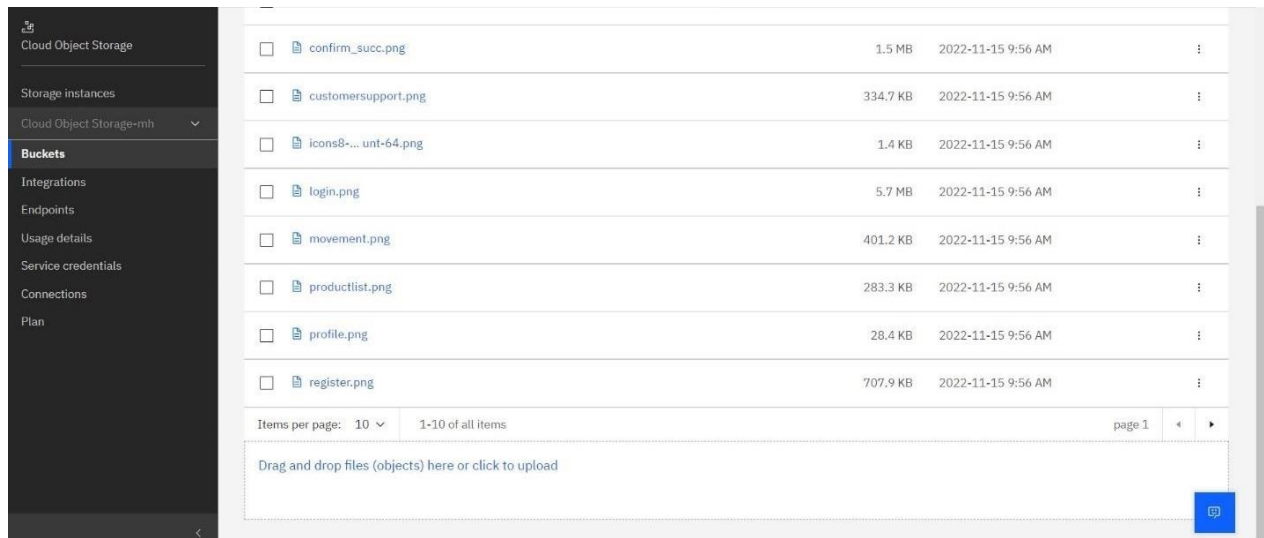
7 CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 IBM Object Storage Service

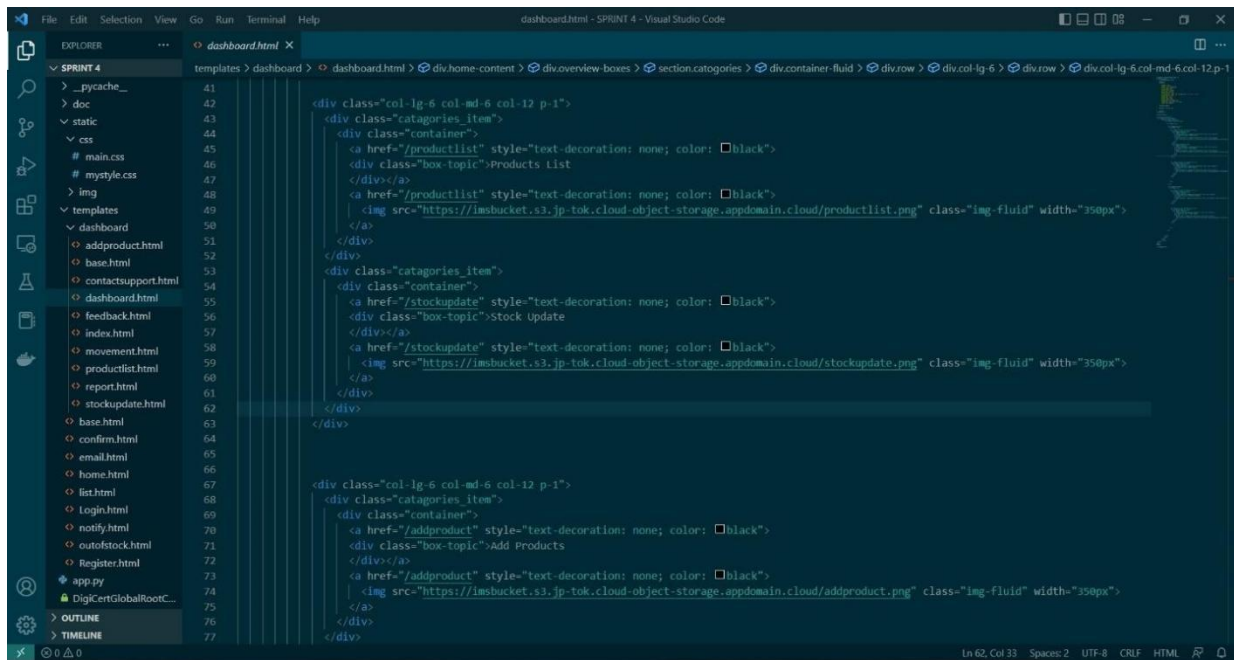
IBM Cloud Object Storage, with its global presence and flexible resiliency options, supports exponential data growth for your cloud-native workloads with best-in-class cost optimization, robust data security, and data governance with ease of use. Built-in data lifecycle operations also make it easy to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

