IBM NALAIYATHIRAN PROJECT REPORT

INVENTORY MANAGEMENT FOR RETAILERS

Date	19 NOVEMBER 2022
Team ID	PNT2022TMID02121
Project Name	Inventory Management for Retailers
Team Members	Alagu Gomathi Brindha A(2116190701014) Ancy Tera B(2116190701018) Dhananjeyan RM(2116190701038) Dinesh SP(2116190701044)

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

1.1 Project Overview :

Online Inventory Management System is software which is useful for the companies that operate local stores, where store owners keep the records of sales and buy. The problem with the manual system is, it slows the business. This venture disposes of the executive work, human issues, manual postponement and accelerated process. Online Inventory Management System will have the ability to customer detail, track sales and available inventory, tell a store owner when it is time to reorder and how much to buy. Inventory Management System may be a web based application developed for operating the systems which are focused within the area of Inventory control and generates the varied required reports. Inventory management system could be a web application for Windows that focuses on inventory and sales clearance. it absolutely was created for Windows operating systems. The inventory management system includes a number of features. This web application has logical tools for evaluating ideal inventory levels and selecting the acceptable replenishment strategies automatically. It also has capabilities just like the ability to spot stock levels, compute reorder points automatically, and highlight potential stock-outs. This system eliminates the chance of stock-outs of fast-moving goods by minimising delays.

1.2 Purpose :

The inventory management system is a software, methods, and technologies for managing and controlling inventories at a shopkeeper warehouse or shop. This software works on an admin system only which focuses on the needs and scale of the shop owner, as well as the capabilities and utility of the management software. Inventory management system software may be a necessary and valuable tool for all firms that affect inventory. It regulates the movement of stock in and out, keeps track of inventory levels for all items and stock, provides access to sales data and analytics, and helps businesses specify specific safety stock requirements. By keeping all the records in the system, admin can keep an eye on how much stock is in and how much stock is out so that they can order the inventory in a timely manner. This system provides an exact report of the month to an admin and monthly about stocks, sales and expenditure. At last, when the software is created and implemented

successfully then it would help businesses to increase their productivity

CHAPTER - 2
LITERATURE SURVEY

2.1 Existing problem :

In this digital world everyone wants to be digital in any field whether it is online shopping or online payment. So online inventory management is also one of the best things we want to do to cause inventory management by creating more problems. The aim of this project is to create a web application that will make it easier to manage Inventory Online items. Here Admin can add, remove, edit an item in his Inventory for any categories such as Medical item, electronic item, etc. They can generate sales invoices and much more. With the help of this the Internet-Based Assets Manager or Owner can easily see which item is in stock. Which can help owners to analyse their Business and work accordingly.

2.2 References :

[1] Joshni S Pasaribu, “Development of web based inventory information system” Vol-1, No. 2 (2021) pp 24-31, eISSN: 2775-2674.
[2] Trupti Shirsat, “Online inventory management system” Vol-2 No. 6 (2019), pp 118-119, ISSN: 2581-7175.
[3] Varalakshmi GS, “A review of inventory management system” Vol-10 No. 6 (2021), pp 421-423, ISSN: 2278-1021.
[4] Anas M. Atieh, “Performance improvement of inventory management system process by an automated warehouse management system”, (2016) pp 568-572, ISSN: 2212-8271.
[5] Balavishnu, “Stock Management System”, Vol-7 No. 2 (2021) pp 342-347, ISSN: 2456-3307.
[6] Duangpun Kritchanchai, “Developing Inventory Management in Hospital”, Vol-4 No. 2 pp 11-19(2015)

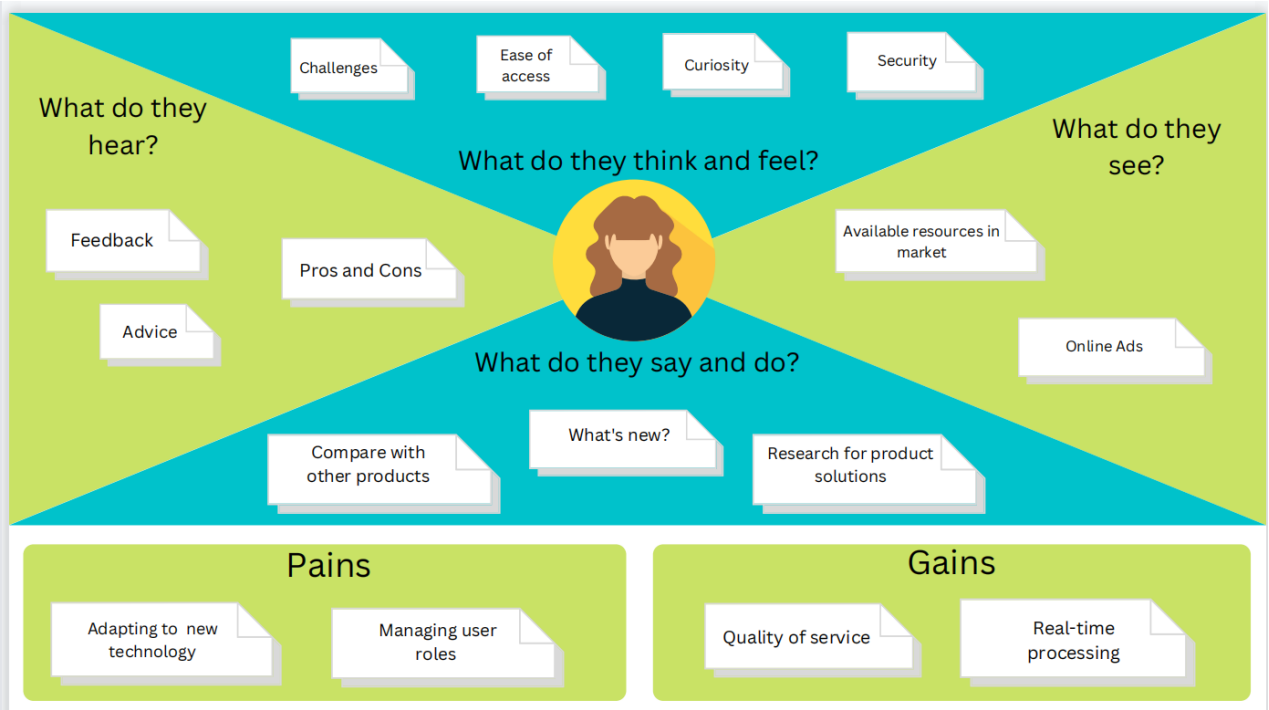
2.3 Problem statement definition :

