



INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

IBM PROJECT

Submitted By

G. ARUN (815119104009)

S. GHOUSONISAN (815119104015)

R. ILAVARASAN (815119104017)

M.S. LOGESH (815119104025)

in partial fulfilment for the award of the degree

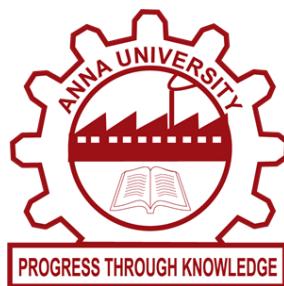
of

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 **G. ARUN (815119104009)**, **S. GHOUSONISAN (815119104015)**, **R. ILAVARASAN (815119104017)** and **M.S. LOGESH (815119104025)**, 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.

NALAIYA THIRAN - Project Based Learning on Professional Readiness for Innovation, Employment and Entrepreneurship.



Team ID PNT2022TMID46132

Project Name	<i>Inventory Management System for Retailers</i>
Project Members	<i>M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R</i>

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	V
1	INTRODUCTION	1
	1.1 Project Overview	1
	1.2 Purpose	1
2	LITERATURE SURVEY	3
	2.1 Existing problem	4
	2.2 References	5
	2.3 Problem Statement Definition	5
3	IDEATION & PROPOSED SOLUTION	7
	3.1 Empathy Map Canvas	7
	3.2 Ideation & Brainstorming	8
	3.3 Proposed Solution	10
	3.4 Problem Solution fit	13
4	REQUIREMENT ANALYSIS	14
	4.1 Functional requirements	14
	4.2 Non-Functional requirements	14
5	PROJECT DESIGN	16
	5.1 Data Flow Diagrams	16
	5.2 Solution & Technical Architecture	17
	5.3 User Stories	19
6	PROJECT PLANNING & SCHEDULING	20
	6.1 Sprint Planning & Estimation	20
	6.2 Sprint Delivery Schedule	21
	6.3 Reports from JIRA	21
7	CODING & SOLUTIONING (Explain the features added in the project along with code)	23
	7.1 Feature 1	23
	7.2 Feature 2	26
	7.3 Database Schema (if Applicable)	27
8	TESTING	28
	8.1 Test Cases	28
	8.2 User Acceptance Testing	28
9	RESULTS	30
	9.1 Performance Metrics	30
10	ADVANTAGES & DISADVANTAGES	31
11	CONCLUSION	32
12	FUTURE SCOPE	33
13	APPENDIX	34
	Source Code	34
	GitHub & Project Demo Link	92

ABSTRACT

Inventory management is one of the pillars of a successful retail operation. Inventory management techniques help stores and ecommerce sellers satisfy customers, reduce costs and increase profits. 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 inventory management results in lower costs and a better understanding of sales patterns. Inventory management tools and methods give retailers more information with which to run their businesses. Inventory management is vital for retailers because the practice helps them increase profits. They are more likely to have enough inventory to capture every possible sale while avoiding overstock and minimizing expenses. From a strategic point of view, inventory management increases efficiency.

CHAPTER 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 primary purpose of inventory management is to ensure there is enough goods or materials to meet demand without creating overstock, or excess inventory. Accurate inventory management is key to running a successful product

business. Tracking stock regularly can help avoid stock errors and other problems. And 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.

And 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.

CHAPTER 2

LITERATUTE SURVEY

S.NO	TITLE	TECHNIQUES	PROS	CONS
1.	ZOHO INVENTORY	Enable you to track items as they move throughout the supply chain.	User Friendly, Easy to Use, Can Track the Order, Integrated & Stock Transfer	Customer Support, Reporting, Pointless Features
2.	CHESTER SOFTWARE INVENTORY	Bundling Item as Promotional Tool	Simple & Clean, Regular Updates	More Bugs & More Setup Process
3.	SELF MENTOR INVENTORY	Inventory Reporting & Easy Access	User Friendly & Easy Access	Cost for Premium Version
4.	TESTSERIES SOFTWARE INVENTORY	Follows FIFO Method (First In First Out)	Easy to Use & Value for Money	Late Responsive
5.	ORDERHIVE INVENTORY	Conduct a Complete Inventory Management Self-Audit	Cloud based, Excellent Customer Service & Quick responsive	Pay for Features and Payment Issues

2.1 Existing Problem

- Lack of real-time reporting
- Lack of Inventory Visibility
- Inefficient Inventory Management Process or Software
- Identifying Incorrectly Located Materials
- Keeping up with Overstocks or Understocks
- Managing Inventory Waste & Defects
- Lack of Centralized Inventory Hub
- Changing Demand and Increasing Competition
- Supply Chain Complexity and Evolving Packaging
- Managing Warehouse Space and Efficiency
- Insufficient Order Management and Poor Production Planning
- Lack of Expertise and Poor communication
- Inconsistent Tracking
- Inaccurate Data and Inventory Loss
- Manual Documentation and Inadequate Software
- Stock Problem and Inefficient Processes
- Insufficient Order Management
- Expanding Product Portfolios

2.2 References

1. <https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-management-challenges.shtml>
2. <https://www.tranquilbs.com/inventory-management-problems/>
3. <https://www.wisys.com/what-are-the-common-problems-in-inventory-management/>
4. <https://www.hakunamatata.tech/our-resources/blog/inventory-management-most-common-challenges-and-solution-to-overcome-in-2022/>
5. <https://www.kapturecrm.com/blog/20-top-challenges-solutions-of-inventory-management/>
6. <https://tallysolutions.com/business-guides/common-inventory-management-problems-challenges-solutions/>
7. <https://magenest.com/en/inventory-management-challenges/>

2.3 Problem Statement Definition

Demand is frequently unpredictable in inventory systems, and lead times can often vary. Managers frequently keep a safety supply to minimize shortages. In such cases, it's difficult to say what order amounts and reorder points will result in the lowest total inventory cost. The inventory issue refers to the general issue of deciding how much inventory to keep on hand in expectation of possible demand. Loss occurs when a business is unable to meet demand (for example, when a store loses sales or when soldiers in a war run out of ammunition) or when commodities are stocked for which there is no demand.



miro



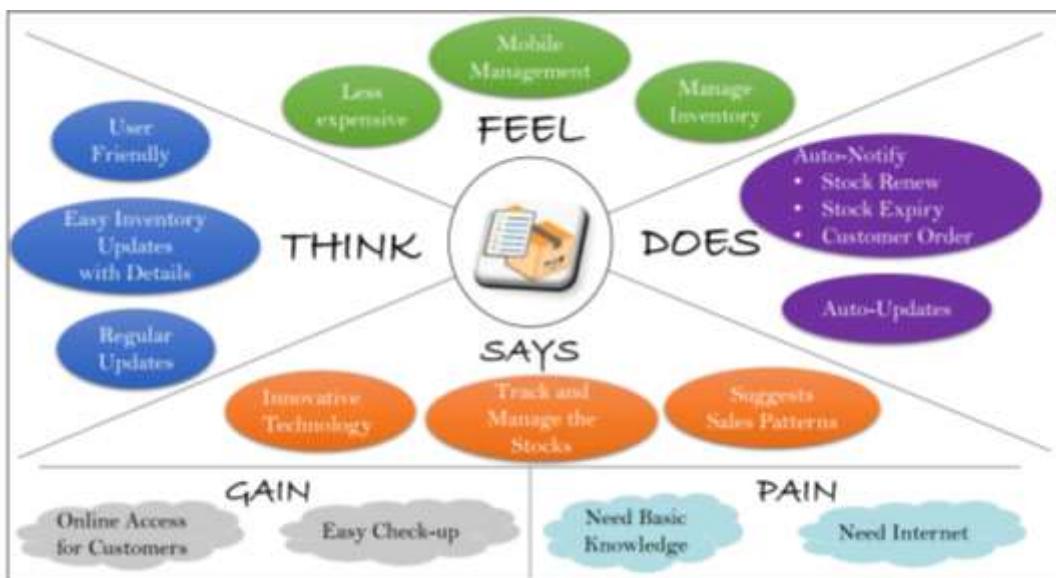
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS - 1	Pharmacist	Save time and reduce operational effort	Software can be too expensive for small and mid-sized companies	Lack of control and organization & difficulty maintaining the accuracy of information	Frustrated & Stressed
PS - 2	Automotive Industry	Minimize loss occurring from damages and wasted materials	Poor real-time updates on reporting & Data-entry errors	Lack of streamlined communication between departments	Frustrated & Tired

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



3.2 Ideation & Brainstorming

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that **ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.**

BRAINSTORM & IDEA PRIORITIZATION

Using this
brainstorm
everyone can
get to know
about our
project plan and
the idea's

PROBLEM STATEMENT

PROBLEM

Demand is frequently unpredictable in inventory systems, and lead times can often vary. Managers frequently keep a safety supply to minimize shortages. In such cases, it's difficult to say what order amounts and reorder points will result in the lowest total inventory cost.

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.
	
	

BRAIN STORM

Logesh

Give Sales Pattern for inventory	Inventory Alert	Minimizes Out-of-Stocks
Batch Tracking	Improves Customers Satisfaction	Improves Multi-Channel and Omnichannel Performance and Order Fulfillment
Prioritize Accuracy	Customer relationship management	Physical Inventory Audit

Arun

Developing your inventory strategy	Cost Savings	Improves Profit Margins
Simplifies Processes and Facilitates Growth	Improves Forecasting	Wholesale cost
Gross Margin Return on Investment	Spot Checking	Better forecasting

Ghousonisan

Safe and secure	Cycle Counting	Inventory Turnover Rate
Reduces Shrinkage	First In, First Out (FIFO)	Prevents Spoilage and Obsolescence
Do Regular and Accurate Stock Counts	Saves time and money	Custom notifications

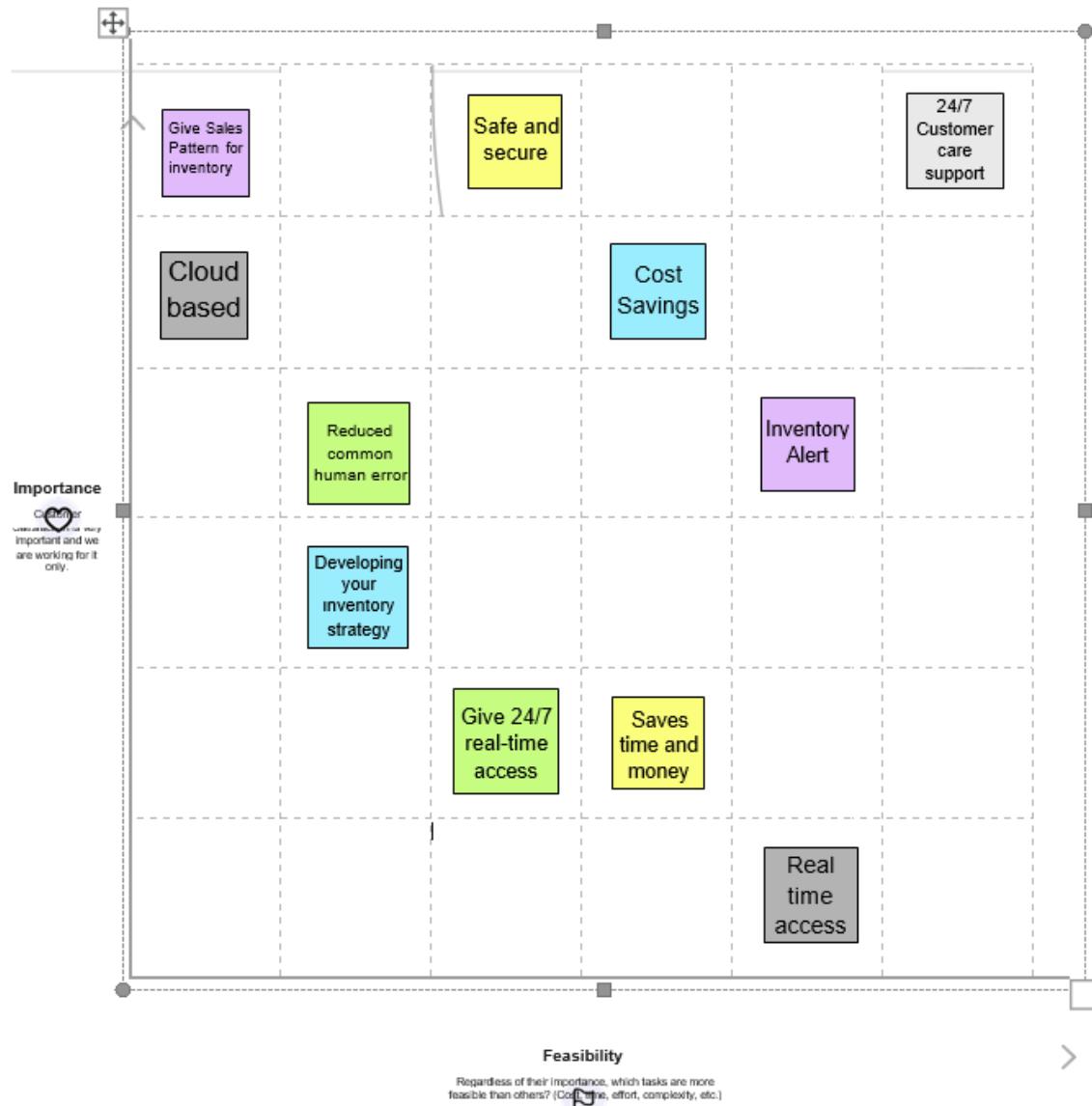
Ilavarasan

Reduced common human error	Give 24/7 real-time access	Decreases Inventory Costs
Eases Supply Chain Management	Combine Sales Data With Inventory Data to Simplify Reporting	Last In, First Out (LIFO)
Retail Demand Forecasting	Maximize Efficiency	Lower Overhead

GROUP IDEA

Give Sales Pattern for inventory	Reduced common human error	Cloud based
Safe and secure	Cost Savings	24/7 Customer care support
Give 24/7 real-time access	Inventory Alert	Saves time and money