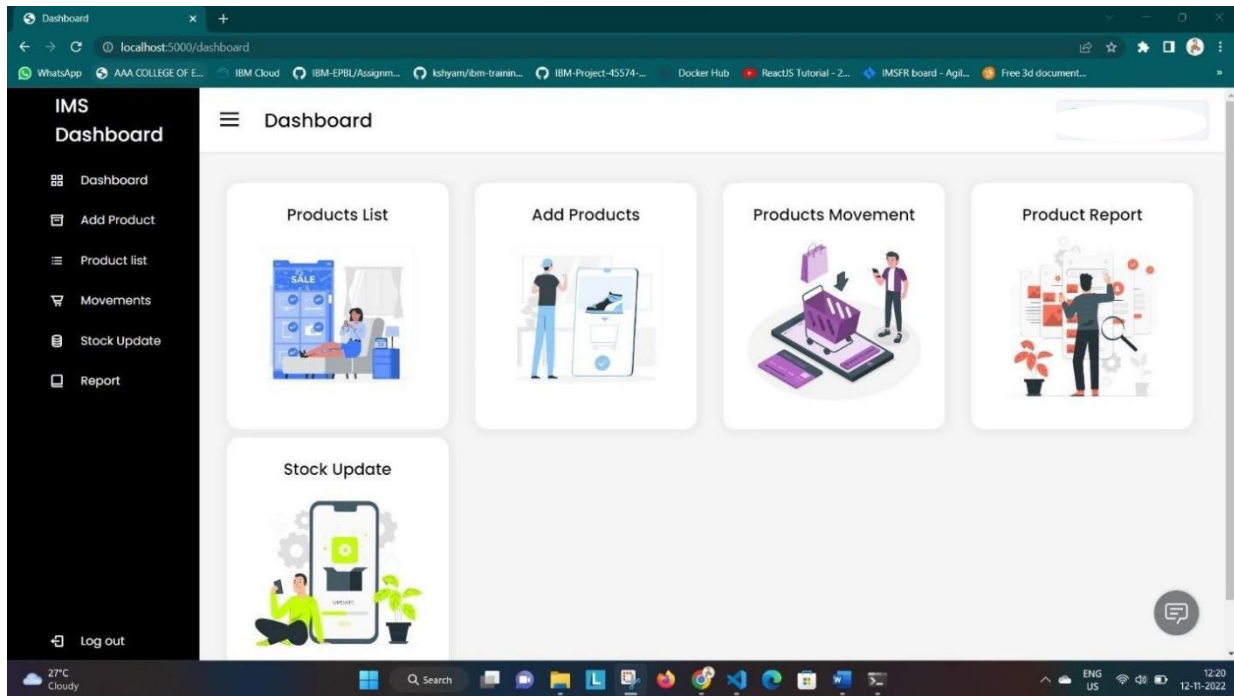
Creating a bucket in IBM Object Storage and uploading the images to the bucket:





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 the leader in trustworthy AI.

Setting Actions to IBM Watson Assistant:

The screenshot shows the IBM Watson Assistant interface. On the left, a sidebar displays the conversation steps. The main area is titled 'Customer starts with:' and contains instructions: '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.' Below this, there is a list of phrases: 'jarvis', 'hi', 'hello', 'hey', and 'I want to know about the IMS'. A 'Total: 5' indicator is shown. A 'Preview' button is located at the bottom right.

This screenshot shows the same IBM Watson Assistant interface as the previous one, but with additional steps visible in the sidebar. The 'Customer starts with:' section remains the same, with the same list of phrases and the 'Preview' button. The sidebar on the left shows a sequence of steps, including 'Contact us', 'Chat with an agent', and 'Defined'. A 'New step' button is visible at the bottom of the sidebar.

Project Documen...Project Report D...Project Report D...20 Inventory Ma...IBMIBMIBM Watson Servi...IBM Watson Assi...Virtual Agent - IB...+
jp-tok.assistant.watson.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Aconversation%3Ajp-tok%3Aa%2Fa23b914be20346fa83a3da7951decc98%3Ab7ec50cd-af28-4bb0-aa53-52dc00c3...
WhatsAppAAA COLLEGE OF E...IBM CloudIBM-EPBL/Assignm...kshyam/ibm-trainin...IBM-Project-45574...Docker HubReactJS Tutorial - 2...IMSFR board - Agil...Free 3d document...