In this digital world everyone wants to be digital in any field whether it is online shopping or online payment. So online inventory management is also one of the best things we want to do to cause inventory management by creating more problems. The aim of this project is to create a web application that will make it easier to manage Inventory Online items. Here Admin can add, remove, edit an item in his Inventory for any categories such as Medical item, electronic item, etc. They can generate sales invoices and much more. With the help of this the Internet-Based Assets Manager or Owner can easily see which item is in stock. Which can help owners to analyse their Business and work accordingly.

CHAPTER - 3

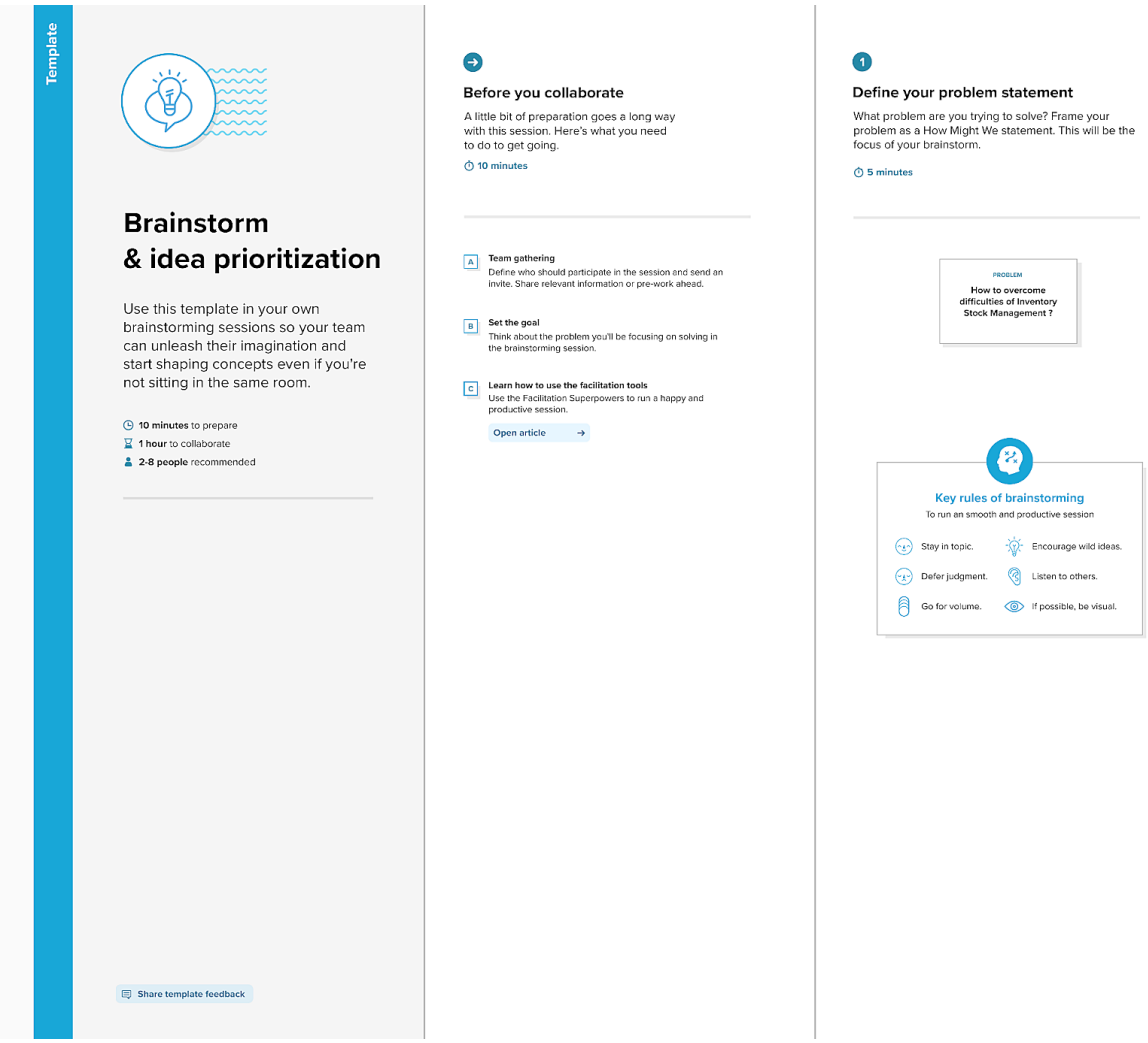
IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas :



3.2 Ideation and Brainstorming :

Team Gathering, Collaboration, Problem Statement Selection :