IDEA PRIORITIZATION



3.3 Proposed Solution

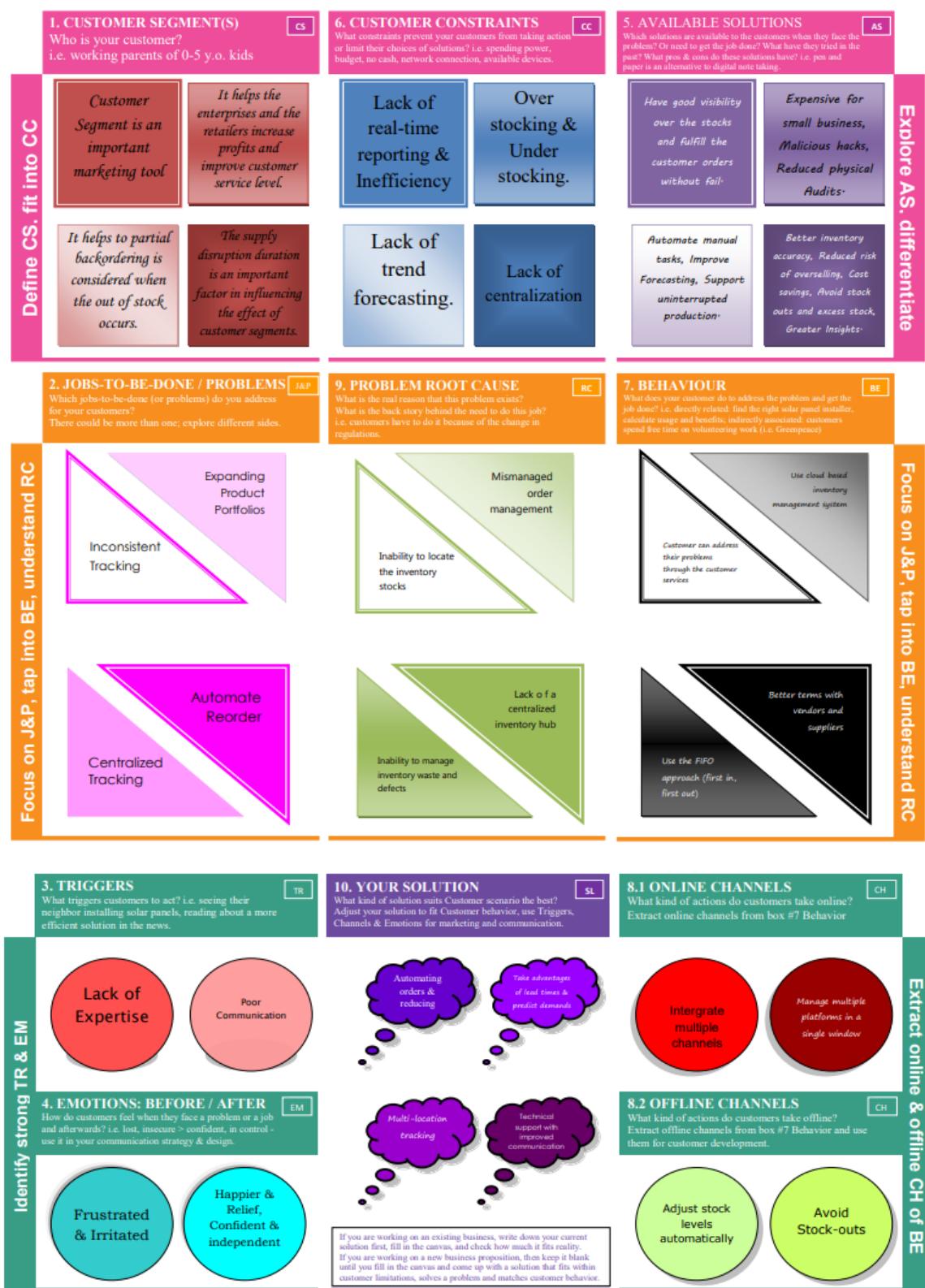
Proposed Solution means **the technical solution to be provided by the Implementation agency in response to the requirements and the objectives of the Project.**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Stock-taking becomes very challenging when you have inventories in multiple locations. Discrete stock data from various locations makes shipping complex, resulting in delays. It's one of the biggest and continual challenges faced by most businesses today.
2.	Idea / Solution description	<p>Central inventory system</p> <p>You can significantly reduce expenses and save a great deal of time by simply creating a centralized inventory hub for your inventory-related data, including stock-taking.</p> <p>This gives you comprehensive visibility and control of inventory and data in one single location, making stock management simple.</p> <p>It also becomes much easier to track the inventory that enters and leaves your business premises.</p>
3.	Novelty / Uniqueness	Provision of better customer service through a focus on skilled workers, better ways to respond to inquiries and requests and also better equipment.
4.	Social Impact / Customer Satisfaction	The availability of product is just one way in which an inventory management system attempts to create customer satisfaction. A comprehensive understanding of the impact of inventory control on customer satisfaction helps you to create an effective inventory management system. Customers want to receive precisely what they ordered within an acceptable time frame, and at a reasonable cost.
5.	Business Model (Revenue Model)	When starting your ecommerce business, you have a lot of decisions to make but, one of the most important is where you'll store your inventory.

		<p>More specifically, you'll need to decide whether to centralize your inventory, or distribute inventory across multiple locations.</p> <p>Centralized inventory is an inventory storage strategy in which you keep all your stock in one central location. This central hub can be a retail store, a warehouse, or another type of storage facility.</p> <p>Because all inventories is stored at a central location, instead of different distribution centers, that location also typically houses all or most of the logistics operations, including order processing and fulfillment.</p>
6.	Scalability of the Solution	<p>The centralized inventory management approach involves a business storing all of their stock at a single warehouse or location, and managing logistical functions such as order fulfillment from that hub.</p> <p>This is the most traditional approach to inventory management, as it simplifies your supply chain management by keeping most critical functions under one roof. Centralized inventory makes inventory control and management simpler.</p> <p>Since you only need to monitor inventory levels in one location, it's easier to keep track of how much inventory you have left, and properly manage fulfillment for multiple channels at once.</p> <p>Operating only one warehouse also consolidates expenses and decreases facilities and operations costs.</p> <ul style="list-style-type: none"> • Streamlined inventory control and management • Lower operational and facilities costs • Easier multi-channel fulfillment.

3.4 Problem Solution fit

The Problem-Solution Fit simply means that **you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.**



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirement

A Functional Requirement (FR) is *a description of the service that the software must offer*. It describes a software system or its component. Functional requirements are **product features that developers must implement to enable the users to achieve their goals**. They define the basic system behavior under specific conditions.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Facebook Registration through phone number
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User login	Login with Fingerprints Login with Password Login with Username
FR-4	Periodic stock checking	Cycle counting Physical counting
FR-5	Promotion	Maintain enough stock on hand to meet demand.
FR-6	Markdown	Show product discount
FR-7	Death stock	Return to the vendor for credits
FR-8	Returns Management System	1. Add it to inventory counts 2. Check for damage

4.2 Non-Functional requirements

Non-Functional Requirements are the constraints or the requirements imposed on the system. They specify the quality attribute of the software. Non-Functional Requirements deal with issues like scalability, maintainability, performance, portability, security, reliability, and many more. Non-Functional Requirements address vital issues of quality for software systems.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The System must be intuitive and simple in the way it displays a relevant data and relationships and the menus of the system are easily navigated by the users with buttons that are easy to understand. If it takes hours for your staff to learn the software, then it's not worth it. You should remember to choose a solution that simplifies inventory management.
NFR-2	Security	Only authorized users can access the system with their name and password of administrator. Inventory is the process of ensuring the safety and optimum management control of stored goods. It is of central importance for optimum warehouse management because the performance of a company stands or falls with the safety and efficiency of a warehouse.
NFR-3	Reliability	Important for several reasons. Your delivery reliability depends on it, but also consider the cost involved. The system must successfully add any recipe, ingredients, vendors, or special occasions given by the user and provide estimations and inventory status in relevance to the newly updated entities. The system must give accurate inventory status to the user continuously. Any inaccuracies are corrected by regularly comparing the actual levels to the levels displayed in the system.
NFR-4	Performance	The goal of inventory performance metrics is to compare actual on-hand dollars versus forecasted cost of goods sold. The system must successfully complete updating the databases, adding new recipes, ingredients, vendors, and occasions every time the user requests such a process All the functions of the system must be available to the user every time the system is turned on, the calculations performed by the system must comply with the norms set by the user and should not vary unless implicitly changed by the user.
NFR-5	Availability	The software will be available only to administrator of the organization and the product as well as other details will be recorded by him. Inventory availability refers to whether a specific item is available for customer orders. Additional information provided by retailers may include the quantity available.
NFR-6	Scalability	The inventory management software or app you choose be able to grow as your business does? The last thing you want is to have to manually re-enter all your inventory if you outgrow your current system. The ability of a system to handle growing amount of work.

CHAPTER 5

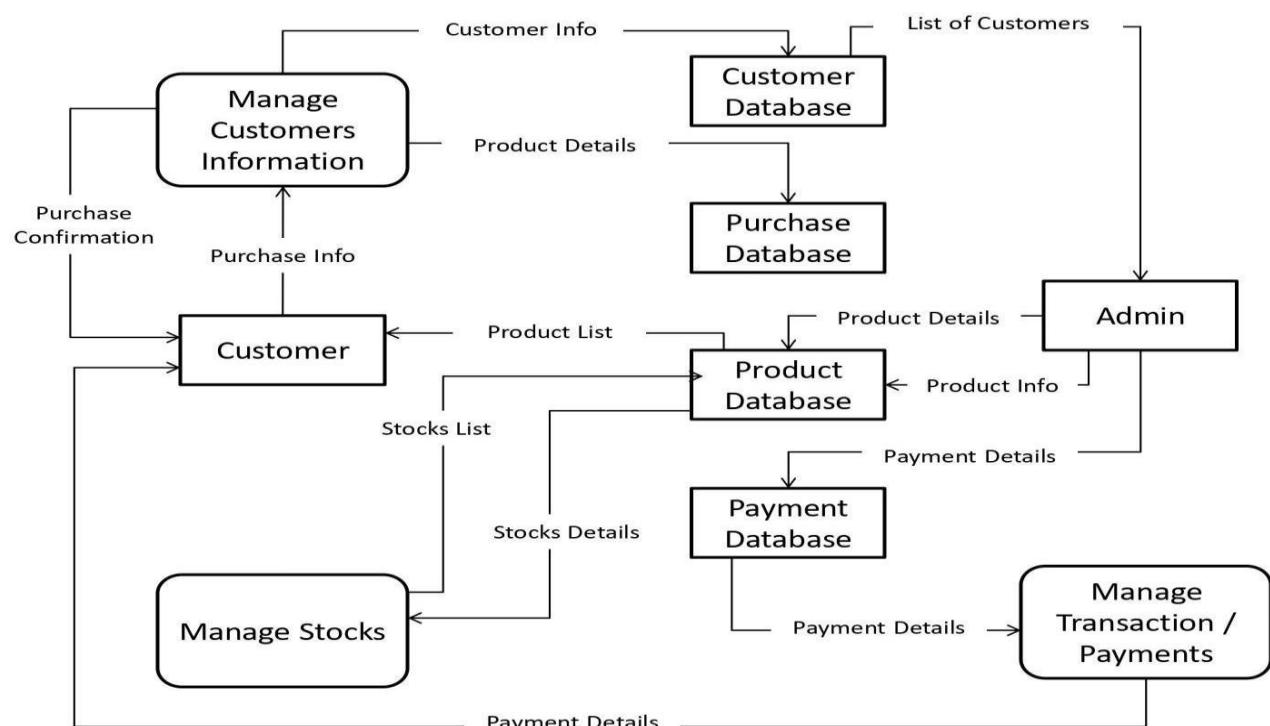
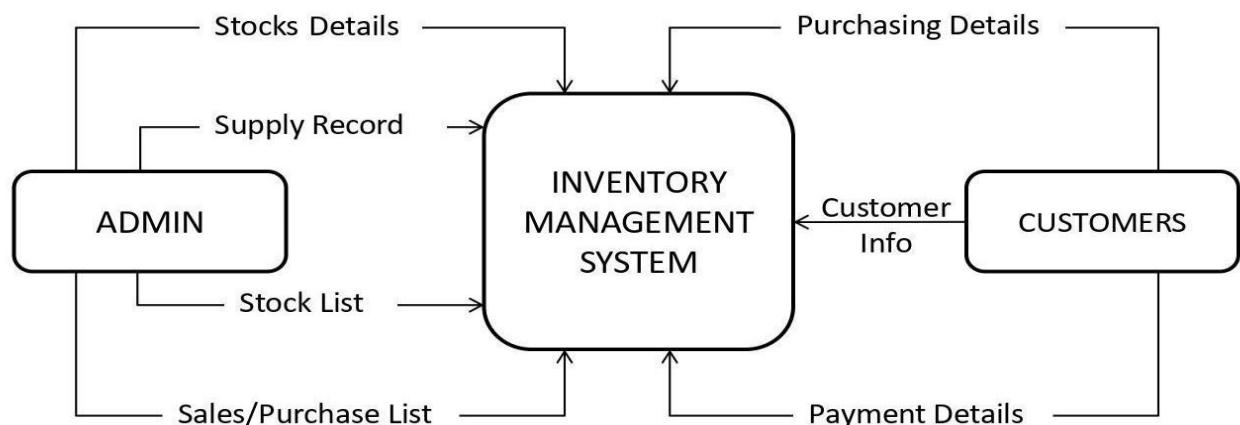
PROJECT DESIGN

5.1 Data Flow Diagrams

A **data flow diagram (DFD)** maps out the flow of information for any process or system. It includes **data inputs and outputs, data stores, and the various subprocesses the data moves through**. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

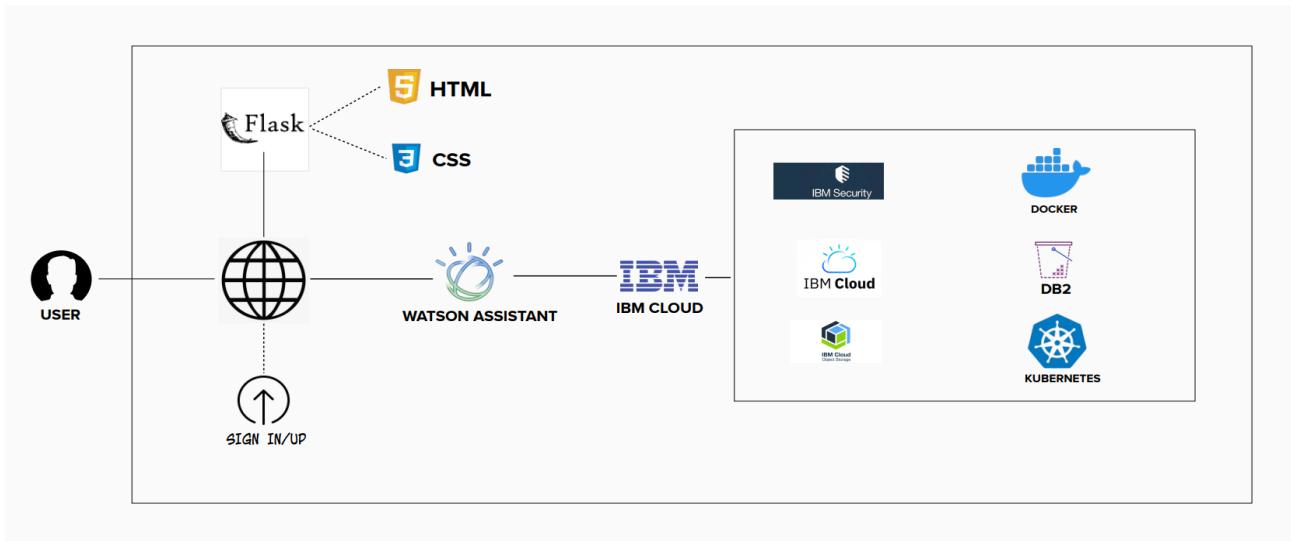
Data Flow Diagrams:

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



5.2 Solution & Technical Architecture

Solution architecture is the process of developing solutions based on predefined processes, guidelines and best practices with the objective that the developed solution fits within the enterprise architecture in terms of information architecture, system portfolios, integration requirements and many more.