IBM Watson Assistant LiteUpgradeIMS bot▼Learning center

I want to know about the IMS

connected to our sales agent now
Action complete

2 is Login / Sign up
Is there anything else that I can assist you with?
8 No, Thank you Login +2
Continue to next step

8 is Back to Main Menu
9 Anything else I can help you with?
Re-ask previous step(s)

8 is No, Thank you
10 Bye
Action complete

8 is Login
Please use this link1
New 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
Preview▶

Project Documen...Project Report D...Project Report D...20 Inventory Ma...IBMIBMIBM Watson Servi...IBM Watson Assi...Virtual Agent - IB...+
jp-tok.assistant.watson.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Aconversation%3Ajp-tok%3Aa%2Fa23b914be20346fa83a3da7951decc98%3Ab7ec50cd-af28-4bb0-aa53-52dc00c3...
WhatsAppAAA COLLEGE OF E...IBM CloudIBM-EPBL/Assignm...kshyam/ibm-trainin...IBM-Project-45574...Docker HubReactJS Tutorial - 2...IMSFR board - Agil...Free 3d document...

IBM Watson Assistant LiteUpgradeIMS bot▼Learning center

I want to know about the IMS

Continue to next step

8 is Sign up
Please use this [link]
(http://localhost:5000/register) to Sign Up
Continue to next step

This step has no content
13 Back to Main ... End
Continue to next step

13 is End
14 Thank you for your time. We will reach you shortly
Action complete

13 is Back to Main Menu
15 Anything else I can help you with?
Continue to next step
New step +

Customer starts with:

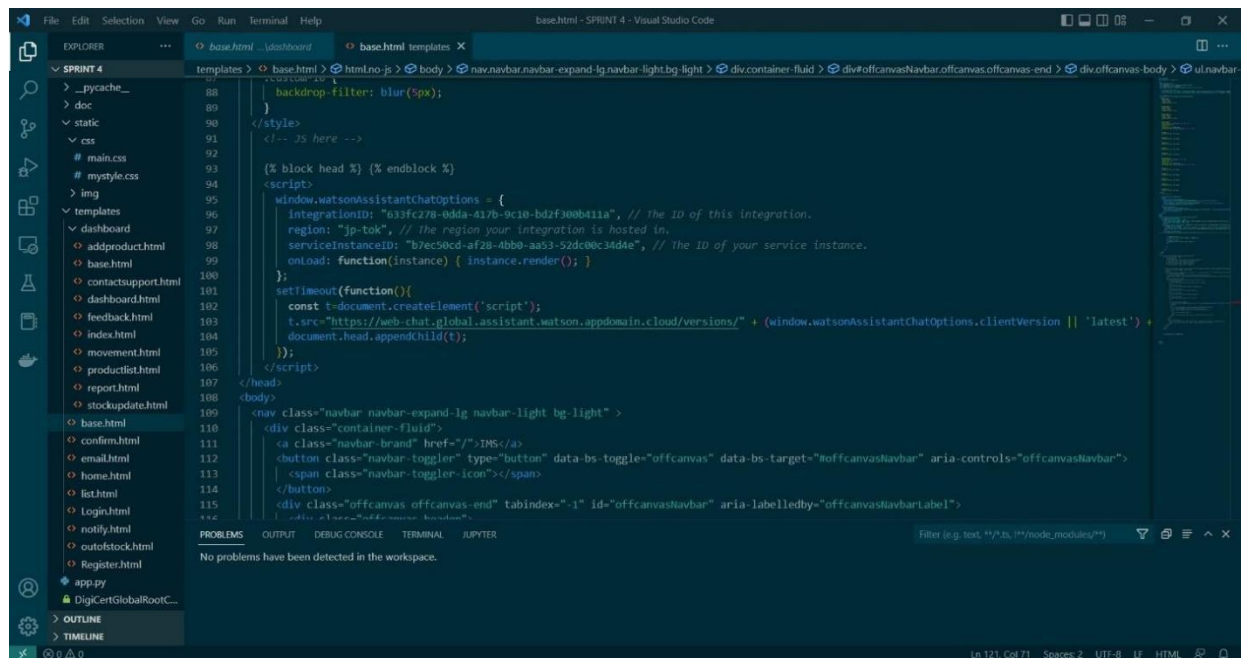
Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.
The more phrases you enter, the better your assistant can recognize what the customer wants.