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

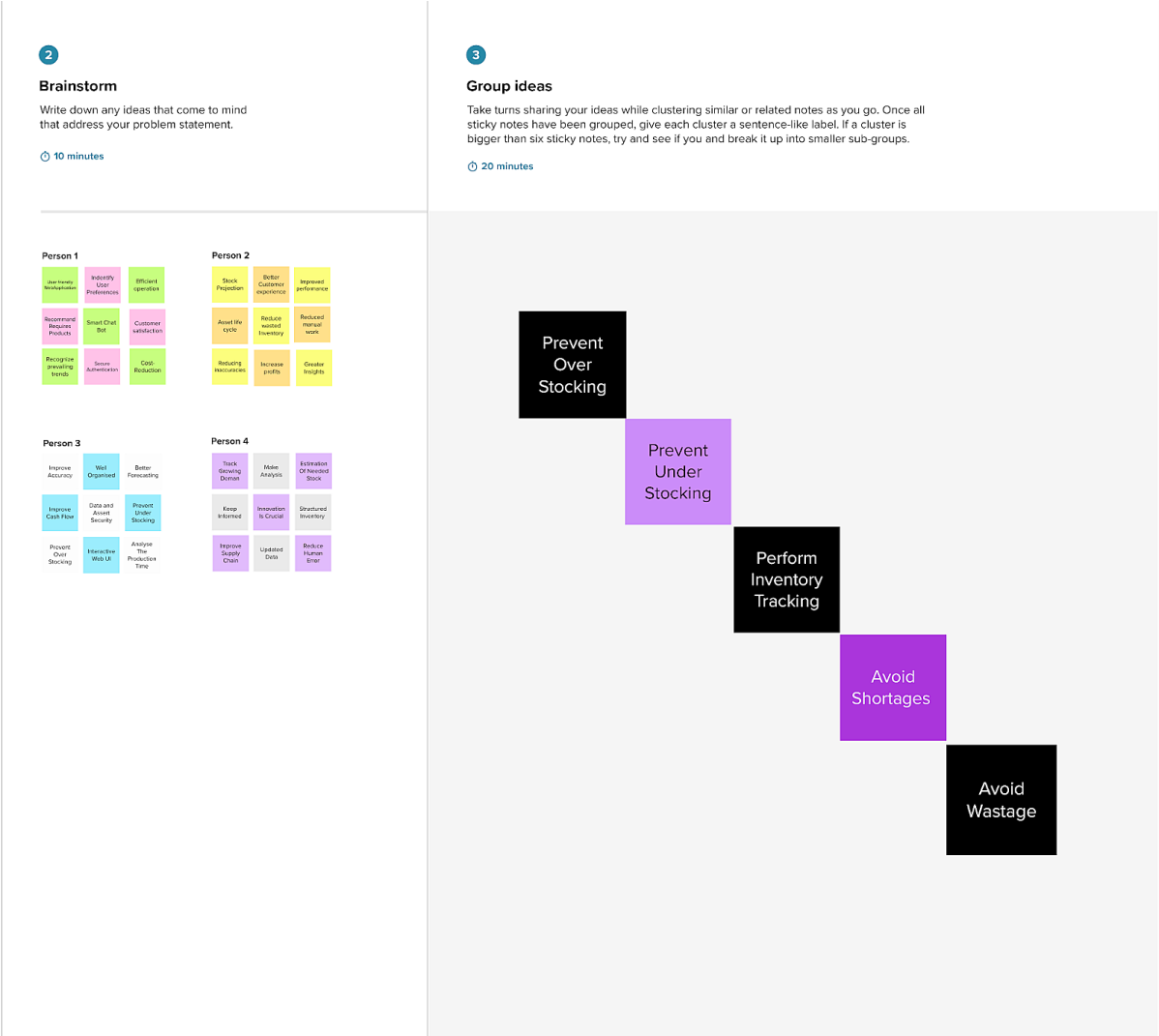
5 minutes

PROBLEM
How to overcome difficulties of Inventory Stock Management ?

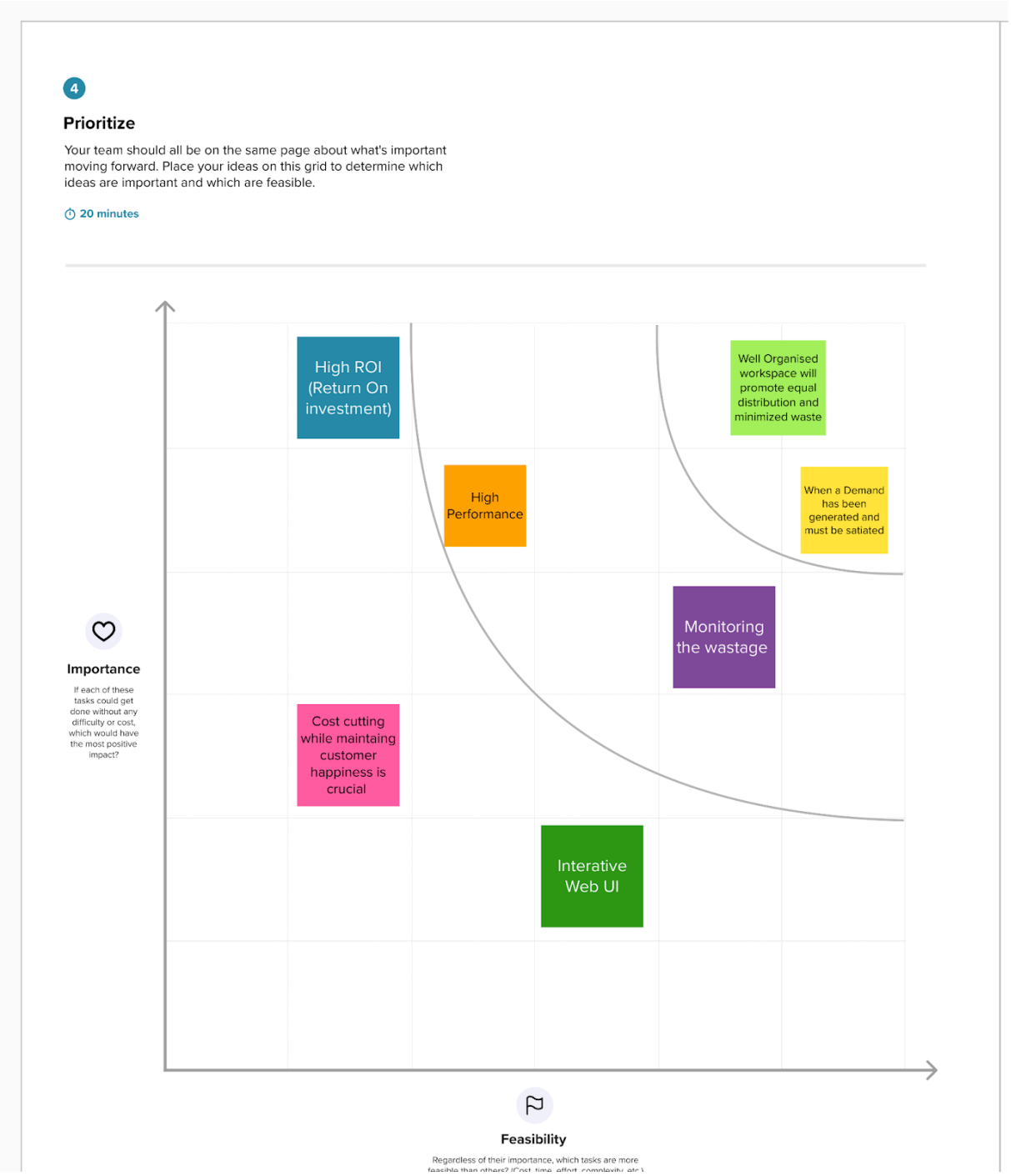
Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Brainstorm, Idea listing and Grouping :