Technical Architecture ensures that technology fits into existing computer systems by specifying its hardware, access methods, protocols and more.

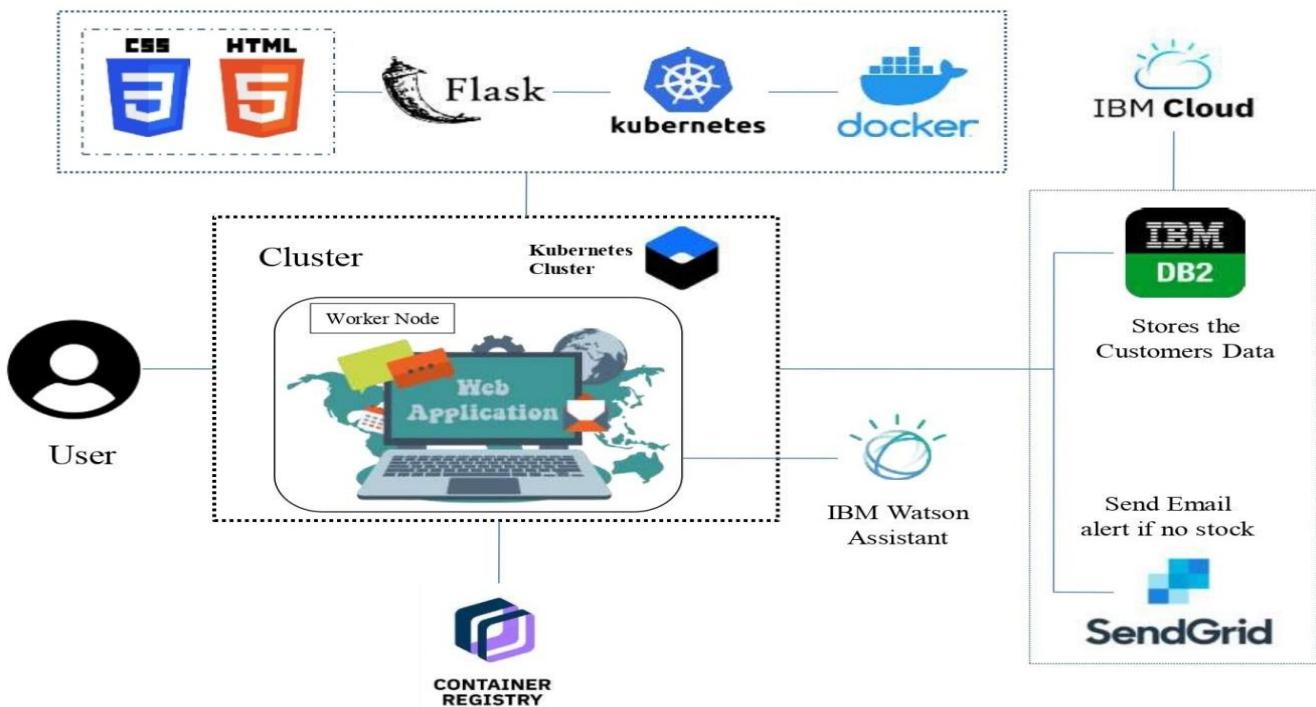


Table-1: Components & Technologies:

S. No.	Component	Description	Technology
1.	User Interface	Web UI, Mobile App, Chatbot	HTML, CSS, JavaScript, Query etc.
2.	Calculating Product Count	By entering barcode details into the application	Barcode Scanner
3.	Alert and notification	Alerting the retailers regarding the low stock count of the product	Send Grid
4.	Cloud Database	Database Service on cloud	IBM DB2
5.	Chat	Chat with Watson assistant	IBM Watson Assistant
6.	File Storage	File Storage Requirements	IBM Object Storage
7.	Infrastructure	Cloud Server Configuration	Cloud Foundry, Kubernetes

Table-2: Application Characteristics:

S. No.	Characteristics	Description	Technology
1.	Open-Source Frameworks	Styling our page, Python flask micro framework	Python Flask, Bootstrap
2.	Security Implementations	For securing our cloud data	SSL Certificates
3.	Scalable Architecture	Three – tier architecture (MVC)	Technology used are: Web server - HTML, CSS, Java script Application server - Python Flask Docker Database Server – IBM DB2
4.	Availability	availability of application	Technology used is IBM Load Balancer
5.	Performance	2 requests per seconds, Use of Local Machine Cache Memory	Technology used is IBM Cloud, CDN (content delivery network)

5.3 User Stories

A **user story** is an informal, general explanation of a software feature written from the perspective of the end user.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming mypassword.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Face book	I can register & access the dashboard with Face book Login	Low	Sprint-2
		USN-4	As a user, I can register for the application throughGmail	I can register and access the dashboard with Gmail login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by enteringemail & password	I can register and access the dashboard with email and password	High	Sprint-1
	Dashboard	USN-6	As a user, I can view the stock availability status	I can view the stock availability status	High	Sprint-2
		USN-7	As a user, I can view the orders status	I can view the order status	Medium	Sprint-3
		USN-8	As a user, I can view the shipping tracking status	I can view the shipping tracking status	Medium	Sprint-4
CustomerCare Executive		USN-9	As a customer care executive, I can view thecomplaints on chat box	I can view the complaints on chatbox	Medium	Sprint-4
		USN-10	As a customer, I should be able solve and reply for the customers queries	I can reply to customer queries in the chat thread	Low	Sprint-4
		USN-11	As a customer, I can close the complaint after assisting	I can close the complaint after assisting	Low	Sprint-4
Administrator		USN-12	As a Administrator, I would take care of registrationsand maintenance of accounts	I can take care of registrations and maintenance of accounts	High	Sprint-3
		USN-13	As a Administrator, I Would resolve issues on Access	I can resolve issues in Access	High	Sprint-2

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint planning is **an event in scrum that kicks off the sprint**. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team. Estimation is **done by the entire team during Sprint Planning Meeting**. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

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	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-2	Dashboard	USN-6	As a user, I can view the stock availability status	2	High	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-3		USN-7	As a user, I can view the orders status	1	Medium	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-4		USN-8	As a user, I can view the shipping Tracking status	2	Medium	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-4		USN-9	As a customer care executive, I can view the complaints on chat box	1	Medium	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-4		USN-10	As a customer, I should be able solve and reply for the customers queries	2	Low	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-4		USN-11	As a customer, I can close the complaint after assisting	1	Low	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-3		USN-12	As a Administrator, I would take care of registrations and maintenance of accounts	2	High	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R
Sprint-2		USN-13	As a Administrator, I Would resolve issues on Access	2	High	M.S. Logesh, Arun. G, Ghousonisan. S, Ilavarasan. R

6.2 Sprint Delivery Schedule

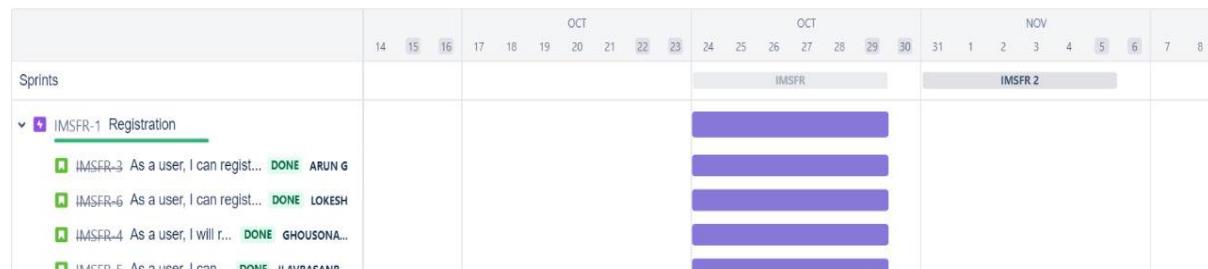
A sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process and something that requires adequate research, planning, and communication.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

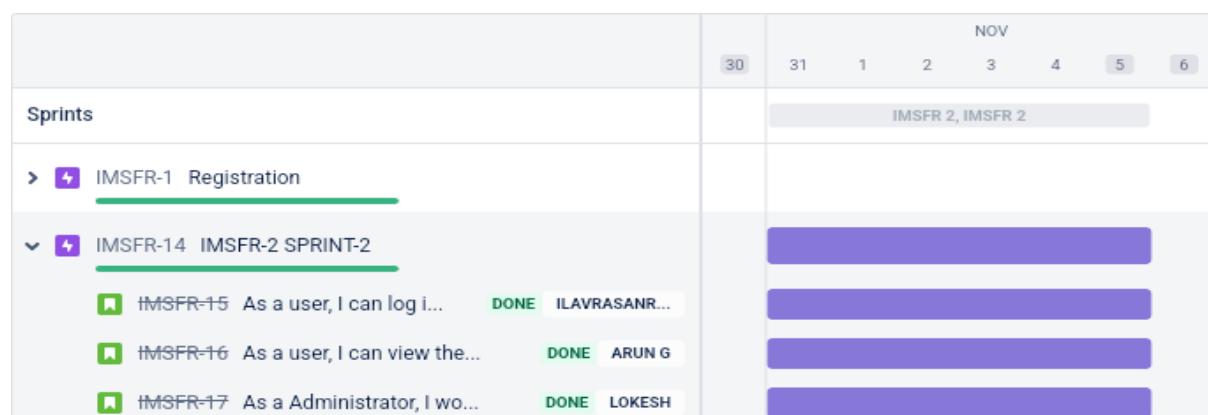
6.3 Reports from JIRA

Jira is a software application used for issue tracking and project management. The tool, developed by the Australian software company Atlassian, has become widely used by agile development teams to track bugs, stories, epics, and other tasks.

Sprint 1



Sprint 2



Sprint 3

		NOV					
Sprints		IMSFR 3 SPRINT-3					
>	⚡ IMSFR-1 Registration	<div style="width: 100%;"><div style="width: 100%;"> </div></div>					
>	⚡ IMSFR-14 IMSFR-2 SPRINT-2	<div style="width: 100%;"><div style="width: 100%;"> </div></div>					
▼	⚡ IMSFR-18 SPRINT 3	<div style="width: 100%;"><div style="width: 100%;"> </div></div>					
	└ IMSFR-19 As a user, I can view...	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE	GHOUSONA...			
	└ IMSFR-20 As a Administrator,...	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE	ILAVRASANR...			

Sprint 4

		NOV					
Sprints		IMSFR 3 SPRINT-4					
>	⚡ IMSFR-1 Registration	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE				
>	⚡ IMSFR-14 IMSFR-2 SPRINT-2	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE				
>	⚡ IMSFR-18 SPRINT 3	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE				
▼	⚡ IMSFR-22 SPRINT 4	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE				
	└ IMSFR-23 As a user, I can view the...	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE	LOKESH			
	└ IMSFR-25 As a customer, I sho...	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE	GHOUSONA...			
	└ IMSFR-26 As a customer, I ca...	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE	ILAVRASANR...			
	└ IMSFR-24 As a customer care exec...	<div style="width: 100%;"><div style="width: 100%;"> </div></div>	DONE	ARUN G			

CHAPTER 7

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

7.1 Feature 1

IBM WATSON ASSISTANT

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.