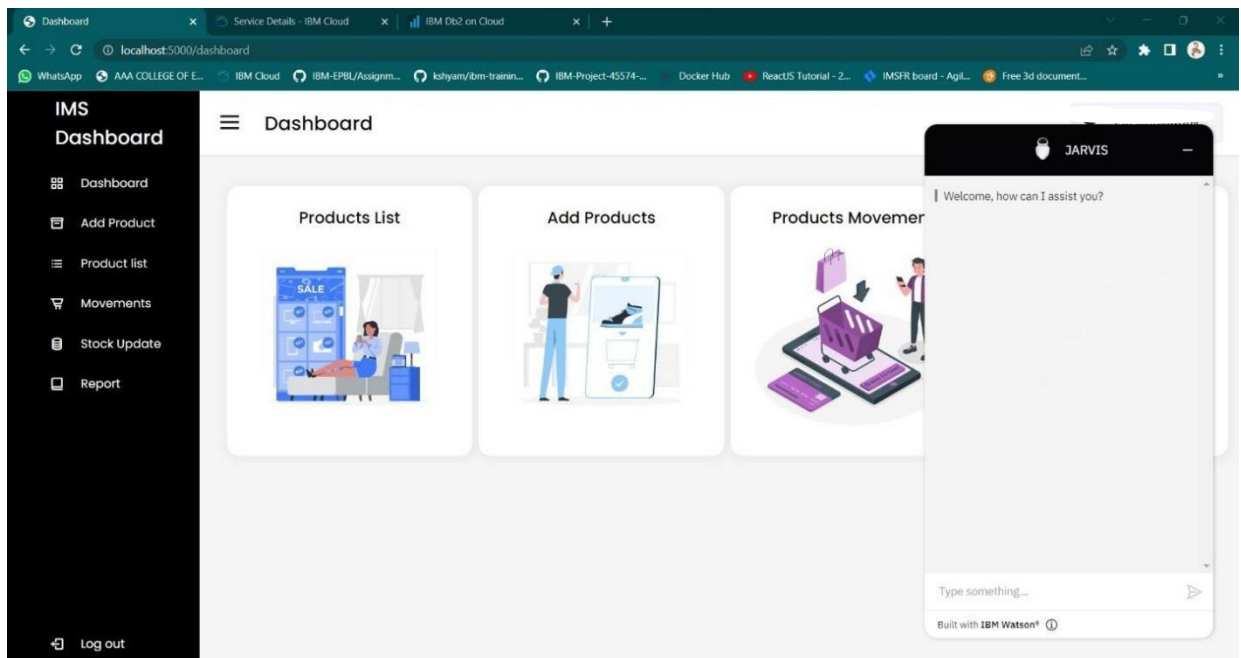
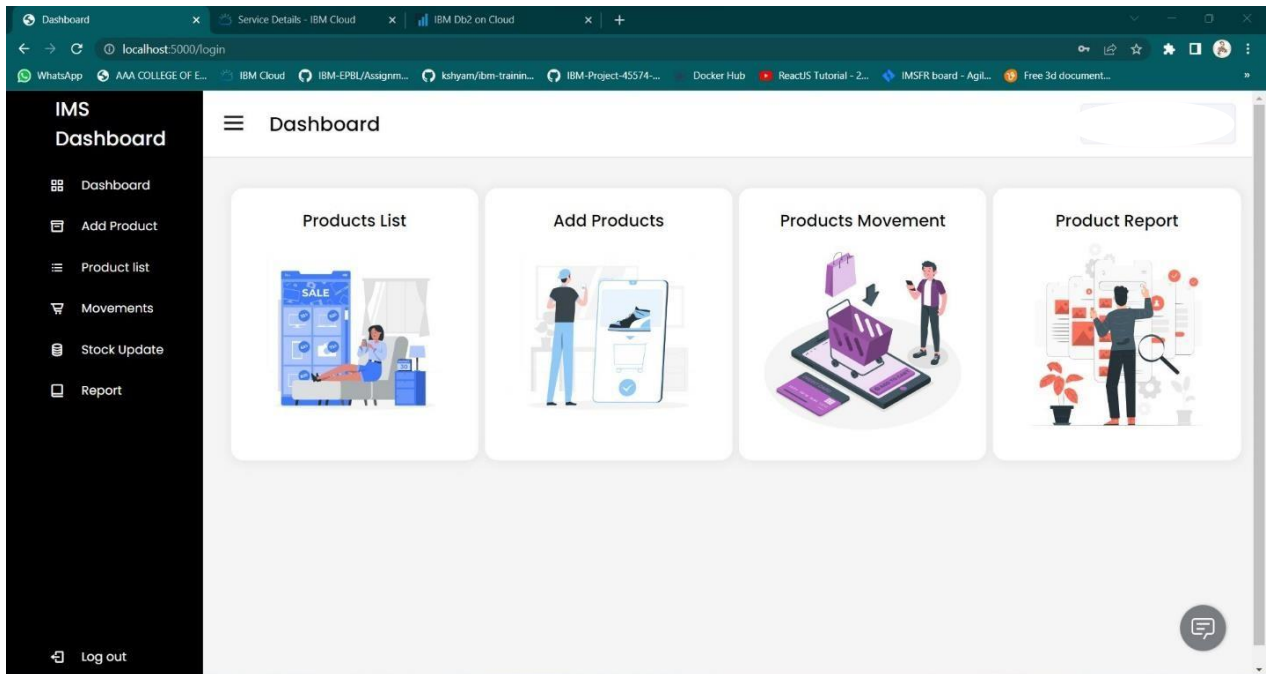
Enter phrases your customer might use to start this action Total: 5
Enter a phrase
jarvis
hi
hello
hey
I want to know about the IMS
Preview▶