Idea Prioritization :



3.3 Proposed Solution :

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Retailers want to carry out merchandise with sufficient stock in hand which is neither too little nor too much, so an end-to-end web application can be created which is capable of displaying the current amount of stock present in the warehouse, which helps inventory managers to keep track of products all the time.
2.	Idea / Solution description	To develop an end-to-end web application which in default shows the amount of stock present in the inventory at that time. Users can add or reduce the number of goods based on purchase and sales.
3.	Novelty / Uniqueness	Though we have a lot of inventory management applications, this one is unique with a feature that when any stock count gets reduced to minimum, an alert email with stock details will be automatically generated for the retailer and the appropriate seller from whom the retailer purchases goods. This helps in saving time.
4.	Social Impact / Customer Satisfaction	Retailers will be benefited as effective retail management results in lower costs and better understanding of sales patterns which give them more information with which to run the business.
5.	Business Model (Revenue Model)	We can provide the application for retailers on a subscription basis with which revenue can be generated.
6.	Scalability of the Solution	Inventory data can be scaled up and scaled down based on the number of available inventory in the warehouse.

3.4 Problem Solution Fit :

Problem-Solution fit

INVENTORY MANAGEMENT FOR RETAILERS

Team Id : PNT2022TMID02121

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?

The main customer of our project is retailers who want to carry out merchandise with sufficient stock in hand which is neither too little nor too much.

6. CUSTOMER CONSTRAINTS — CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

- Network connection
- Inadequate product knowledge
- Time consuming
- Bulk data transfer

5. AVAILABLE SOLUTIONS — AS

Which solutions are available to the customers when they face the problem or need to get the job done?

- Manual tracking of stock count
- Local data storage solutions
- Traditional software systems with conventional database architecture

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

- Create a platform to track the amount of stock present in inventory at any instant
- To add or reduce the number of goods based on purchase and sales
- Making inventory management simpler
- To refill the stock from convenient seller

9. PROBLEM ROOT CAUSE — RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

- Getting accurate stock details
- Changing customer demand
- Purchasing too much / too less of the wrong inventory
- Slow pickup, packaging and shipping of customer orders

7. BEHAVIOUR — BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

- Track the flow of products
- Calculate costs and benefits
- Learn about usage
- Frequently update database

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS — TR

What triggers customers to act?

- Increasing customer demand
- Market competition

4. EMOTIONS: BEFORE / AFTER — EM

How do customers feel when they face a problem and afterwards?

Before	After
More Stress Excessive Manual work Time for stock calculation	Less Stress Relatively less Manual work Counting is not necessary

10. YOUR SOLUTION — SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.

To develop an end-to-end web application which in default shows the amount of stock present in the inventory at that time. User can add or reduce goods based on purchase and sales

8. CHANNELS of BEHAVIOUR — CH

1. ONLINE

What kind of actions do customers take online? Extract online channels from #7

- Intimating the concerned person when specific stock count reaches minimum
- Constant checking and updating of stock quantity

2. OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

- Manual checking
- Distribution of inventory

Extract online & offline CH of BE

CHAPTER - 4

REQUIREMENT ANALYSIS

4.1 Functional Requirement :

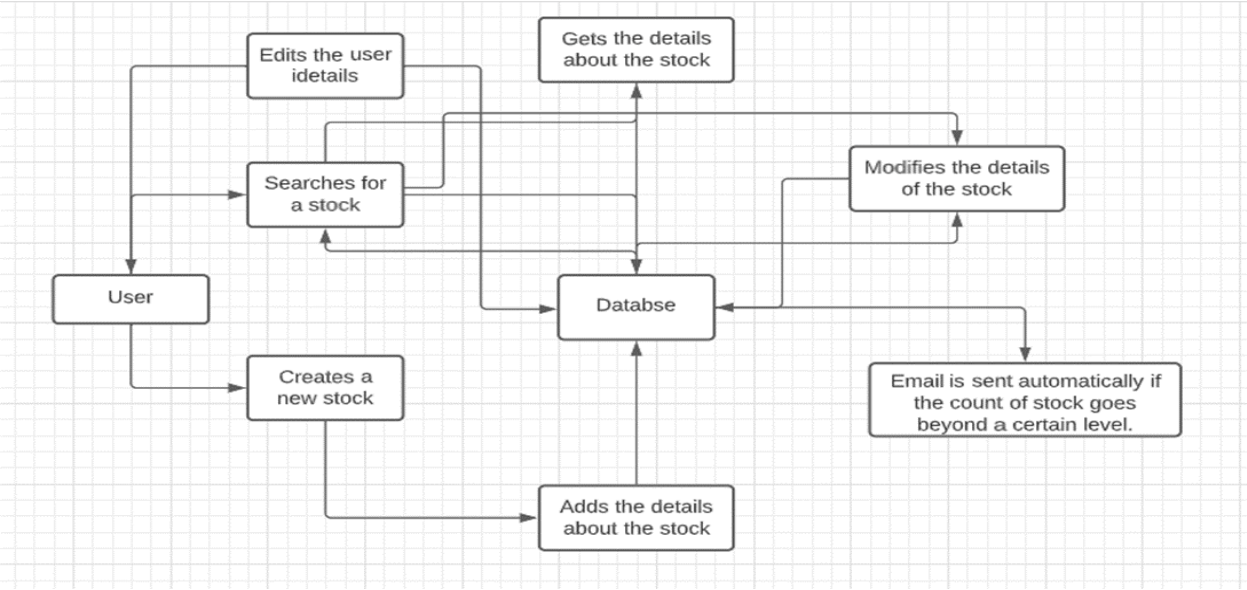
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login with username Login with password
FR-4	Product record	Product name Stock count Product category Vendor details
FR-5	Email Notification	Email through SendGrid Reduced stock quantity Email to both retailer and seller
FR-6	Audit Monitoring	Monitor incoming and outgoing stock