Code:

```
<script>

    window.watsonAssistantChatOptions = {

        integrationID: "01ebdd1a-1a11-43a6-a065-ed2662bdcf2b", // The ID of this
        integration.

        region: "jp-tok", // The region your integration is hosted in.

        serviceInstanceID: "9a500ac7-1766-402a-b00d-22ffef87956", // The ID of
        your service instance.

        onLoad: function(instance) { instance.render(); }

    };

    setTimeout(function(){

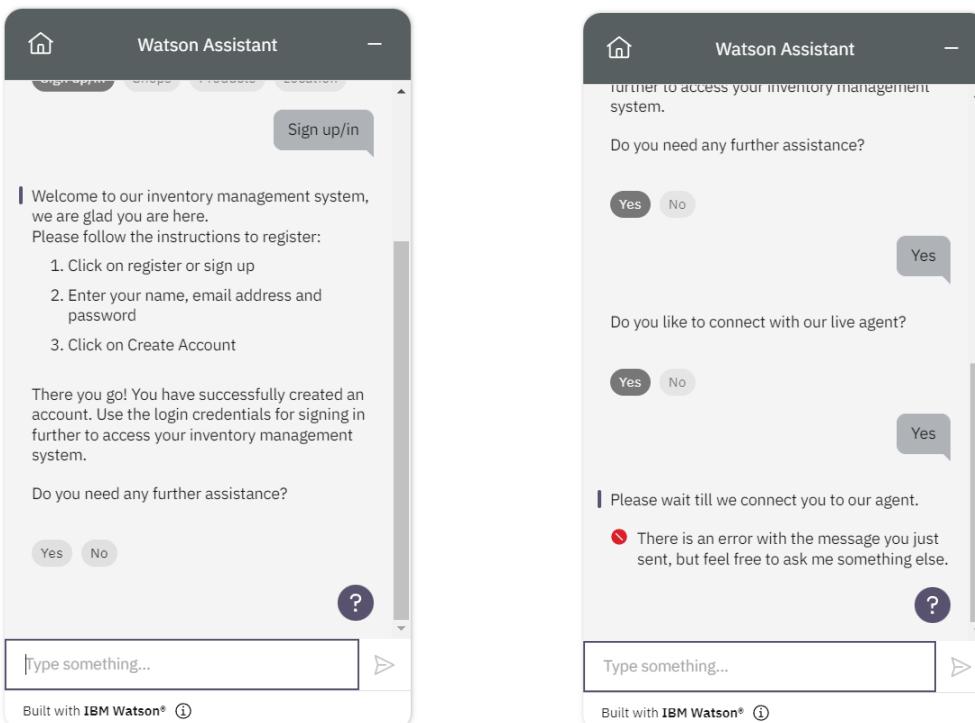
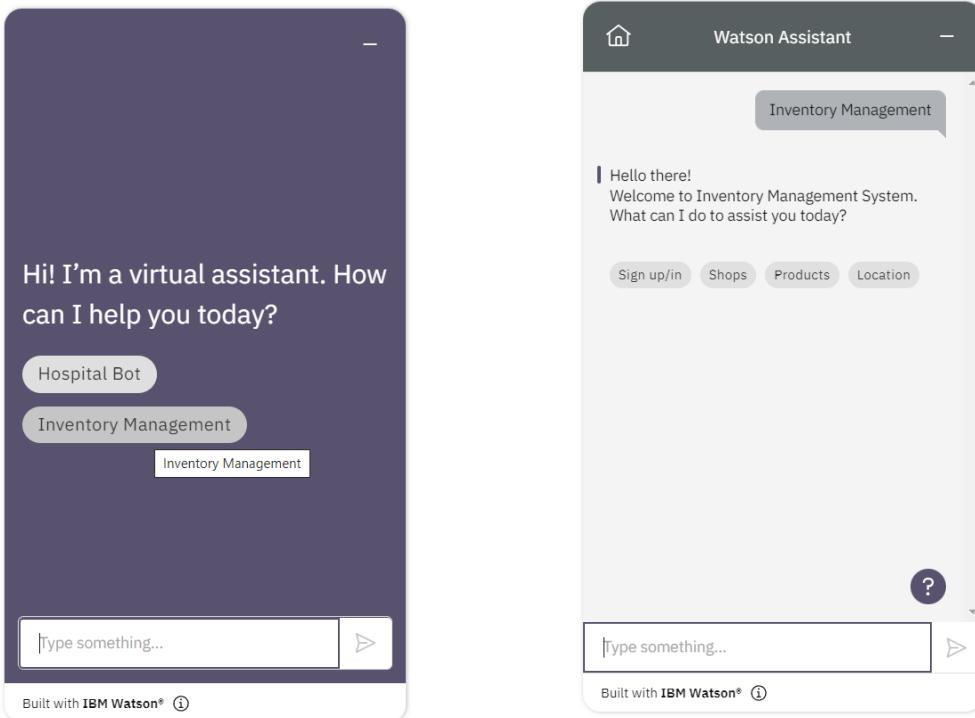
        const t=document.createElement('script');

        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
        (window.watsonAssistantChatOptions.clientVersion || 'latest') +
        "/WatsonAssistantChatEntry.js";

        document.head.appendChild(t);

    });

</script>
```



Watson Assistant

Shops Tab:

- Store is used to show whether the store is open or closed.
- How to add shops.

 - Click on shops tab.
 - Enter the store name.
 - Enter the status of the store, whether the shop is opened or closed.
 - Click add button.

Your store and store status will be shown below!

Do you need any further assistance?

Yes No

Type something...

Built with IBM Watson® ⓘ

Products Tab:

 - Products tab is used to list the products available and the quantity of it.
 - How to use:
 - Click on the products tab.
 - Enter the product's name, quantity and cost of it.
 - Click add.

You can view the product's details below.

Do you need any further assistance?

Yes No

Type something...

Built with IBM Watson® ⓘ

Watson Assistant

Hello there!

Welcome to Inventory Management System.

What can I do to assist you today?

Location Tab:

- Location tab is used to locate the shop location.
- How to use:
 - Click on location tab.
 - Add where the store location is.
 - Click add button.

The shop locations will be shown below.

Do you need any further assistance?

Yes No

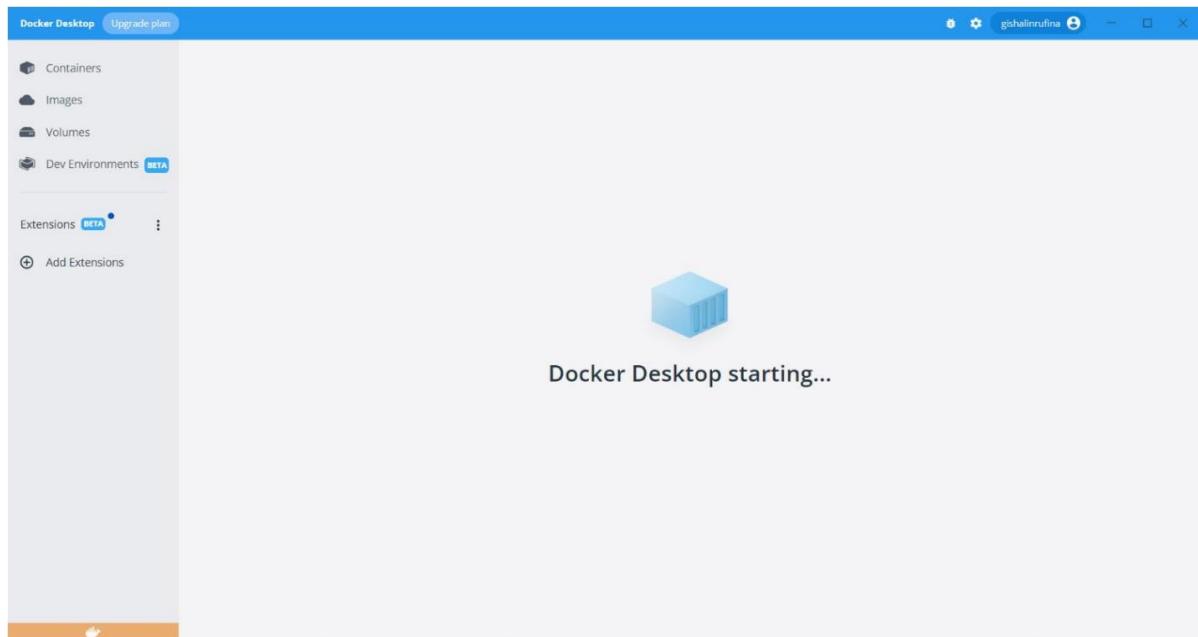
Type something...

Built with IBM Watson® ⓘ

7.2 Feature 2

DOCKER

Docker is a platform designed to help developers build, share, and run modern applications. We handle the tedious setup, so you can focus on the code.



A screenshot of the Docker Desktop application window. The title bar says "Docker Desktop" and "Upgrade plan". The sidebar on the left has icons for Containers, Images (selected), Volumes, Dev Environments (BETA), Extensions (BETA), and a plus sign for "Add Extensions". The main area is titled "Images on disk" and shows a table of local images. The table has columns for NAME, TAG, STATUS, CREATED, SIZE, and ACTIONS. One entry is shown: "docker/getting-started cb90f98fd791" with status "In use", created 7 months ago, and size 28.78 MB. A search bar at the top right contains "Search". At the bottom, it says "Showing 1 items" and "v4.13.1".

7.3 Database Schema (if Applicable)

A database schema defines how data is organized within a relational database. This is inclusive of logical constraints such as, table names, fields, data types, and the relationships between these entities. Schemas commonly use visual representations to communicate the architecture of the database, becoming the foundation for an organization's data management discipline. This process of database schema design is also known as data modelling.

The screenshot shows the IBM Db2 on Cloud interface. The top navigation bar includes 'IBM Db2 on Cloud' and user icons. Below the bar, a menu bar has items: Load Data, Load History, Tables (which is selected and highlighted in blue), Views, Indexes, Aliases, MQTs, Sequences, and Application objects. A search bar says 'Find schemas or tables'. On the left, there's a sidebar with icons for SQL, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The main content area is divided into two sections: 'Schemas' and 'Tables'. The 'Schemas' section shows one schema named 'ZND71133' (Type: User, Tables: 8). The 'Tables' section lists eight tables: ITEM_DETAILS, ORDER_DETAILS, ORDER_STATUS, STOCK_AVAILABILITY, STOCK_DETAILS, SUPPLIER_DETAILS, and USERS, all belonging to the ZND71133 schema. A 'New table' button is visible at the top right of the 'Tables' section.

The screenshot shows the 'Service credentials' page for a resource named 'Db2-id'. The top navigation bar includes 'Resource list / Db2-id' with status 'Active', 'Add tags', 'Details', and 'Actions...'. A sidebar on the left has links: Manage, Getting started, Service credentials (which is selected and highlighted in blue), and Connections. The main content area is titled 'Service credentials' with a sub-instruction: 'You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service.' It includes a search bar 'Search credentials...' and a table. The table has columns: 'Key name' (with a dropdown arrow and a checkbox) and 'Date created'. One row is shown: 'Service credentials-1' (checkbox checked) and '2022-11-17 6:36 PM' (with edit and delete icons). A large blue 'New credential' button is located at the top right of the table area.

CHAPTER 8

TESTING

8.1 Test Cases

A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly.

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(V/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		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		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		Enter URL and click go 2.Click on My Account dropdown button 3. Enter Valid username/email in E-mail text box 4.Enter valid password in password text box 5 Click on login button	Username: lokkylovely@gmail.com Password: lokky1	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 and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in E-mail text box 4.Enter valid password in password text box 5.Click on login button	Username: lokkylovely@gmail.com Password: lokky1	Application should show 'Incorrect email or password' validation message.	Working as expected				
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with Invalid credentials		1. Enter URL and click go 2. Click on My Account dropdown button 3. Enter Valid username/email in E-mail text box 4. Enter Invalid password in password text box 5. Click on login button	Username: lokkylovely@gmail.com Password: lokky1	Application should show 'Incorrect email or password' validation message.	Working as expected	Pass			
LoginPage_TC_006	Functional	Login page	Verify user is able to log into application with Invalid credentials		1. Enter URL and click 2. Click on My Account dropdown button go 3.Enter Invalid username/email in E-mail text box 4.Enter Invalid password in password text box 5. Click on login button	Username: lokkylovely@gmail.com Password: lokky1	Application should show 'Incorrect email or password' validation message.	Working as expected	Pass			

8.2 User Acceptance Testing

User acceptance testing (UAT), also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by its intended audience.

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	9	4	2	3	20
Duplicate	1	0	3	0	4
External	4	3	0	1	6
Fixed	11	6	4	19	36
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	75

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	8	0	0	8
Client Application	20	0	0	20
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	12	0	0	12
Final Report Output	2	0	0	2
Version Control	1	0	0	1

CHAPTER 9

RESULTS

9.1 Performance Metrics

Performance metrics are defined as figures and data representative of an organization's actions, abilities, and overall quality.

S.No.	Project Name	Scope / Feature	NFT-Risk Assessment							
			Functional Changes	Hardware Changes	Software Changes	Impact of Downloading	Load / Volume Changes	Risk Score	Justification	
1	1 Inventory Management System for Retailers	New	Moderate	No Changes	Moderate		> 30 to 50 %	Orange	As we have seen the changes	
2	2 Inventory Management System for Retailers	New	Low	No Changes	Low		> 5 to 10 %	Green	As we have seen the changes	
3	3 Inventory Management System for Retailers	New	High	No Changes	High		> 50 to 70 %	Red	As we have seen the changes	
4	4 Inventory Management System for Retailers	New	Moderate	No Changes	Moderate		> 30 to 50 %	Orange	As we have seen the changes	
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Better Inventory Accuracy
- Reduced Risk of Overselling
- Cost Savings
- Avoiding Stockouts and Excess Stock
- Greater Insights
- Better Terms with Vendors and Suppliers
- More Productivity
- Increased Profits
- A More Organized Warehouse
- Better Customer Experience
- Support Uninterrupted Production
- Scale Inventory as Your Business Grows
- Improve Product Visibility

DISADVANTAGES

- Expensive for Small Businesses
- Complex to Learn
- Risk of System Crashes
- Malicious Hacks
- Reduced Physical Audits

CHAPTER 11

CONCLUSION

Inventory management's major goal is to make ordering, stocking, storing, and using inventory as simple and efficient as possible for firms. You'll always know what things are in stock, how many there are, and where they are if you manage your inventory properly. An inventory system's main function is to keep track of your products and supplies. When you buy inventory, you need to keep track of when you bought it, when you sold it, and how much you have on hand. It also tells you where your inventory is. The Inventory Management System is developed and designed for recording and managing the inventory of an organization. It can also be used for different institution with fewer modification as per requirement. the system can be easily updated as the other institutional requirement may not be integrated on our project. After the continuous effort, testing and debugging the current system is ready to be implemented in an organization. The Inventory management system project that allows user to manage and maintain his/her inventory with ease. The inventory management system has been developed to allow users to add an inventory, delete an inventory, enter inventory quantity and other details, update inventory status and more. The inventory management system has its own intelligently managed support system that allows user to view and manage various inventories added in the system.

CHAPTER 12

FUTURE SCOPE

Inventory management refers to the process of ordering, storing, using, and selling a company's inventory. This includes the management of raw materials, components, and finished products, as well as warehousing and processing of such items. An inventory system's scope can include valuing inventory, measuring inventory change, and forecasting future inventory levels, among other things. At the end of each period, the value of the company's inventory is used to figure out how much money the company makes and loses.