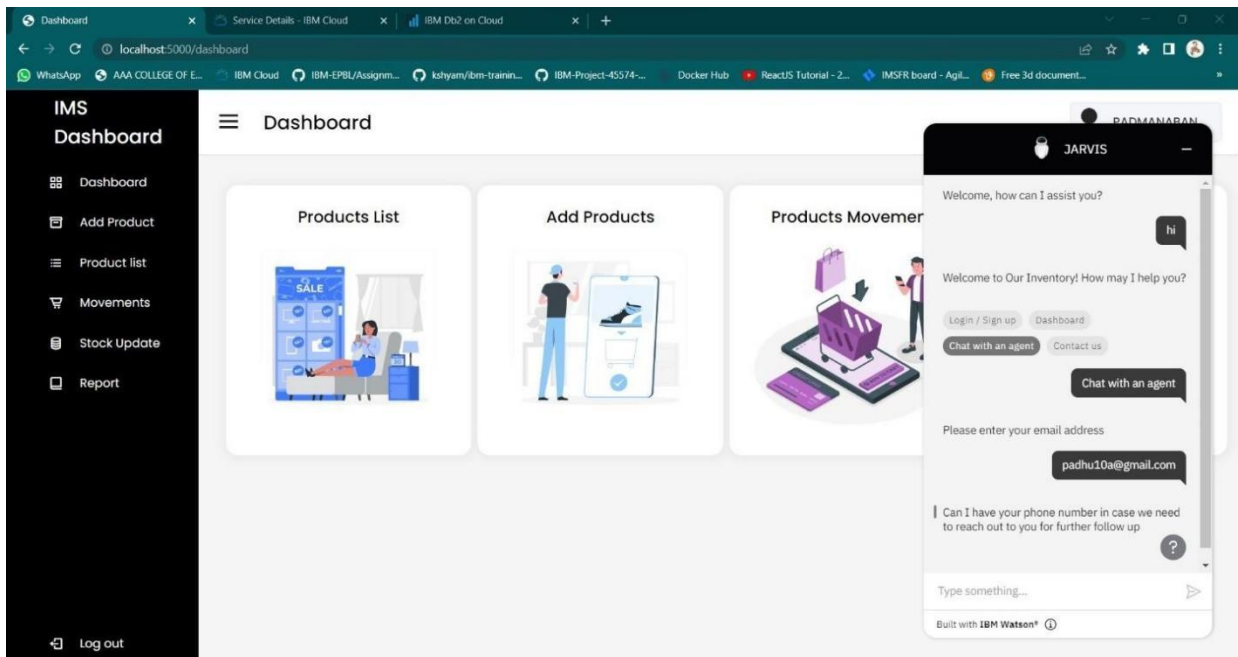
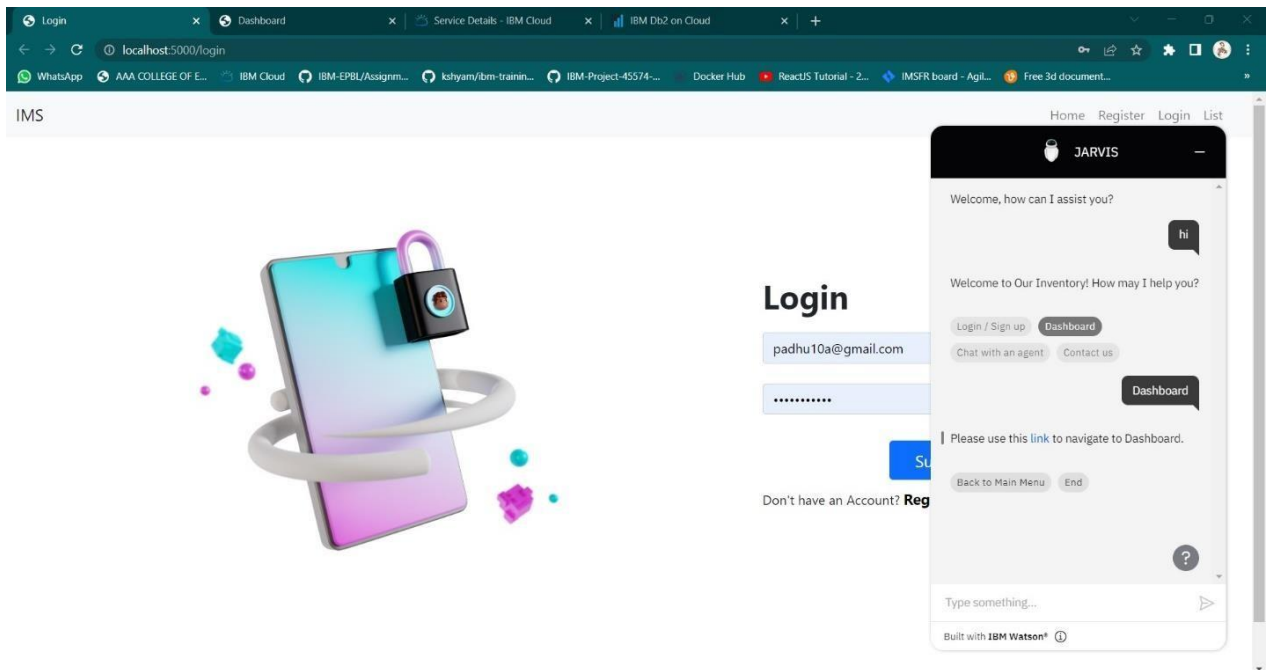
Code:

```
<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 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(t);
  });
</script>
```