4.2 Non-Functional Requirement :

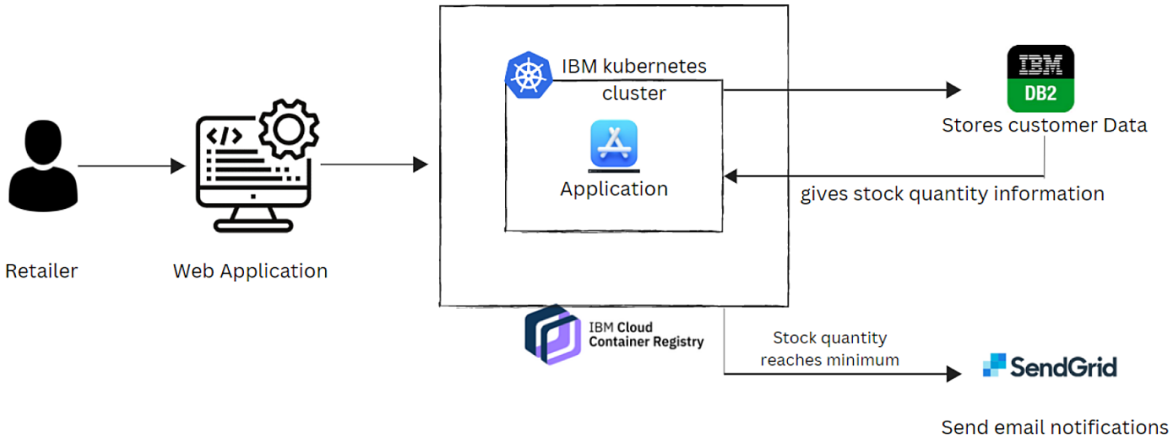
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Highly portable, User-friendly and highly responsive UI for easy access
NFR-2	Security	Access Control, User privileges, Password management features
NFR-3	Reliability	Secure server for reliable and fault tolerant connection
NFR-4	Performance	Reliable performance with high-end servers
NFR-5	Availability	Service hosting server downtime should be negligible during upgradation
NFR-6	Scalability	The resources and service provided by the software should be scalable

CHAPTER - 5
PROJECT DESIGN

5.1 Data Flow Diagrams :



5.2 Solution and Technology Architecture :



5.3 User Stories :

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 and password and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application	I can receive a confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can Sign Up	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can Sign In	High	Sprint-1
	Dashboard	USN-6	As a user, I can log into my account and access the Dashboard	I can access the Dashboard	High	Sprint-1
Customer Care Executive	Customer Support	USN-7	As a user, I can handle customer queries	Clear customer complaints	High	Sprint-2
Administrator	Management	USN-8	As an administrator, I can specify access, privileges, roles and responsibilities of new and existing users.	Administer all activities	High	Sprint-2

CHAPTER - 6

PROJECT PLANNING AND SCHEDUING

6.1 Sprint Planning and Estimation :

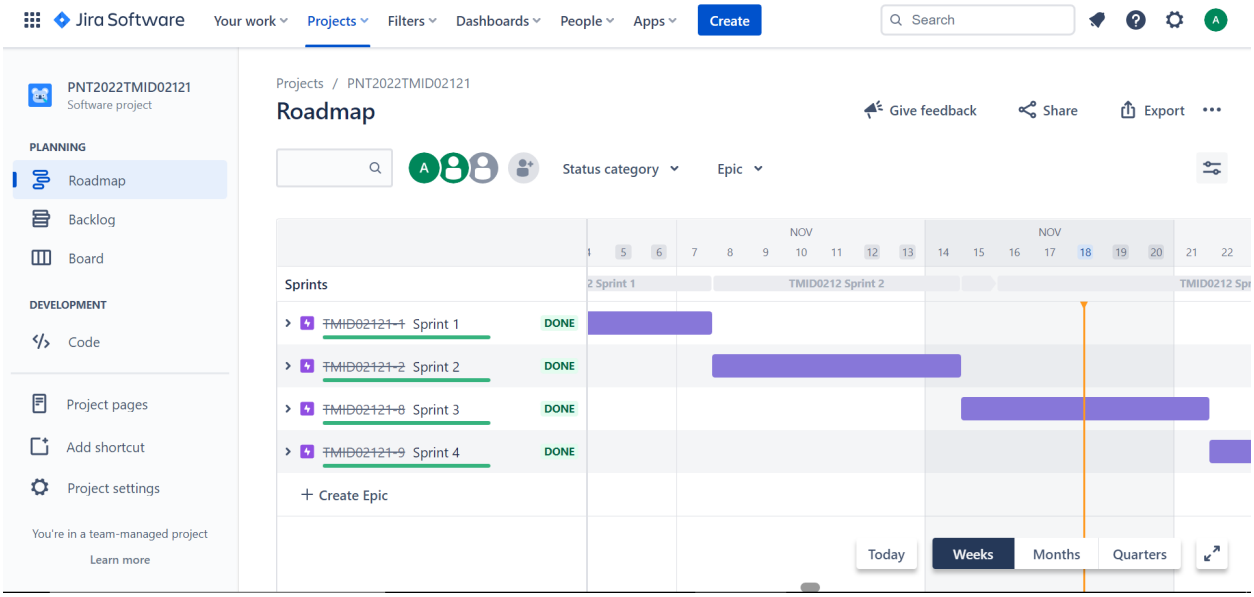
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High
Sprint-1	Registration	USN-2	As a user, I will receive a confirmation email once I have registered for the application	1	High
Sprint-2	Registration	USN-3	As a user, I can register for the application through Facebook	2	Low
Sprint-1	Registration	USN-4	As a user, I can register for the application through Gmail	2	Medium
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High
Sprint-1	Dashboard	USN-6	As a user, I must be able to see my details on the dashboard	3	High
Sprint-1	Dashboard	USN-7	As a user, I should be able to change my account settings whenever I prefer.	2	Medium
Sprint-2	Inventory	USN-8	As a retailer, I should be able to alter product details	2	Medium
Sprint-2	Inventory	USN-9	As a retailer, I should be able to add or reduce the number of product	3	Medium
Sprint-3	Inventory	USN-10	As a retailer, I should be able to get alert or notification on shortage of stock via email	5	High