CHAPTER 13

APPENDIX

Source Code

Base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='styles/login.css') }}">
<link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='styles/signup.css') }}">
<link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='styles/style.css') }}">
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" />
<!-- Bootstrap Font Icon CSS -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" />
<style>
*{
  margin: 0;
  box-sizing: border-box;
}
body {
  font-family: Arial, Helvetica, sans-serif;
  margin: 0;
  background: #F6F6F2
}
#navs{
  background-color: black;
}
#header {
  text-align: center;
```

```
}

.header h1 {
    font-size: 50px;
}

#box{
    padding: 80px 100px;
    border: none;
    outline: none;
    width: 100%;
    text-align: center;
    color:#388087

}

#btn{
    padding: 15px 40px;
    border: 5px solid rgb(177, 177, 177);
    font-size: 20px;
    border-radius: 10px;
    margin-left: 500px;
}

a{
    margin: 10px;
}

.logo a{
    font-size: 25px;
    text-decoration: none;
    color: aliceblue;
}

</style>
{%
    block head%
}{%
    endblock %
}
</head>
<body>
    <div class="container-fluid">
        {%
            block body %
        }%
        {%
            endblock %
        }%
        {%
            block nav %
        }%
        {%
            endblock %
        }%
        {%
            block signup %
        }%
        {%
            endblock %
        }%
        {%
            block signin %
        }%
        {%
            endblock %
        }
    </div>
</body>
```

```

</div>

<script src="{{url_for('static', filename='main.js')}}"></script>
<script src="https://kit.fontawesome.com/94af869075.js"
crossorigin="anonymous"></script>

</body>
</html>

```

Dashboard.html

```

{ % extends 'index.html' % }
{ % block head%}<title>Dashboard</title>{ % endblock % }
{ % block dash %}
<main id="main" class="main">

    <div class="pagetitle">
        <h1>Dashboard</h1><br>

    <section class="section dashboard">
        <div class="row">

            <!-- Left side columns -->
            <div class="col-lg-8">
                <div class="row">

                    <!-- Sales Card -->
                    <div class="col-xxl-4 col-md-6">
                        <div class="card info-card sales-card">

                            <div class="card-body">
                                <h5 class="card-title">Shops </h5>
                                <div class="d-flex align-items-center">
                                    <div class="card-icon rounded-circle d-flex align-items-center
justify-content-center">
                                        <i class="bi bi-shop"></i>
                                    </div>
                                    <div class="ps-3">
                                        <h6>No: {{ numshop }}</h6>
                                        <span class="text-dark small pt-1 fw-bold">12%</span> <span
class="text-muted small pt-2 ps-1">increase</span>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>
</main>

```

```

        </div>
        </div>
        </div>
    </div><!-- End Sales Card -->

    <!-- Revenue Card -->
    <div class="col-xxl-4 col-md-6">
        <div class="card info-card revenue-card">
            <div class="card-body">
                <h5 class="card-title">Location </h5>
                <div class="d-flex align-items-center">
                    <div class="card-icon rounded-circle d-flex align-items-center justify-content-center">
                        <i class="bi bi-geo-alt"></i>
                    </div>
                    <div class="ps-3">
                        <h6>No: { {numlocation} }</h6>
                        <span class="text-dark small pt-1 fw-bold">8%</span> <span class="text-muted small pt-2 ps-1">increase</span>
                    </div>
                    </div>
                    </div>
                    </div>
                </div>
            </div><!-- End Revenue Card -->

            <!-- Customers Card -->
            <div class="col-xxl-4 col-xl-12">
                <div class="card info-card customers-card">
                    <div class="card-body">
                        <h5 class="card-title">Products </h5>
                        <div class="d-flex align-items-center">
                            <div class="card-icon rounded-circle d-flex align-items-center justify-content-center">
                                <i class="bi bi-window-dock"></i>
                            </div>
                            <div class="ps-3">
                                <h6>No: { {numprod} }</h6>
                                <span class="text-dark small pt-1 fw-bold">12%</span> <span class="text-muted small pt-2 ps-1">decrease</span>
                            </div>
                            </div>
                            </div>
                            </div>
                        </div>
                    </div><!-- End Customers Card -->

```

```

<!-- Reports -->

</div>
</div><!-- End Left side columns -->

<!-- Right side columns -->
<div class="col-lg-4">

    <!-- Recent Activity -->
    <div class="card">
        <div class="card-body">
            <h5 class="card-title">Recent Activity </h5>
        </div>
    </div><!-- End Recent Activity -->
    <div class="card">
        <div class="card-body">
            <h5 class="card-title">Reports </h5>
        </div>
    </div>
</div><!-- End Right side columns -->
</div>
</section>

</main><!-- End #main -->

{ % endblock %}

```

Index.html

```

{ % extends 'base.html' %}

{ % block head%}<title>Dashboard</title>{ % endblock %}

{ % block nav %}

<header id="header" class="header fixed-top d-flex flex-wrap align-items-center justify-content-between ">
    <div><i class="bi bi-list toggle-sidebar-btn"></i></div>

    <div></div>
    <a href="index.html" class="logo d-flex ">
        
        <span class="text-center d-none d-lg-block">STAR's Inventory</span>
    </a>
    <div><span></span></div>
    <div><span></span></div>

```

```

</header><!-- End Header -->

<!-- ====== Sidebar ====== -->
<aside id="sidebar" class="sidebar">

    <ul class="sidebar-nav" id="sidebar-nav">

        <li class="nav-item">
            <a class="nav-link " href="/dashboard">
                <i class="bi bi-grid"></i>
                <span>Dashboard</span>
            </a>
        </li><!-- End Dashboard Nav -->
        <li class="nav-item">
            <a class="nav-link " data-bs-target="#components-nav" data-bs-
toggle="collapse" href="/shops">
                <i class="bi bi-shop"></i><span>Shops</span>
            </a>
        </li><!-- End Components Nav -->

        <li class="nav-item">
            <a class="nav-link " data-bs-target="#tables-nav" data-bs-
toggle="collapse" href="/products">
                <i class="bi bi-window-dock"></i><span>Products</span>
            </a>
        </li><!-- End Tables Nav -->

        <li class="nav-item">
            <a class="nav-link " data-bs-target="#forms-nav" data-bs-
toggle="collapse" href="/location">
                <i class="bi bi-geo-alt"></i><span>Location</span>
            </a>
        </li><!-- End Forms Nav -->

        <li class="nav-item">
            <a class="nav-link " data-bs-target="#charts-nav" data-bs-
toggle="collapse" href="#">
                <i class="bi bi-question-square"></i><span>Help</span>
            </a>
        </li><!-- End Charts Nav -->

        <li class="nav-item">

```

```

<a class="nav-link " href="/main">
  <i class="bi bi-box-arrow-in-left"></i>
  <span>Logout</span>
</a>
</li><!-- End Login Page Nav -->

</ul>

</aside><!-- End Sidebar-->

<!--Chat bot script-->
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "01ebdd1a-1a11-43a6-a065-ed2662bdcf2b", // The ID of this
    integration,
    region: "jp-tok", // The region your integration is hosted in.
    serviceInstanceID: "9a500ac7-1766-402a-b00d-22ffef87956", // 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>

{%
  block shops %
}
{%
  endblock %
}
{%
  block prod %
}
{%
  endblock %
}
{%
  block loc %
}
{%
  endblock %
}
{%
  block dash %
}
{%
  endblock %

}

{%
  endblock %
}
```

Location.html

```
{% extends 'index.html' %}  
{% block head%}<title>Location</title>{% endblock %}  
{% block loc %}  
  
<main id="main" class="main">  
  
    <div class="pagetitle">  
        <h1>Location</h1><br>  
        <form class="row g-3 needs-validation" action = "{{ url_for('location') }}"  
method = "POST">  
            <div class="col-12">  
                <label class="form-add-label">Location</label>  
                <div class="input-group has-validation">  
                    <input type="text" name="name" class="form-add-  
control" required>  
                </div>  
            </div>  
  
            <div class="col-12" >  
                <button class="btn btn-outline-dark btn-lg" type="submit"  
>Add</button>  
            </div>  
  
        </form>  
  
        <table class="table table-striped">  
            <thead>  
                <tr>  
                    <th>Location</th>  
                    <th>Options</th>  
                </tr>  
            </thead>  
            {% for row in location %}  
            <tbody>  
                <tr>  
                    <td>{{ row['NAME'] }}</td>  
                    <td><button class="btn btn-danger"  
id="del"><a href="/dellocation/{{ row['NAME'] }}">Delete</a></button></td>  
                </tr>  
            {% endfor %}  
        </tbody>  
    </table>
```

```
</main><!-- End #main -->
```

```
{% endblock %}
```

Main.html

```
{% extends 'base.html' %}

{% block head%}<title>Main</title>{% endblock %}

{% block body %}

<div>
    <header id="header" class="header fixed-top d-flex">
        <div class="d-flex align-items-center justify-content-between">
            <a href="index.html" class="logo d-flex align-items-center">
                
                <span class="d-none d-lg-block">START's Inventory</span>
            </a>
        </div>
        <nav class="header-nav ms-auto">
            <a href="/signin"><button class="btn btn-light" id="sign-btn">signin</button></a>
        </nav><!-- End Icons Navigation -->
    </header>
    <div class="slogan">
        <b><p>Inventory management for Retailers</p></b>
    </div>
    <div class="benifit">
        <p>Inventory refers to the goods stocked for future use.<br>
        Every retail chain has its own warehouse to stock the merchandise
        to<br>
        be used when the existing stock replenishes.</p>
        <a href="/signin"><button class="btn btn-dark" id="sign-btn">Get
        started</button></a>
    </div>
    <div id="image">
        </img>
    </div>
</div>
```

```

<!--Chat bot script-->
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "01ebdd1a-1a11-43a6-a065-ed2662bdcf2b", // The ID of this
integration.
    region: "jp-tok", // The region your integration is hosted in.
    serviceInstanceID: "9a500ac7-1766-402a-b00d-22ffef87956", // 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>

{ % endblock %}

```

Products.html

```

{ % extends 'index.html' % }
{ % block head%}<title>Products</title>{ % endblock % }
{ % block prod % }

<main id="main" class="main">

  <div class="pagetitle">
    <h1>Products</h1><br>
    <form class="row g-3 needs-validation" novalidate action = "{ {
url_for('products') } }" method = "POST">
      <div class="col-12">
        <label class="form-add-label">Product Name</label>
        <div class="input-group has-validation">
          <input type="text" name="name" class="form-add-control" required
>
        </div>
      </div>
      <div class="col-12">

```

```

<label class="form-add-label">Qantity</label>
<div class="input-group has-validation">
    <input type="text" name="quantity" class="form-add-control" required
>
    </div>
</div>
<div class="col-12">
    <label class="form-add-label">Cost</label>
    <div class="input-group has-validation">
        <input type="text" name="cost" class="form-add-control" required
>
        </div>
    </div>

    <div class="col-12" >
        <button class="btn btn-outline-dark btn-lg" type="submit"
>Add</button>
    </div>
</form>

<table class="table table-striped">
    <thead>
        <tr>
            <th>Name</th>
            <th>Quantity</th>
            <th>Cost</th>
        </tr>
    </thead>
    { % for row in products %}
    <tbody>
        <tr>
            <td>{{row["NAME"]}}</td>
            <td>{{row["QUANTITY"]}}</td>
            <td>{{row["COST"]}}</td>
            <td><button class="btn btn-danger"
id="del"><a href="/delete/{{row['NAME']}}">Delete</a></button></td>
        </tr>
    </tbody>
    { % endfor %}
</table>

</main><!-- End #main -->

{ % endblock %

```

Shops.html

```
{% extends 'index.html' %}  
{% block head%}<title>Shops</title>{% endblock %}  
{% block shops %}  
  
<main id="main" class="main">  
  
    <div class="pagetitle">  
        <h1>Stores</h1><br>  
        <form class="row g-3 needs-validation" novalidate action = "{  
url_for('shops') }" method = "POST">  
            <div class="col-12">  
                <label class="form-add-label">Store-Name</label>  
                <div class="input-group has-validation">  
                    <input type="text" name="name" class="form-add-control" required  
>  
                </div>  
            </div>  
            <div class="col-12">  
                <label class="form-add-label">Status</label>  
                <div class="input-group has-validation">  
  
                    <select id="status" name="status" class="form-add-control">  
                        <option value="select">Select</option>  
                        <option value="Active">Active</option>  
                        <option value="Inactive">Inactive</option>  
                    </select>  
                </div>  
            </div>  
            <div class="col-12" >  
                <button class="btn btn-outline-dark btn-lg" type="submit">  
>Add</button>  
            </div>  
  
        </form>  
  
        <table class="table table-striped">  
            <thead>  
                <tr>  
                    <th>Name</th>  
                    <th>Status</th>  
                    <th>Options</th>
```

```

</tr>
</thead>
{ % for row in shops % }
<tbody>
<tr>
<td>{{row['NAME']}}</td>
<td>{{row['Status']}}</td>
<td><button class="btn btn-danger"
id="del"><a href="/delete/{{row['NAME']}}">Delete</a></button></td>
</tbody>

{ % endfor % }
</table>

</main><!-- End #main -->

{ % endblock % }

```

Signin.html

```

{ % extends 'base.html' % }
{ % block head % } <title>SignIn</title> { % endblock % }
{ % block signin % }
<main>
<div class="container-fluid" id="signin-back">
<section class="section register min-vh-100 d-flex flex-column align-items-center justify-content-center py-4">
<div class="container">
<div class="row justify-content-center">
<div class="col-lg-4 col-md-6 d-flex flex-column align-items-center justify-content-center">
<div class="d-flex justify-content-center py-4">
<a href="/main" class="logo d-flex align-items-center w-auto">

<span class="d-none d-lg-block">Inventory</span>
</a>
</div>
<div class="card mb-3">
<div class="card-body">
<div class="pt-4 pb-2">
<h5 class="card-title text-center pb-0 fs-4">Login to Your
Account</h5>
</div>

```