Integrating the IBM Watson Assistant in the base template:







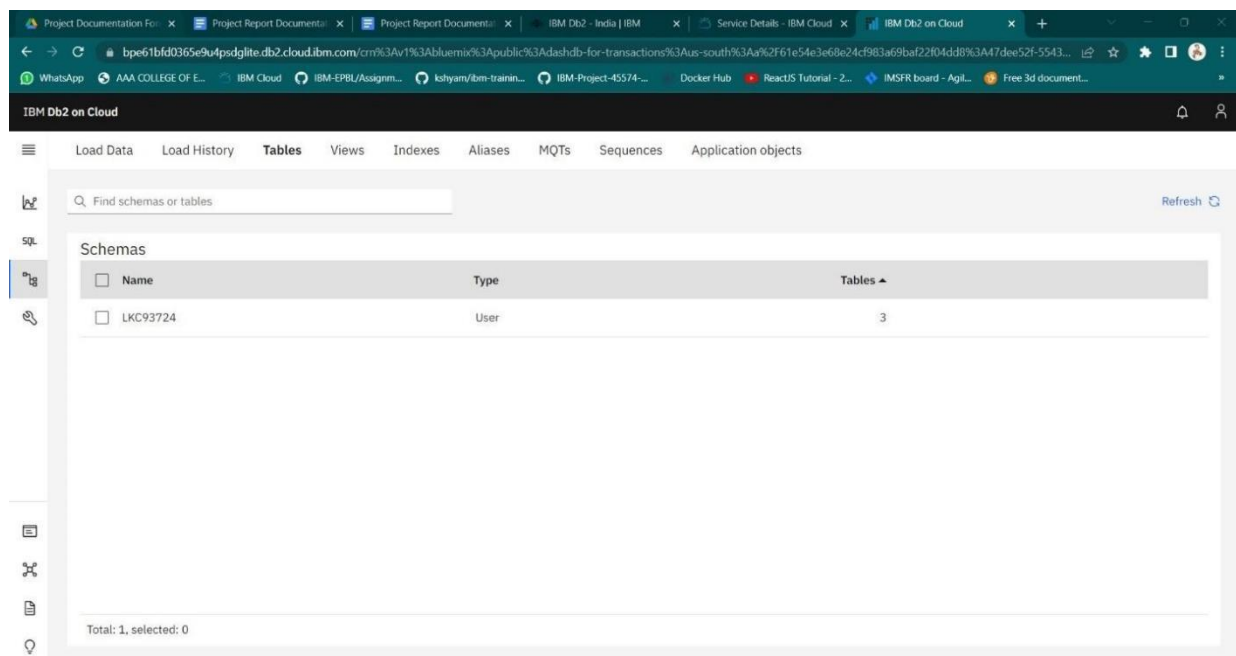
7.3 Database Schema(IBM DB2)

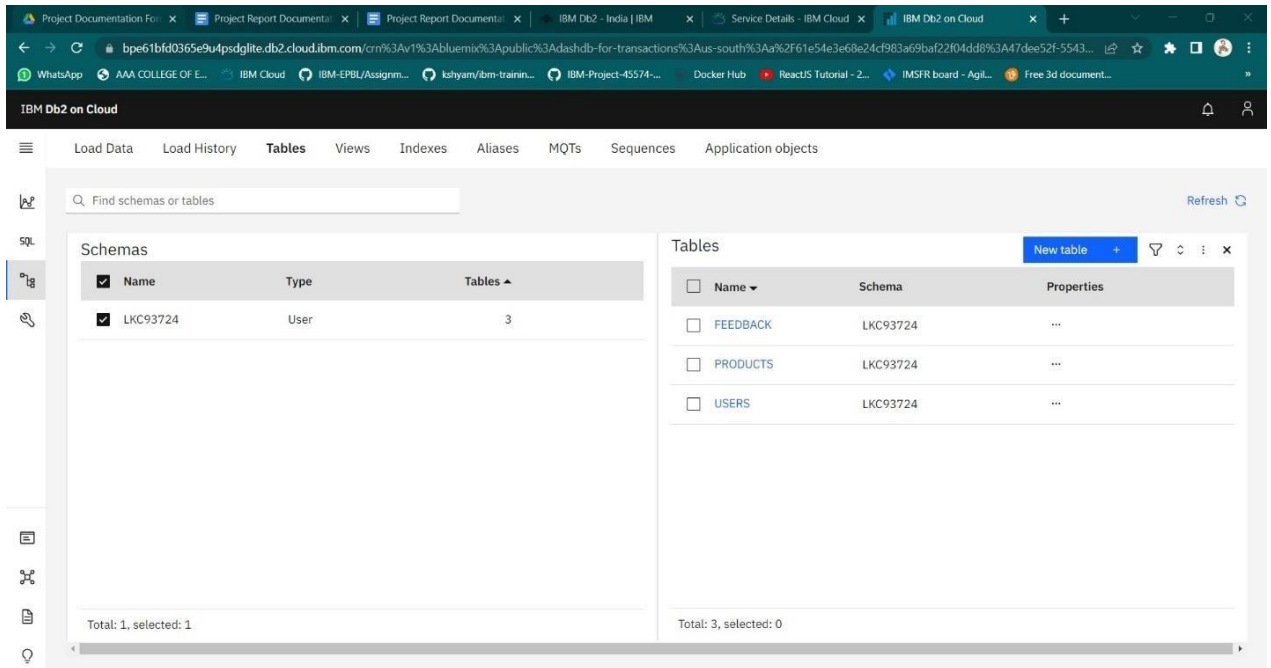
IBM Db2 Database on IBM Cloud combines a proven, AI-infused, enterprise-ready data management system with an integrated data and AI platform built on the security-rich, scalable Red Hat® OpenShift® foundation. Derive insights with machine learning embedded into query processing. Cut costs with the multimodal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