Sprint-3	Communication	USN-11	As a user, I should be able to get the needed details with the help of a chat bot	1	Low
Sprint-4	Maintenance	USN-12	As an admin, I should be able to access control	3	High

6.2 Sprint Delivery Schedule :

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	21 Nov 2022

6.3 Reports from JIRA :



CHAPTER - 7

CODING AND SOLUTIONING

7.1 Adding Products to Inventory :

```
@app.route('/addproduct', methods=["GET","POST"])
def addproduct():
    if request.method=="POST":
        name=request.form['productName']
        stock=request.form['qty']
        expiryDate=request.form['expiryDate']
        wholesalerName=request.form['wholesalerName']
        wholesalerNumber=request.form['wholesalerNumber']
        costPrice=request.form['costPrice']
        sellingPrice=request.form['sellingPrice']
        Retailer_email=request.form['Retailer_email']
        stmt=ibm_db.prepare(conn,"Insert into
ProductsInformation(Name,stock,wholesalername,wholesalernumber,costprice,selling
price,retaileremail) values(?,?,?,?,?,?,?)")
        ibm_db.bind_param(stmt,1,name)
        ibm_db.bind_param(stmt,2,stock)
        ibm_db.bind_param(stmt,3,wholesalerName)
        ibm_db.bind_param(stmt,4,wholesalerNumber)
        ibm_db.bind_param(stmt,5,costPrice)
        ibm_db.bind_param(stmt,6,sellingPrice)
        ibm_db.bind_param(stmt,7,Retailer_email)
        ibm_db.execute(stmt)
        return redirect(url_for("home"))
    else:
        return render_template("AddProduct.html")
```

7.2 Editing product details :

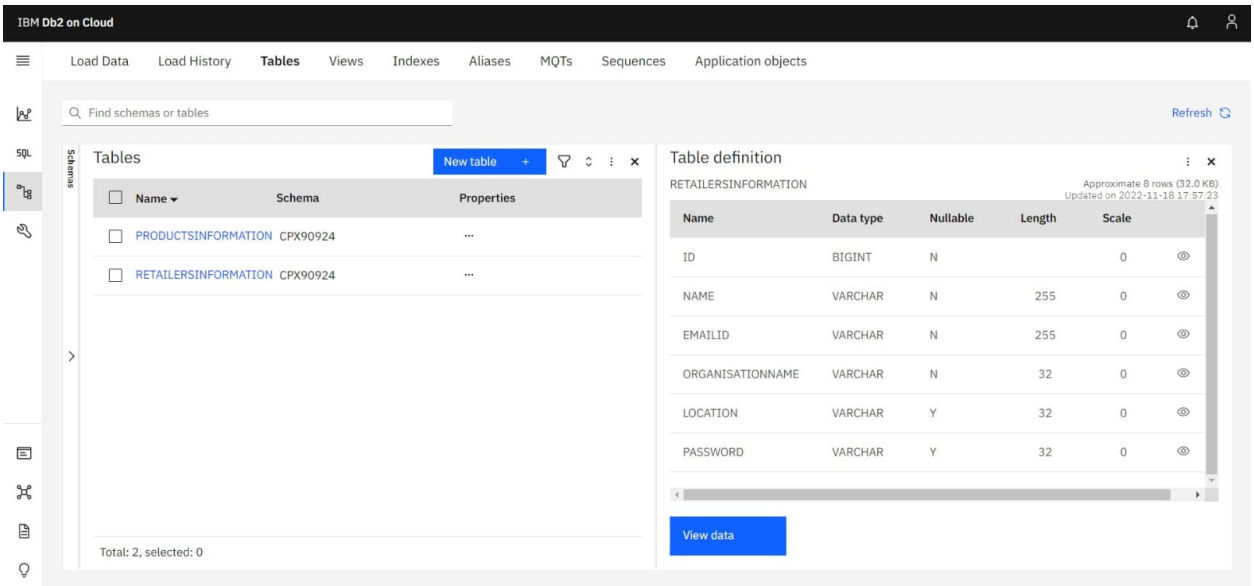
```
@app.route('/editproduct', methods=["POST"])
```