```

<form class="row g-3 needs-validation"
novalidate action="{ { url_for('signin') } }" method="post">
    <div class="col-12">
        <label for="yourUsername" class="form-label">Username</label>
        <div class="input-group has-validation">
            <input type="text" name="username" class="form-control"
id="yourUsername" required placeholder="Enter your username">
        </div>
    </div>
    <div class="col-12">
        <label for="yourPassword" class="form-label">Password</label>
        <input type="password" name="password" class="form-control"
id="yourPassword" required placeholder="Enter your password">
    </div>
    <div class="col-12">
        <div class="form-check">
            <input class="form-check-input" type="checkbox"
name="remember" value="true" id="rememberMe">
            <label class="form-check-label" for="rememberMe">Remember
me</label>
        </div>
    </div>
    <div class="col-12">
        <button class="btn w-100" type="submit" style="background-
color: #343a40;color:white;">Login</button>
    </div>
    <div class="col-12">
        <p class="small mb-0">Don't have account? <a
href="/signup">Create an account</a></p>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</section>
</div>
</main>
{ % endblock %

```

Signup.html

```
{% extends 'base.html' %}  
{% block head %} <title>SignUp</title> {%endblock%}  
{% block signup %}  
<main>  
    <div class="container-fluid" id="signin-back">  
        <section class="section register min-vh-100 d-flex flex-column align-items-center justify-content-center py-4">  
            <div class="container">  
                <div class="row justify-content-center">  
                    <div class="col-lg-4 col-md-6 d-flex flex-column align-items-center justify-content-center">  
                        <div class="d-flex justify-content-center py-4">  
                            <a href="/main" class="logo d-flex align-items-center w-auto">  
                                  
                                <span class="d-none d-lg-block">Inventory</span>  
                            </a>  
                        </div>  
                        <div class="card mb-3">  
                            <div class="card-body">  
                                <div class="pt-4 pb-2">  
                                    <h5 class="card-title text-center pb-0 fs-4">Create an  
Account</h5>  
                                    <p class="text-center small">Enter your personal details to create  
account</p>  
                                </div>  
                                <form class="row g-3 needs-validation" novalidate  
action="{{ url_for('signup') }}" method="POST">  
                                    <div class="col-12">  
                                        <label for="yourUsername" class="form-label">Name</label>  
                                        <div class="input-group has-validation">  
                                            <input type="text" name="username" class="form-control" id="yourUsername" required placeholder="Enter your name">  
                                        </div>  
                                    </div>  
                                    <div class="col-12">  
                                        <label for="yourEmail" class="form-label">Your Email</label>  
                                        <input type="email" name="email" class="form-control" id="yourEmail" required placeholder="Enter your email">  
                                    </div>  
                                    <div class="col-12">  
                                        <label for="yourPassword" class="form-label">Password</label>  
                                        <input type="password" name="password" class="form-control" id="yourPassword" required placeholder="Enter your password">  
                                    </div>  
                                </form>  
                            </div>  
                        </div>  
                    </div>  
                </div>  
            </div>  
        </section>  
    </div>  
</main>
```

Login.css

```
body {  
    font-family: Arial, Helvetica, sans-serif;  
    margin: 0;  
    background: #F6F6F2  
}  
.box{  
    border-radius:10px;  
    width:600px;  
    height: 400px;  
    margin: 0px 100px;  
    padding:5px;  
    box-shadow: rgba(0, 0, 0, 0.3) 0px 19px 38px, rgba(0, 0, 0, 0.22) 0px 15px  
12px ;  
}  
.login {  
    padding: 10px 0px 30px 30px;  
    position: absolute;  
    max-width: 500px;  
}  
.name{
```

```
margin-right: 200px;
font-size: 2em;
font-family: Arial Black, sans-serif;
color:#388087;
margin-top:10px;
text-align: right;
}
.user label{
font-size: 20px;
margin-left: 10px;
font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
color:rgb(0, 0, 0)
}
.pass label{
font-size: 20px;
margin-left: 10px;
font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
color:rgb(0, 0, 0)
}
.user input {
width: 100%;
padding: 10px;
margin: 10px 30px 15px 0px;
border: none ;
outline: none;
background: #ecf0f3;
height: 50px;
font-size: 16px;
border-radius: 50px;
}
.pass input {
width: 100%;
padding: 10px;
margin: 10px 30px 30px 0px;
border: none ;
outline: none;
background: #ecf0f3;
height: 50px;
font-size: 16px;
border-radius: 50px;
}
```

```
.btnlog {  
    padding: 15px 40px;  
    border: none;  
    outline: none;  
    border-radius: 50px;  
    font-size: 18px;  
    color: white;  
    background-color: rgb(0, 0, 0);  
    margin-left: 50px ;  
}  
.btnlog:hover{  
    color: rgb(168, 169, 171);  
}  
.ca {  
    font-size: 18px;  
    border: none;  
    outline:none;  
    background: transparent;  
    text-decoration: none;  
    color: rgb(18, 125, 212);  
    margin-left: 100px;  
    transition: 0.2s;  
}  
  
.ca:hover{  
    color: red;  
    font-size: 20px;  
    transition: 0.2s;  
}  
form h1 {  
    font-size: 30px;  
    color:#388087;  
    font-family:'Poppins', sans-serif;  
    position:relative;  
}  
  
.log{  
    width: 70%;  
    margin-left: 700px;  
    height: auto;  
    color: black;  
}
```

Style.css

```
/***
 * Template Name: NiceAdmin - v2.4.1
 * Template URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/
 * Author: BootstrapMade.com
 * License: https://bootstrapmade.com/license/
 */

/*
# General
*/
:root {
    scroll-behavior: smooth;
}

body {
    font-family: "Open Sans", sans-serif;
    background: #f6f9ff;
    color: #444444;
}

a {
    color: #4154f1;
    text-decoration: none;
}

a:hover {
    color: #717ff5;
    text-decoration: none;
}

h1,
h2,
h3,
h4,
h5,
h6 {
    font-family: "Nunito", sans-serif;
}

/*
# Main
*/
```

```
#main {  
    margin-top: 60px;  
    padding: 20px 30px;  
    transition: all 0.3s;  
}  
  
@media (max-width: 1199px) {  
    #main {  
        padding: 20px;  
    }  
}  
  
/*-----  
# Page Title  
-----*/  
.pagetitle {  
    margin-bottom: 10px;  
}  
  
.pagetitle h1 {  
    font-size: 24px;  
    margin-bottom: 0;  
    font-weight: 600;  
    color: #012970;  
}  
  
/*-----  
# Back to top button  
-----*/  
.back-to-top {  
    position: fixed;  
    visibility: hidden;  
    opacity: 0;  
    right: 15px;  
    bottom: 15px;  
    z-index: 99999;  
    background: #4154f1;  
    width: 40px;  
    height: 40px;  
    border-radius: 4px;  
    transition: all 0.4s;  
}  
  
.back-to-top i {  
    font-size: 24px;
```

```

color: #fff;
line-height: 0;
}

.back-to-top:hover {
background: #6776f4;
color: #fff;
}

.back-to-top.active {
visibility: visible;
opacity: 1;
}

/*
-----#
# Override some default Bootstrap stylings
-----*/
/* Dropdown menus */
.dropdown-menu {
border-radius: 4px;
padding: 10px 0;
-webkit-animation-name: dropdown-animate;
animation-name: dropdown-animate;
-webkit-animation-duration: 0.2s;
animation-duration: 0.2s;
-webkit-animation-fill-mode: both;
animation-fill-mode: both;
border: 0;
box-shadow: 0 5px 30px 0 rgba(82, 63, 105, 0.2);
}

.dropdown-menu .dropdown-header,
.dropdown-menu .dropdown-footer {
text-align: center;
font-size: 15px;
padding: 10px 25px;
}

.dropdown-menu .dropdown-footer a {
color: #444444;
text-decoration: underline;
}

.dropdown-menu .dropdown-footer a:hover {
text-decoration: none;
}

```

```
}

.dropdown-menu .dropdown-divider {
  color: #a5c5fe;
  margin: 0;
}

.dropdown-menu .dropdown-item {
  font-size: 14px;
  padding: 10px 15px;
  transition: 0.3s;
}

.dropdown-menu .dropdown-item i {
  margin-right: 10px;
  font-size: 18px;
  line-height: 0;
}

.dropdown-menu .dropdown-item:hover {
  background-color: #f6f9ff;
}

@media (min-width: 768px) {
  .dropdown-menu-arrow::before {
    content: "";
    width: 13px;
    height: 13px;
    background: #fff;
    position: absolute;
    top: -7px;
    right: 20px;
    transform: rotate(45deg);
    border-top: 1px solid #eaedf1;
    border-left: 1px solid #eaedf1;
  }
}

@-webkit-keyframes dropdown-animate {
  0% {
    opacity: 0;
  }

  100% {
    opacity: 1;
  }
}
```

```
}

0% {
  opacity: 0;
}
}

@keyframes dropdown-animate {
0% {
  opacity: 0;
}
100% {
  opacity: 1;
}

0% {
  opacity: 0;
}
}

/* Light Backgrounds */
.bg-primary-light {
  background-color: #cfe2ff;
  border-color: #cfe2ff;
}

.bg-secondary-light {
  background-color: #e2e3e5;
  border-color: #e2e3e5;
}

.bg-success-light {
  background-color: #d1e7dd;
  border-color: #d1e7dd;
}

.bg-danger-light {
  background-color: #f8d7da;
  border-color: #f8d7da;
}

.bg-warning-light {
  background-color: #fff3cd;
  border-color: #fff3cd;
```

```
}
```

```
.bg-info-light {
  background-color: #cff4fc;
  border-color: #cff4fc;
}
```

```
.bg-dark-light {
  background-color: #d3d3d4;
  border-color: #d3d3d4;
}
```

```
/* Card */
.card {
  margin-bottom: 30px;
  border: none;
  border-radius: 5px;
  box-shadow: 0px 0 30px rgba(1, 41, 112, 0.1);
}
```

```
.card-header,
.card-footer {
  border-color: #ebeef4;
  background-color: #fff;
  color: #798eb3;
  padding: 15px;
}
```

```
.card-title {
  padding: 20px 0 15px 0;
  font-size: 18px;
  font-weight: 500;
  color: #012970;
  font-family: "Poppins", sans-serif;
}
```

```
.card-title span {
  color: #899bbd;
  font-size: 14px;
  font-weight: 400;
}
```

```
.card-body {
  padding: 0 20px 20px 20px;
}
```

```
.card-img-overlay {  
  background-color: rgba(255, 255, 255, 0.6);  
}  
  
/* Alerts */  
.alert-heading {  
  font-weight: 500;  
  font-family: "Poppins", sans-serif;  
  font-size: 20px;  
}  
  
/* Close Button */  
.btn-close {  
  background-size: 25%;  
}  
  
.btn-close:focus {  
  outline: 0;  
  box-shadow: none;  
}  
  
/* Accordion */  
.accordion-item {  
  border: 1px solid #ebeef4;  
}  
  
.accordion-button:focus {  
  outline: 0;  
  box-shadow: none;  
}  
  
.accordion-button:not(.collapsed) {  
  color: #012970;  
  background-color: #f6f9ff;  
}  
  
.accordion-flush .accordion-button {  
  padding: 15px 0;  
  background: none;  
  border: 0;  
}  
  
.accordion-flush .accordion-button:not(.collapsed) {  
  box-shadow: none;  
}
```

```
color: #4154f1;
}

.accordion-flush .accordion-body {
  padding: 0 0 15px 0;
  color: #3e4f6f;
  font-size: 15px;
}

/* Breadcrumbs */
.breadcrumb {
  font-size: 14px;
  font-family: "Nunito", sans-serif;
  color: #899bbd;
  font-weight: 600;
}

.breadcrumb a {
  color: #899bbd;
  transition: 0.3s;
}

.breadcrumb a:hover {
  color: #51678f;
}

.breadcrumb .breadcrumb-item::before {
  color: #899bbd;
}

.breadcrumb .active {
  color: #51678f;
  font-weight: 600;
}

/* Bordered Tabs */
.nav-tabs-bordered {
  border-bottom: 2px solid #ebeef4;
}

.nav-tabs-bordered .nav-link {
  margin-bottom: -2px;
  border: none;
  color: #2c384e;
}
```

```
.nav-tabs-bordered .nav-link:hover,  
.nav-tabs-bordered .nav-link:focus {  
    color: #4154f1;  
}  
  
.nav-tabs-bordered .nav-link.active {  
    background-color: #fff;  
    color: #4154f1;  
    border-bottom: 2px solid #4154f1;  
}  
  
/*-----  
# Header  
-----*/  
  
#signin-back{  
    background-color: #343a40;  
}  
  
.logo {  
    line-height:1;  
    text-decoration: none;  
}  
  
@media (min-width: 1200px) {  
    .logo {  
        width: 280px;  
    }  
}  
  
.logo img {  
    max-height: 35px;  
    margin-right: 6px;  
}  
.logo a{  
    text-align: center;  
}  
.logo span {  
    font-size: 26px;  
    font-weight: 700;  
    color: #ffffff;  
    font-family: "Nunito", sans-serif;  
}  
  
#sign-btn{  
    margin-top: 10px;
```

```
margin-right: 20px;
font-size: 18px;
padding: 5px 20px;
}
.data-img img{
width:50%;
height: 60%;
position: absolute;
bottom: 50px;
right: 16px;

}
.slogan{
margin-top: 100px;
color: #2b2b2b;
font-size: 30px;
text-align: center;
}
.benifit{
margin-top: 100px;
font-size: 25px;
color:rgb(91, 91, 91);
text-align: center;
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.header {
transition: all 0.5s;
z-index: 997;
height: 60px;
box-shadow: 0px 2px 20px rgba(1, 41, 112, 0.1);
background-color: #343a40;
padding-left: 20px;
/* Toggle Sidebar Button */
/* Search Bar */
}

.header .toggle-sidebar-btn {
font-size: 32px;
padding-left: 10px;
cursor: pointer;
color: #ffffff;
}

.header .search-bar {
min-width: 360px;
```

```
padding: 0 20px;
}

@media (max-width: 1199px) {
.header .search-bar {
position: fixed;
top: 50px;
left: 0;
right: 0;
padding: 20px;
box-shadow: 0px 0px 15px 0px rgba(1, 41, 112, 0.1);
background: white;
z-index: 9999;
transition: 0.3s;
visibility: hidden;
opacity: 0;
}