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

```
import ibm_db
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lkc93724;PWD=zAzNGa6DaNk6xvle", "", "")
```





8 TESTING

8.1 Test Cases

					Date	16-Nov-22						
					Team ID	PNT2022TMD51170						
					Project Name	Project - Inventory Management System For Retail						
					Maximum Marks	4 marks						
Test case ID	Feature Type	Component	Test Scenario	Pre-Req	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID
LoginPage_TC_001	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/Signup popup displayed or not	https://shopencr.com/	Login/Signup popup should display	Working as expected	Pass			
LoginPage_TC_002	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/Signup 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://shopencr.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_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1. Enter URL(https://shopencr.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: padhu10a@gmail.com password: padhu123	User should navigate to user account homepage	Working as expected	Pass			
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials		1. Enter URL(https://shopencr.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: padhu10a@gmail.com password: padhu123	Application should show 'Incorrect email or password' validation message.	Working as expected	Pass			
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials		1. Enter URL(https://shopencr.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: padhu10a@gmail.com password: padhu123	Application should show 'Incorrect email or password' validation message.	Working as expected	Pass			
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with Invalid credentials		1. Enter URL(https://shopencr.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: padhu10a@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 number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	19
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	25	0	0	20
Security	2	0	0	2

Outsource Shipping	3	0	0	3
Exception Reporting	7	0	0	7
Final Report Output	4	0	0	4
Version Control	2	0	0	2

RESULTS

9 ADVANTAGES & DISADVANTAGES

9.1 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.

9.2 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.

10 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.

11 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.

12 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=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lkc93724;PWD=zAzNGa6DaNk6xvle",'')

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']
mobilenno = request.form['mobilenno']
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(mobilenno)
    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%;">View Email in Browser</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">

</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">
```

```

```

```

            </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>

```

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

<!--[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%;">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!</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]-->`

``

CONFIRM

</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;">Once again, Thanks for signing up with us!</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 = 'padhu10a@gmail.com'
epassword = 'rbjibzksssszsbrjo'
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 confirmation():
    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(prepare_stmt, 1, var_list[0])
    ibm_db.bind_param(prepare_stmt, 2, var_list[1])
    ibm_db.bind_param(prepare_stmt, 3, var_list[2])
    ibm_db.bind_param(prepare_stmt, 4, var_list[3])
    ibm_db.bind_param(prepare_stmt, 5, var_list[4])
    ibm_db.bind_param(prepare_stmt, 6, var_list[5])
    ibm_db.bind_param(prepare_stmt, 7, var_list[6])
    ibm_db.bind_param(prepare_stmt, 8, var_list[7])
    ibm_db.bind_param(prepare_stmt, 9, var_list[8])
    ibm_db.execute(prepare_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(prepare_stmt, 1, session['name'])
        ibm_db.execute(prepare_stmt)
        dictionary = ibm_db.fetch_both(prepare_stmt)
        while dictionary != False:
            # print ("The Name is : ", dictionary)
            products.append(dictionary)
            dictionary = ibm_db.fetch_both(prepare_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(prepare_stmt, 1,quantityout)
        ibm_db.bind_param(prepare_stmt, 2, tow)
        ibm_db.bind_param(prepare_stmt, 3, pname)
        ibm_db.bind_param(prepare_stmt, 4, session['name'])
        ibm_db.execute(prepare_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">

</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)]><!-->
```

```
<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]-->
```

```





```

```





```

```
<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%;">Please order new stocks to get rid of the out-of-stock.

</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;">Post queries in the Contact Support for further clearance!</p>

<p style="line-height: 160%; font-size: 14px; text-align: center;"> </p>

<p style="line-height: 160%; font-size: 14px; text-align: center;">Thank you!</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 = 'padhu10a@gmail.com'
```

```
epassword = 'rbjibzkssszsbrjo'
```