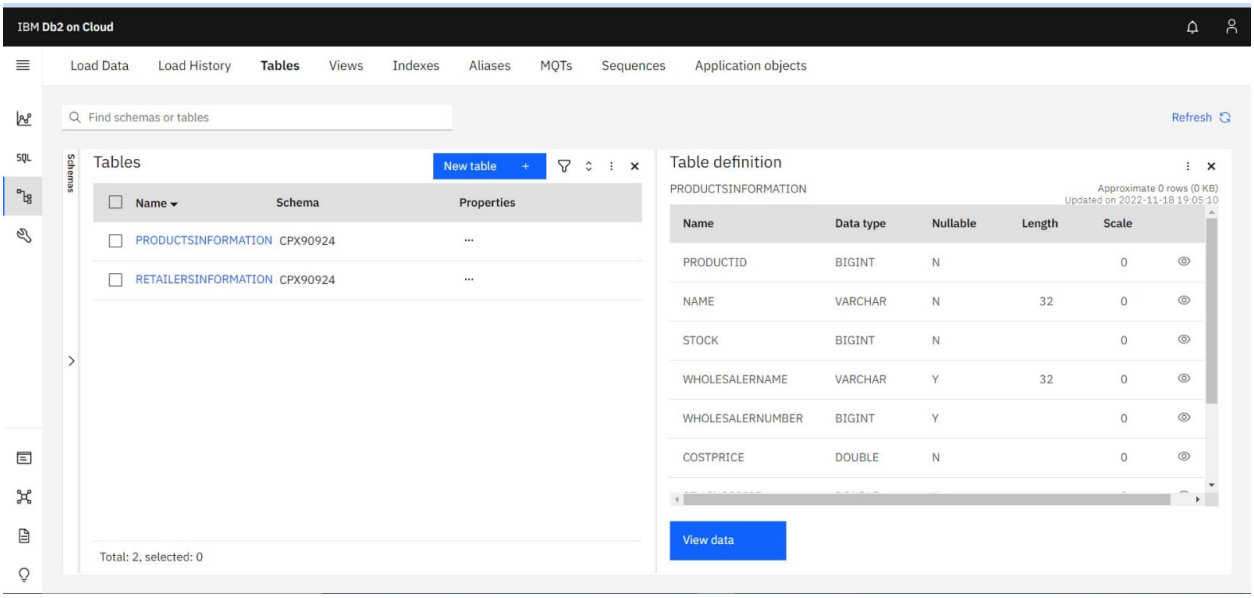
```
def editproduct():
    pid=request.args.get("id")
    name=request.form['productName']
    stock=request.form['qty']
    expiryDate=request.form['expiryDate']
    wholesalerName=request.form['wholesalerName']
    wholesalerNumber=request.form['wholesalerNumber']
    costPrice=request.form['costPrice']
    sellingPrice=request.form['sellingPrice']
    Retailer_email=request.form['Retailer_email']
    sql=f"update ProductsInformation set
productname='{name}',stock='{stock}',wholesalerName='{wholesalerName}',wholesal
erNumber='{wholesalerNumber}',costPrice='{costPrice}',sellingPrice='{sellingPrice}',R
etailer_email='{Retailer_email}'  where productid='{pid}'")
    stmt=ibm_db.exec-immediate(conn,sql)
    return redirect(url_for("home"))
```

7.3 Deleting Products from Inventory :

```
@app.route('/deleteproduct', methods=["POST"])
def deleteproduct():
    pid=request.args.get('productid')
    sql=f"Delete from ProductsInformation where productid='{pid}'"
    stmt=ibm_db.exec_immediate(conn,sql)
    return redirect(url_for("home"))
```

7.4 Database Schema :





CHAPTER - 8

TESTING

8.1 Test Cases :

It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectation and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement

Test Case ID	Feature Type	Test Scenario	Expected Result	Actual Result	Status	TC for Automation (Y/N)	Bug ID
LoginPage_TC_001	UI	Verify whether admin is able to Login/Signup to the application	Login/Signup should be successful by sending verification email	Working as expected	Pass	Y	
EditProduct_TC_001	Functional	Verify whether user is able to edit product details	Product details can be edited and reflected in database	Showing output "Key Not Found Error"	Fail	N	bug0123
EditProduct_TC_002	Functional	Verify whether user is able to edit	Product details can be edited and reflected in database	Working as expected	Pass	Y	

		product details					

8.2 User Acceptance Testing :

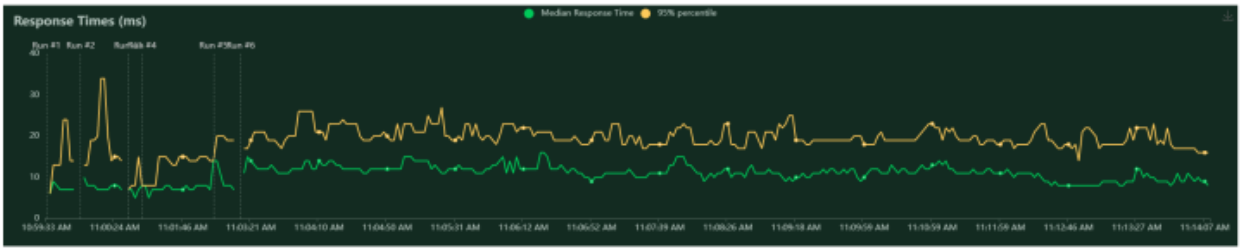
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Sub total
By Design	8	4	2	3	17
Duplicate	1	0	2	1	4
External	2	3	0	1	6
Fized	10	2	5	18	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	2	1	6
Totals	21	12	13	25	71

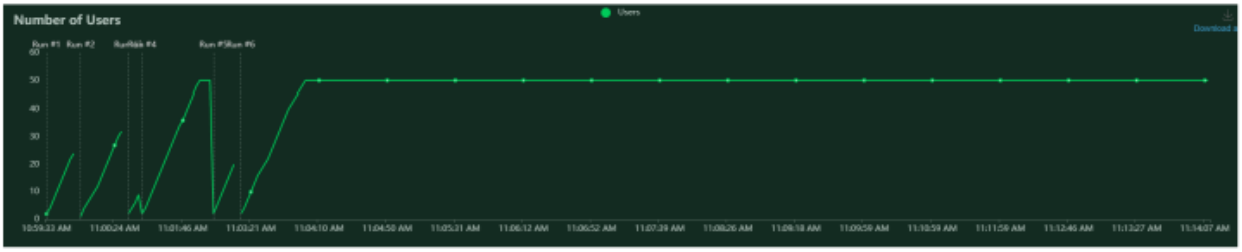
LOCUST

HOST http://127.0.0.1:5000 | STATUS RUNNING 50 users Edit | RPS 5.4 | FAILURES 0% | STOP | Reset Stats

Statistics | Charts | Failures | Exceptions | Current ratio | Download Data

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	354	0	16	19	24	15	5	27	11	1.9	0
GET	/home	329	0	10	13	26	10	4	33	11	1.4	0
GET	/request	352	0	15	20	27	15	5	31	11	2.1	0
	Aggregated	**1035**	**0**	**12**	**19**	**26**	**13**	**4**	**33**	**11**	**5.4**	**0**