.header .search-bar-show {
top: 60px;
visibility: visible;
opacity: 1;
}
}

.header .search-form {
width: 100%;
}

.header .search-form input {
border: 0;
font-size: 14px;
color: #012970;
border: 1px solid rgba(1, 41, 112, 0.2);
padding: 7px 38px 7px 8px;
border-radius: 3px;
transition: 0.3s;
width: 100%;
}

.header .search-form input:focus,
.header .search-form input:hover {
outline: none;
box-shadow: 0 0 10px 0 rgba(1, 41, 112, 0.15);
border: 1px solid rgba(1, 41, 112, 0.3);
```

```
}

.header .search-form button {
    border: 0;
    padding: 0;
    margin-left: -30px;
    background: none;
}

.header .search-form button i {
    color: #012970;
}

/*
-----#
# Header Nav
-----*/
.header-nav ul {
    list-style: none;
}

.header-nav>ul {
    margin: 0;
    padding: 0;
}

.header-nav .nav-icon {
    font-size: 22px;
    color: #012970;
    margin-right: 25px;
    position: relative;
}

.header-nav .nav-profile {
    color: #012970;
}

.header-nav .nav-profile img {
    max-height: 36px;
}

.header-nav .nav-profile span {
    font-size: 14px;
    font-weight: 600;
}
```

```
.header-nav .badge-number {
  position: absolute;
  inset: -2px -5px auto auto;
  font-weight: normal;
  font-size: 12px;
  padding: 3px 6px;
}

.header-nav .notifications {
  inset: 8px -15px auto auto !important;
}

.header-nav .notifications .notification-item {
  display: flex;
  align-items: center;
  padding: 15px 10px;
  transition: 0.3s;
}

.header-nav .notifications .notification-item i {
  margin: 0 20px 0 10px;
  font-size: 24px;
}

.header-nav .notifications .notification-item h4 {
  font-size: 16px;
  font-weight: 600;
  margin-bottom: 5px;
}

.header-nav .notifications .notification-item p {
  font-size: 13px;
  margin-bottom: 3px;
  color: #919191;
}

.header-nav .notifications .notification-item:hover {
  background-color: #f6f9ff;
}

.header-nav .messages {
  inset: 8px -15px auto auto !important;
}

.header-nav .messages .message-item {
```

```
padding: 15px 10px;
transition: 0.3s;
}

.header-nav .messages .message-item a {
  display: flex;
}

.header-nav .messages .message-item img {
  margin: 0 20px 0 10px;
  max-height: 40px;
}

.header-nav .messages .message-item h4 {
  font-size: 16px;
  font-weight: 600;
  margin-bottom: 5px;
  color: #444444;
}

.header-nav .messages .message-item p {
  font-size: 13px;
  margin-bottom: 3px;
  color: #919191;
}

.header-nav .messages .message-item:hover {
  background-color: #f6f9ff;
}

.header-nav .profile {
  min-width: 240px;
  padding-bottom: 0;
  top: 8px !important;
}

.header-nav .profile .dropdown-header h6 {
  font-size: 18px;
  margin-bottom: 0;
  font-weight: 600;
  color: #444444;
}

.header-nav .profile .dropdown-header span {
  font-size: 14px;
```

```
}

.header-nav .profile .dropdown-item {
    font-size: 14px;
    padding: 10px 15px;
    transition: 0.3s;
}

.header-nav .profile .dropdown-item i {
    margin-right: 10px;
    font-size: 18px;
    line-height: 0;
}

.header-nav .profile .dropdown-item:hover {
    background-color: #f6f9ff;
}

/*
-----#
# Sidebar
-----*/
.sidebar {
    position: fixed;
    top: 60px;
    left: 0;
    bottom: 0;
    width: 300px;
    z-index: 996;
    transition: all 0.3s;
    padding: 20px;
    overflow-y: auto;
    scrollbar-width: thin;
    scrollbar-color: #aab7cf transparent;
    box-shadow: 0px 0px 20px rgba(1, 41, 112, 0.1);
    background-color: rgba(107, 107, 107, 0.808);
}

@media (max-width: 1199px) {
    .sidebar {
        left: -300px;
    }
}

.sidebar::-webkit-scrollbar {
    width: 5px;
```

```
height: 8px;
background-color: #fff;
}

.sidebar::-webkit-scrollbar-thumb {
background-color: #aab7cf;
}

@media (min-width: 1200px) {

#main,
#footer {
margin-left: 300px;
}
}

@media (max-width: 1199px) {
.toggle-sidebar .sidebar {
left: 0;
}
}

@media (min-width: 1200px) {

.toggle-sidebar #main,
.toggle-sidebar #footer {
margin-left: 0;
}

.toggle-sidebar .sidebar {
left: -300px;
}
}

.sidebar-nav {
padding: 0;
margin: 0;
list-style: none;
}

.sidebar-nav li {
padding: 0;
margin: 0;
list-style: none;
}
```

```
.sidebar-nav .nav-item {  
    margin-bottom: 30px;  
}  
  
.sidebar-nav .nav-heading {  
    font-size: 11px;  
    text-transform: uppercase;  
    color: #343a40;  
    font-weight: 600;  
    margin: 10px 0 5px 15px;  
}  
  
.sidebar-nav .nav-link {  
    display: flex;  
    align-items: center;  
    font-size: 15px;  
    font-weight: 600;  
    color: #343a40;  
    transition: 0.3;  
    background: #ffffff;  
    padding: 10px 15px;  
    border-radius: 4px;  
}  
  
.sidebar-nav .nav-link i {  
    font-size: 16px;  
    margin-right: 10px;  
    color: #6a7fa5;  
}  
  
.sidebar-nav .nav-link.collapsed {  
    color: #000000;  
    background: #fff;  
}  
  
.sidebar-nav .nav-link.collapsed i {  
    color: #899bbd;  
}  
  
.sidebar-nav .nav-link:hover {  
    color: #ffffff;  
    background: #565656;  
}
```

```
.sidebar-nav .nav-link:hover i {
  color: #4781919f;
}

.sidebar-nav .nav-link .bi-chevron-down {
  margin-right: 0;
  transition: transform 0.2s ease-in-out;
}

.sidebar-nav .nav-link:not(.collapsed) .bi-chevron-down {
  transform: rotate(180deg);
}

.sidebar-nav .nav-content {
  padding: 5px 0 0 0;
  margin: 0;
  list-style: none;
}

.sidebar-nav .nav-content a {
  display: flex;
  align-items: center;
  font-size: 14px;
  font-weight: 600;
  color: #012970;
  transition: 0.3s;
  padding: 10px 0 10px 40px;
  transition: 0.3s;
}

.sidebar-nav .nav-content a i {
  font-size: 6px;
  margin-right: 8px;
  line-height: 0;
  border-radius: 50%;
}

.sidebar-nav .nav-content a:hover,
.sidebar-nav .nav-content a.active {
  color: #4154f1;
}

.sidebar-nav .nav-content a.active i {
  background-color: #4154f1;
```

```
}

/*
# Dashboard
*/
/* Filter dropdown */
.dashboard .filter {
  position: absolute;
  right: 0px;
  top: 15px;
}

.dashboard .filter .icon {
  color: #aab7cf;
  padding-right: 20px;
  padding-bottom: 5px;
  transition: 0.3s;
  font-size: 16px;
}

.dashboard .filter .icon:hover,
.dashboard .filter .icon:focus {
  color: #4154f1;
}

.dashboard .filter .dropdown-header {
  padding: 8px 15px;
}

.dashboard .filter .dropdown-header h6 {
  text-transform: uppercase;
  font-size: 14px;
  font-weight: 600;
  letter-spacing: 1px;
  color: #aab7cf;
  margin-bottom: 0;
  padding: 0;
}

.dashboard .filter .dropdown-item {
  padding: 8px 15px;
}

/* Info Cards */
```

```
.dashboard .info-card {  
  padding-bottom: 10px;  
}  
  
.dashboard .info-card h6 {  
  font-size: 28px;  
  color: #012970;  
  font-weight: 700;  
  margin: 0;  
  padding: 0;  
}  
  
.dashboard .card-icon {  
  font-size: 32px;  
  line-height: 0;  
  width: 64px;  
  height: 64px;  
  flex-shrink: 0;  
  flex-grow: 0;  
}  
  
.dashboard .sales-card .card-icon {  
  color: #282727;  
  background: #787878;  
}  
  
.dashboard .revenue-card .card-icon {  
  color: #282727;  
  background: #787878;  
}  
  
.dashboard .customers-card .card-icon {  
  color: #282727;  
  background: #787878;  
}  
  
/* Activity */  
.dashboard .activity {  
  font-size: 14px;  
}  
  
.dashboard .activity .activity-item .activite-label {  
  color: #888;  
  position: relative;  
  flex-shrink: 0;
```

```
flex-grow: 0;
min-width: 64px;
}

.dashboard .activity .activity-item .activite-label::before {
content: "";
position: absolute;
right: -11px;
width: 4px;
top: 0;
bottom: 0;
background-color: #eceeef;
}

.dashboard .activity .activity-item .activity-badge {
margin-top: 3px;
z-index: 1;
font-size: 11px;
line-height: 0;
border-radius: 50%;
flex-shrink: 0;
border: 3px solid #fff;
flex-grow: 0;
}

.dashboard .activity .activity-item .activity-content {
padding-left: 10px;
padding-bottom: 20px;
}

.dashboard .activity .activity-item:first-child .activite-label::before {
top: 5px;
}

.dashboard .activity .activity-item:last-child .activity-content {
padding-bottom: 0;
}

/* News & Updates */
.dashboard .news .post-item+.post-item {
margin-top: 15px;
}

.dashboard .news img {
width: 80px;
```

```
float: left;
border-radius: 5px;
}

.dashboard .news h4 {
font-size: 15px;
margin-left: 95px;
font-weight: bold;
margin-bottom: 5px;
}

.dashboard .news h4 a {
color: #012970;
transition: 0.3s;
}

.dashboard .news h4 a:hover {
color: #4154f1;
}

.dashboard .news p {
font-size: 14px;
color: #777777;
margin-left: 95px;
}

/* Recent Sales */
.dashboard .recent-sales {
font-size: 14px;
}

.dashboard .recent-sales .table thead {
background: #f6f6fe;
}

.dashboard .recent-sales .table thead th {
border: 0;
}

.dashboard .recent-sales .dataTable-top {
padding: 0 0 10px 0;
}

.dashboard .recent-sales .dataTable-bottom {
padding: 10px 0 0 0;
```

```
}
```

```
/* Top Selling */
```

```
.dashboard .top-selling {
```

```
    font-size: 14px;
```

```
}
```

```
.dashboard .top-selling .table thead {
```

```
    background: #f6f6fe;
```

```
}
```

```
.dashboard .top-selling .table thead th {
```

```
    border: 0;
```

```
}
```

```
.dashboard .top-selling .table tbody td {
```

```
    vertical-align: middle;
```

```
}
```

```
.dashboard .top-selling img {
```

```
    border-radius: 5px;
```

```
    max-width: 60px;
```

```
}
```

```
-----*
```

```
# Icons list page
```

```
-----*/
```

```
.iconslist {
```

```
    display: grid;
```

```
    max-width: 100%;
```

```
    grid-template-columns: repeat(auto-fit, minmax(180px, 1fr));
```

```
    gap: 1.25rem;
```

```
    padding-top: 15px;
```

```
}
```

```
.iconslist .icon {
```

```
    background-color: #fff;
```

```
    border-radius: 0.25rem;
```

```
    text-align: center;
```

```
    color: #012970;
```

```
    padding: 15px 0;
```

```
}
```

```
.iconslist i {
```

```
    margin: 0.25rem;
```

```
    font-size: 2.5rem;
}

.iconslist .label {
    font-family: var(--bs-font-monospace);
    display: inline-block;
    width: 100%;
    overflow: hidden;
    padding: 0.25rem;
    font-size: 12px;
    text-overflow: ellipsis;
    white-space: nowrap;
    color: #666;
}

/*
-----#
# Profie Page
-----*/
.profile .profile-card img {
    max-width: 120px;
}

.profile .profile-card h2 {
    font-size: 24px;
    font-weight: 700;
    color: #2c384e;
    margin: 10px 0 0 0;
}

.profile .profile-card h3 {
    font-size: 18px;
}

.profile .profile-card .social-links a {
    font-size: 20px;
    display: inline-block;
    color: rgba(1, 41, 112, 0.5);
    line-height: 0;
    margin-right: 10px;
    transition: 0.3s;
}

.profile .profile-card .social-links a:hover {
    color: #012970;
}
```

```
.profile .profile-overview .row {
  margin-bottom: 20px;
  font-size: 15px;
}

.profile .profile-overview .card-title {
  color: #012970;
}

.profile .profile-overview .label {
  font-weight: 600;
  color: rgba(1, 41, 112, 0.6);
}

.profile .profile-edit label {
  font-weight: 600;
  color: rgba(1, 41, 112, 0.6);
}

.profile .profile-edit img {
  max-width: 120px;
}

/*
-----#
# F.A.Q Page
-----*/
.faq .basic h6 {
  font-size: 18px;
  font-weight: 600;
  color: #4154f1;
}

.faq .basic p {
  color: #6980aa;
}

/*
-----#
# Contact
-----*/
.contact .info-box {
  padding: 28px 30px;
}

.contact .info-box i {
```

```
font-size: 38px;
line-height: 0;
color: #4154f1;
}

.contact .info-box h3 {
    font-size: 20px;
    color: #012970;
    font-weight: 700;
    margin: 20px 0 10px 0;
}

.contact .info-box p {
    padding: 0;
    line-height: 24px;
    font-size: 14px;
    margin-bottom: 0;
}

.contact .php-email-form .error-message {
    display: none;
    color: #fff;
    background: #ed3c0d;
    text-align: left;
    padding: 15px;
    margin-bottom: 24px;
    font-weight: 600;
}

.contact .php-email-form .sent-message {
    display: none;
    color: #fff;
    background: #18d26e;
    text-align: center;
    padding: 15px;
    margin-bottom: 24px;
    font-weight: 600;
}