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

```
        prepare_stmt = ibm_db.prepare(conn, sql)
```

```
        ibm_db.bind_param(prepare_stmt, 1, session['name'])
```

```
        ibm_db.execute(prepare_stmt)
```



```

dictionary = ibm_db.fetch_both(prepare_stmt)
while dictionary != False:
    # print ("The Name is : ", dictionary)
    products.append(dictionary)
    dictionary = ibm_db.fetch_both(prepare_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 = ?"
        prepare_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prepare_stmt, 1, session['name'])
        ibm_db.execute(prepare_stmt)
        dictionary = ibm_db.fetch_both(prepare_stmt)
        while dictionary != False:
            # print ("The Name is : ", dictionary)
            products.append(dictionary)
            dictionary = ibm_db.fetch_both(prepare_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(prepare_stmt, 1, id)
```

```
    ibm_db.bind_param(prepare_stmt, 2, session['name'])
```

```
    ibm_db.execute(prepare_stmt)
```

```
    products = []
```

```
    sql = "SELECT * FROM Products WHERE HOLDERNAME = ?"
```

```
    prep_stmt = ibm_db.prepare(conn, sql)
```

```
    ibm_db.bind_param(prepare_stmt, 1, session['name'])
```

```
    ibm_db.execute(prepare_stmt)
```

```
    dictionary = ibm_db.fetch_both(prepare_stmt)
```

```
    while dictionary != False:
```

```
        # print ("The Name is : ", dictionary)
```

```
        products.append(dictionary)
```

```
        dictionary = ibm_db.fetch_both(prepare_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(prepare_stmt, 1, pname)
ibm_db.bind_param(prepare_stmt, 2, quantityin)
ibm_db.bind_param(prepare_stmt, 3, pid)
ibm_db.bind_param(prepare_stmt, 4, session['name'])
ibm_db.execute(prepare_stmt)

```

```

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

```
        ibm_db.bind_param(prepare_stmt, 2, quantity)
```

```
        ibm_db.bind_param(prepare_stmt, 3, '0')
```

```
        ibm_db.bind_param(prepare_stmt, 4, "")
```

```
        ibm_db.bind_param(prepare_stmt, 5, str(the_time))
```

```
        ibm_db.bind_param(prepare_stmt, 6, session['name'])
```

```
        ibm_db.execute(prepare_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(prepare_stmt, 1, pname)
```

```
        ibm_db.bind_param(prepare_stmt, 2, quantity)
```

```
        ibm_db.bind_param(prepare_stmt, 3, '0')
```

```
        ibm_db.bind_param(prepare_stmt, 4, "")
```

```
        ibm_db.bind_param(prepare_stmt, 5, str(the_time))
```

```
        ibm_db.bind_param(prepare_stmt, 6, session['name'])
```

```
        ibm_db.execute(prepare_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(prepare_stmt, 1, session['name'])
        ibm_db.execute(prepare_stmt)
        dictionary = ibm_db.fetch_both(prepare_stmt)
        while dictionary != False:
            # print ("The Name is : ", dictionary)
            products.append(dictionary)
            dictionary = ibm_db.fetch_both(prepare_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']

```

```

mobilenno = request.form['mobilenno']
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>"+mobilenno+"<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 = 'padhu10a@gmail.com'
epassword = 'rbjibzkssszsbrjo'
email_to = 'imsa3258@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(prepare_stmt, 1, interface)
```

```
        ibm_db.bind_param(prepare_stmt, 2, availability)
```

```
    ibm_db.bind_param(prepare_stmt, 3, userfriendly)
    ibm_db.bind_param(prepare_stmt, 4, chatbot)
    ibm_db.bind_param(prepare_stmt, 5, suggest)
    ibm_db.bind_param(prepare_stmt, 6, session['name'])
    ibm_db.execute(prepare_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"]
```


requirements.txt

flask
ibm_db
pandas

flask_service.yaml:

apiVersion: v1
kind: Service
metadata:
 name: flask-app-service
spec:
 type: NodePort
 ports:
 - port: 5000
 selector:
 app: ims-final

flask_ingress.yaml:

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
 name: flask-app-ingress
 annotations:
 kubernetes.io/ingress.class: nginx
 nginx.ingress.kubernetes.io/ssl-redirect: "false"
spec:
 # ingressClassName: nginx
 rules:
 - http:
 paths:

- backend:
 - service:
 - name: flask-app-service
 - port:
 - number: 5000
 - path: /
 - pathType: Prefix

ibm_deployment.yaml:

apiVersion: apps/v1

kind: Deployment

metadata:

- name: ims-final

spec:

- replicas: 3

- selector:

- matchLabels:

- app: ims-final

- template:

- metadata:

- labels:

- app: ims-final

- spec:

- containers:

- name: job-portal-container

- image: jp.icr.io/padmanaban/ims-final

- imagePullPolicy: Always

- ports:

- containerPort: 5000

- protocol: TCP