.contact .php-email-form .loading {
    display: none;
    background: #fff;
    text-align: center;
    padding: 15px;
    margin-bottom: 24px;
```

```
}

.contact .php-email-form .loading:before {
    content: "";
    display: inline-block;
    border-radius: 50%;
    width: 24px;
    height: 24px;
    margin: 0 10px -6px 0;
    border: 3px solid #18d26e;
    border-top-color: #eee;
    -webkit-animation: animate-loading 1s linear infinite;
    animation: animate-loading 1s linear infinite;
}

.contact .php-email-form input,
.contact .php-email-form textarea {
    border-radius: 0;
    box-shadow: none;
    font-size: 14px;
    border-radius: 0;
}

.contact .php-email-form input:focus,
.contact .php-email-form textarea:focus {
    border-color: #4154f1;
}

.contact .php-email-form input {
    padding: 10px 15px;
}

.contact .php-email-form textarea {
    padding: 12px 15px;
}

.contact .php-email-form button[type=submit] {
    background: #4154f1;
    border: 0;
    padding: 10px 30px;
    color: #fff;
    transition: 0.4s;
    border-radius: 4px;
}
```

```
.contact .php-email-form button[type=submit]:hover {  
    background: #5969f3;  
}  
  
@-webkit-keyframes animate-loading {  
    0% {  
        transform: rotate(0deg);  
    }  
  
    100% {  
        transform: rotate(360deg);  
    }  
}  
  
@keyframes animate-loading {  
    0% {  
        transform: rotate(0deg);  
    }  
  
    100% {  
        transform: rotate(360deg);  
    }  
}  
  
/*-----  
# Error 404  
-----*/  
.error-404 {  
    padding: 30px;  
}  
  
.error-404 h1 {  
    font-size: 180px;  
    font-weight: 700;  
    color: #4154f1;  
    margin-bottom: 0;  
    line-height: 150px;  
}  
  
.error-404 h2 {  
    font-size: 24px;  
    font-weight: 700;  
    color: #012970;  
    margin-bottom: 30px;  
}
```

```
.error-404 .btn {  
background: #51678f;  
color: #fff;  
padding: 8px 30px;  
}  
  
.error-404 .btn:hover {  
background: #3e4f6f;  
}  
  
 @media (min-width: 992px) {  
 .error-404 img {  
 max-width: 50%;  
 }  
}  
  
/*-----  
# Footer -----*/  
.footer {  
padding: 20px 0;  
font-size: 14px;  
transition: all 0.3s;  
border-top: 1px solid #cddfff;  
}  
  
.footer .copyright {  
text-align: center;  
color: #012970;  
}  
  
.footer .credits {  
padding-top: 5px;  
text-align: center;  
font-size: 13px;  
color: #012970;  
}  
  
/*-----  
# add form -----*/  
.form-add-label{
```

```
    font-weight: bold;
}
.form-add-control{
    width:50%;
    outline: none;
}
#status{
    height: 30px;
}

body{
    background-color: #dee9ff;
}

.registration-form{
padding: 50px 0;
}

.registration-form form{
background-color: #fff;
max-width: 600px;
margin: auto;
padding: 50px 70px;
border-top-left-radius: 30px;
border-top-right-radius: 30px;
box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.075);
}

.registration-form .form-icon{
text-align: center;
background-color: #5891ff;
border-radius: 50%;
font-size: 40px;
color: white;
width: 100px;
height: 100px;
margin: auto;
margin-bottom: 50px;
line-height: 100px;
}

.registration-form .item{
border-radius: 20px;
margin-bottom: 25px;
padding: 10px 20px;
```

```
}
```

```
.registration-form .create-account{
```

```
    border-radius: 30px;
```

```
    padding: 10px 20px;
```

```
    font-size: 18px;
```

```
    font-weight: bold;
```

```
    background-color: #5791ff;
```

```
    border: none;
```

```
    color: white;
```

```
    margin-top: 20px;
```

```
}
```

```
.registration-form .social-media{
```

```
    max-width: 600px;
```

```
    background-color: #fff;
```

```
    margin: auto;
```

```
    padding: 35px 0;
```

```
    text-align: center;
```

```
    border-bottom-left-radius: 30px;
```

```
    border-bottom-right-radius: 30px;
```

```
    color: #9fadca;
```

```
    border-top: 1px solid #dee9ff;
```

```
    box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.075);
```

```
}
```

```
.registration-form .social-icons{
```

```
    margin-top: 30px;
```

```
    margin-bottom: 16px;
```

```
}
```

```
.registration-form .social-icons a{
```

```
    font-size: 23px;
```

```
    margin: 0 3px;
```

```
    color: #5691ff;
```

```
    border: 1px solid;
```

```
    border-radius: 50%;
```

```
    width: 45px;
```

```
    display: inline-block;
```

```
    height: 45px;
```

```
    text-align: center;
```

```
    background-color: #fff;
```

```
    line-height: 45px;
```

```
}
```

```

.registration-form .social-icons a:hover{
    text-decoration: none;
    opacity: 0.6;
}

@media (max-width: 576px) {
    .registration-form form{
        padding: 50px 20px;
    }

    .registration-form .form-icon{
        width: 70px;
        height: 70px;
        font-size: 30px;
        line-height: 70px;
    }
}

#image img{
    width: 40%;
    height: 50%;
    position: absolute;
    bottom: 50px;
}

```

Signup.css

```

body {
    font-family: Arial, Helvetica, sans-serif;
    margin: 0;
    background: #F6F6F2
}

.box-2{
    border-radius:10px;
    width:600px;
    height: 450px;
    margin-left: 600px;
    box-shadow: rgba(0, 0, 0, 0.3) 0px 19px 38px, rgba(0, 0, 0, 0.22) 0px 15px 12px ;
    position: absolute;
}

.register{
    padding: 10px 30px;
}

```

```
}

.name{
    margin-top: 10px;
    font-size: 2em;
    font-family: Arial Black, sans-serif;
    color:#388087;
    margin-left: 200px;
}

.user1 label{
    font-size: 20px;
    margin-left: 10px;
    font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
    color:rgb(76, 76, 76)
}

.pass1 label{
    font-size: 20px;
    margin-left: 10px;
    font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
    color:rgb(76, 76, 76)
}

.user1 input{
    width: 95% ;
    padding: 7px;
    margin: 10px 30px 15px 0px;
    border: none ;
    outline: none;
    background: #ecf0f3;
    height: 30px;
    font-size: 14px;
    border-radius: 50px;
}

.pass1 input {
    width: 95% ;
    padding: 10px;
    margin: 10px 30px 30px 0px;
    border: none ;
    outline: none;
    background: #ecf0f3;
    height: 30px;
    font-size: 16px;
    border-radius: 50px;
}

}
```

```
.btnlog {  
    padding: 15px 40px;  
    border: none;  
    outline: none;  
    border-radius: 50px;  
    font-size: 18px;  
    color: white;  
    background-color: rgb(0, 0, 0);  
    margin-left: 50px ;  
}  
.btnlog:hover{  
    color: rgb(168, 169, 171);  
}  
.ca {  
    font-size: 18px;  
    border: none;  
    outline:none;  
    background: transparent;  
    text-decoration: none;  
    color: rgb(18, 125, 212);  
    margin-left: 100px;  
    transition: 0.2s;  
}  
.ca:hover{  
    color: red;  
    font-size: 20px;  
    transition: 0.2s;  
}  
form h1 {  
    font-size: 30px;  
    color:#388087;  
    font-family:'Poppins', sans-serif;  
    position: relative;  
}  
.logs{  
    width: 40%;  
    height: auto;  
    margin-right: 700px;  
}
```

App.py

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

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;SECURITY=SSL;SSLSERVERCERTIFICATE=DigiCertGlobalRootCA.crt;UID=mfb71612;PWD=mFTq4HCfJpkcsNCD","","")

app=Flask(__name__)
app.secret_key ='asd'

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

@app.route('/dashboard')
def dashboard():
    sql = "SELECT COUNT(*) FROM shops"
    stmt = ibm_db.exec_immediate(conn, sql)
    shop = ibm_db.fetch_both(stmt)
    numshop=shop["1"]
    sql = "SELECT COUNT(*) FROM products"
    stmt = ibm_db.exec_immediate(conn, sql)
    prod = ibm_db.fetch_both(stmt)
    numprod=prod['1']
    sql = "SELECT COUNT(*) FROM location"
    stmt = ibm_db.exec_immediate(conn, sql)
    location = ibm_db.fetch_both(stmt)
    numlocation=location['1']

    if shop | prod | location:
        return render_template('dashboard.html', numshop = numshop,numprod = numprod ,numlocation = numlocation)
```

```

@app.route('/products',methods = ['POST', 'GET'])
def products():
    if request.method == 'POST':
        name = request.form['name']
        quantity = request.form['quantity']
        cost = request.form['cost']

        insert_sql = "INSERT INTO PRODUCTS VALUES (?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, name)
        ibm_db.bind_param(prep_stmt, 2, quantity)
        ibm_db.bind_param(prep_stmt, 3, cost)
        ibm_db.execute(prep_stmt)

    products = []
    sql = "SELECT * FROM PRODUCTS"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        products.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
    if products:
        return render_template("products.html", products = products)
    return render_template('products.html')

@app.route('/location',methods = ['POST', 'GET'])
def location():
    if request.method == 'POST':
        name = request.form['name']
        insert_sql = "INSERT INTO location VALUES (?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, name)
        ibm_db.execute(prep_stmt)

    location = []
    sql = "SELECT * FROM location"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        location.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
    if location:
        return render_template("location.html", location = location)

```

```

return render_template('location.html')

@app.route('/shops',methods = ['POST', 'GET'])
def shops():
    if request.method == 'POST':
        name = request.form['name']
        status = request.form['status']

        insert_sql = "INSERT INTO SHOPS VALUES (?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, name)
        ibm_db.bind_param(prep_stmt, 2, status)
        ibm_db.execute(prep_stmt)

    shops = []
    sql = "SELECT * FROM SHOPS"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        shops.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
    if shops:
        return render_template("shops.html", shops = shops)
    return render_template('shops.html')

@app.route('/delete/<name>')
def delete(name):
    sql = f"SELECT * FROM products WHERE name='{(name)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    products = ibm_db.fetch_row(stmt)
    if products:
        sql = f"DELETE FROM products WHERE name='{(name)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        products = []
        sql = "SELECT * FROM products"
        stmt = ibm_db.exec_immediate(conn, sql)
        dictionary = ibm_db.fetch_both(stmt)
        while dictionary != False:
            products.append(dictionary)
            dictionary = ibm_db.fetch_both(stmt)
        if products:
            return render_template("products.html", products = products)
    return render_template("products.html")

@app.route('/dele/<name>')

```

```

def dele(name):
    sql = f"SELECT * FROM shops WHERE name='{(name)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    store = ibm_db.fetch_row(stmt)
    if store:
        sql = f"DELETE FROM shops WHERE name='{(name)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        shops = []
        sql = "SELECT * FROM shops"
        stmt = ibm_db.exec_immediate(conn, sql)
        dictionary = ibm_db.fetch_both(stmt)
        while dictionary != False:
            shops.append(dictionary)
            dictionary = ibm_db.fetch_both(stmt)
        if shops:
            return render_template("shops.html", shops = shops)
    return render_template("shops.html")

@app.route('/dellocation/<name>')
def dellocation(name):
    sql = f"SELECT * FROM location WHERE name='{(name)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    location = ibm_db.fetch_row(stmt)
    if location:
        sql = f"DELETE FROM location WHERE name='{(name)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        location = []
        sql = "SELECT * FROM location"
        stmt = ibm_db.exec_immediate(conn, sql)
        dictionary = ibm_db.fetch_both(stmt)
        while dictionary != False:
            location.append(dictionary)
            dictionary = ibm_db.fetch_both(stmt)
        if location:
            return render_template("location.html", location = location)
    return render_template("location.html")

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

@app.route('/user/<id>')
def user_info(id):

```

```

with sql.connect('inventorymanagement.db') as con:
    con.row_factory=sql.Row
    cur =con.cursor()
    cur.execute(f'SELECT * FROM register WHERE email="{id}"')
    user = cur.fetchall()
    return render_template("user_info.html", user=user[0])

@app.route('/signin',methods=['GET', 'POST'])
def signin():
    global userid
    msg = ""
    if request.method == 'POST' :
        un = request.form['username']
        pd = request.form['password']
        sql = "SELECT * FROM register WHERE username =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,un)
        ibm_db.bind_param(stmt,2,pd)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            session['loggedin']= True
            session['id']= account['USERNAME']
            userid= account['USERNAME']
            session['username']= account['USERNAME']
            return render_template('dashboard.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('signin.html', msg = msg)

```

```

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

```

```

print(acnt)

if acnt:
    msg='Account already exists!!'

elif not re.match(r'^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]+', email):
    msg='Please enter the valid email address'
elif not re.match(r'[A-Za-z0-9]+', username):
    msg='name must contain only character and number'
else:
    insert_sql='INSERT INTO register VALUES (?, ?, ?)'
    pstmt=ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(pstmt, 1, username)
    ibm_db.bind_param(pstmt, 2, email)
    ibm_db.bind_param(pstmt, 3, pw)
    ibm_db.execute(pstmt)
    msg='You have successfully registered click signin!!'
    return render_template("signin.html")

elif request.method == 'POST':
    msg="fill out the form first!"
    return render_template("signup.html",msg=msg)

""" import sendgrid
from sendgrid.helpers.mail import *

sg = sendgrid.SendGridAPIClient(api_key='SENDGRID_API_KEY')
from_email = Email("81519104009@smartinternz.com")
to_email = To("815119104025@smartinternz.com")
subject = "Sending with SendGrid is Fun"
content = Content("text/plain", "and easy to do anywhere, even with Python")
mail = Mail(from_email, to_email, subject, content)
response = sg.client.mail.send.post(request_body=mail.get())
print(response.status_code)
print(response.body)
print(response.headers) """

if __name__ == '__main__':
    #app.run(debug=True)
    app.run(host='0.0.0.0')

```

GitHub & Project Demo Link

GitHub Link

<https://github.com/Logesh126/IBM-Project-29923-1660134407>

Project Demo Link

<https://youtu.be/tWbdT2CAZwI>