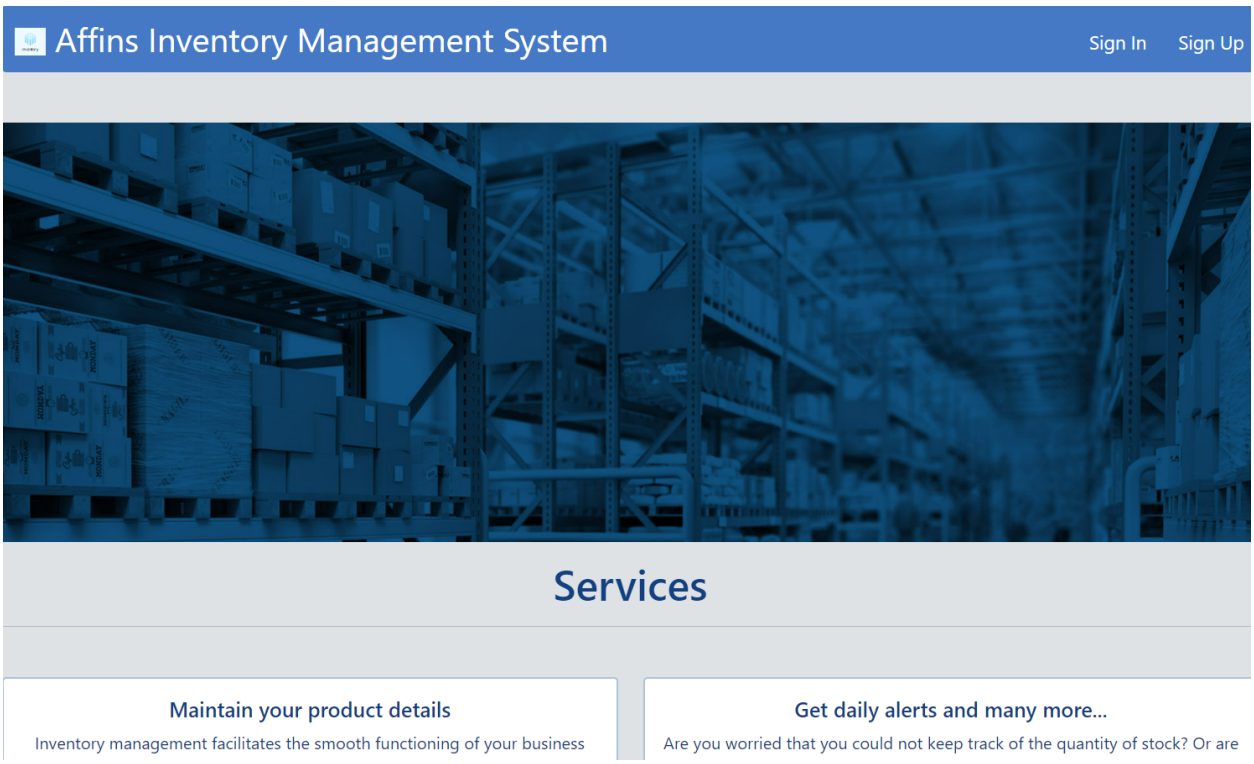


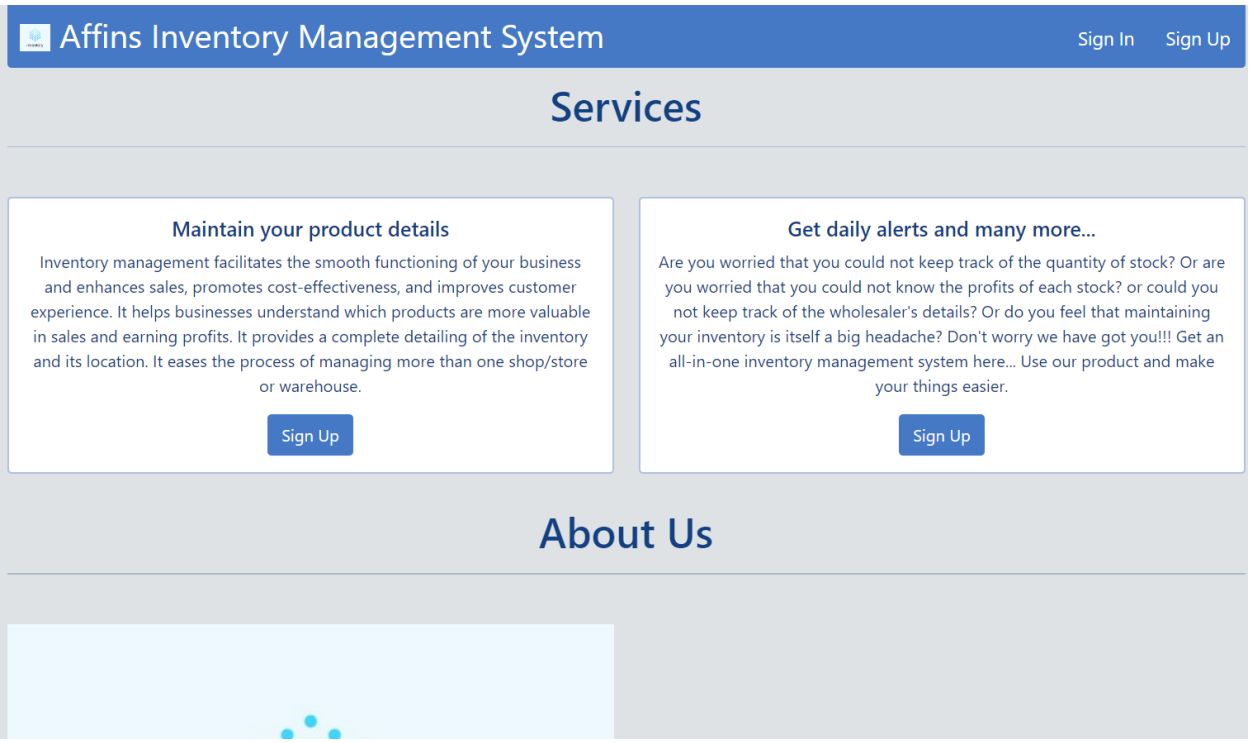
CHAPTER - 9

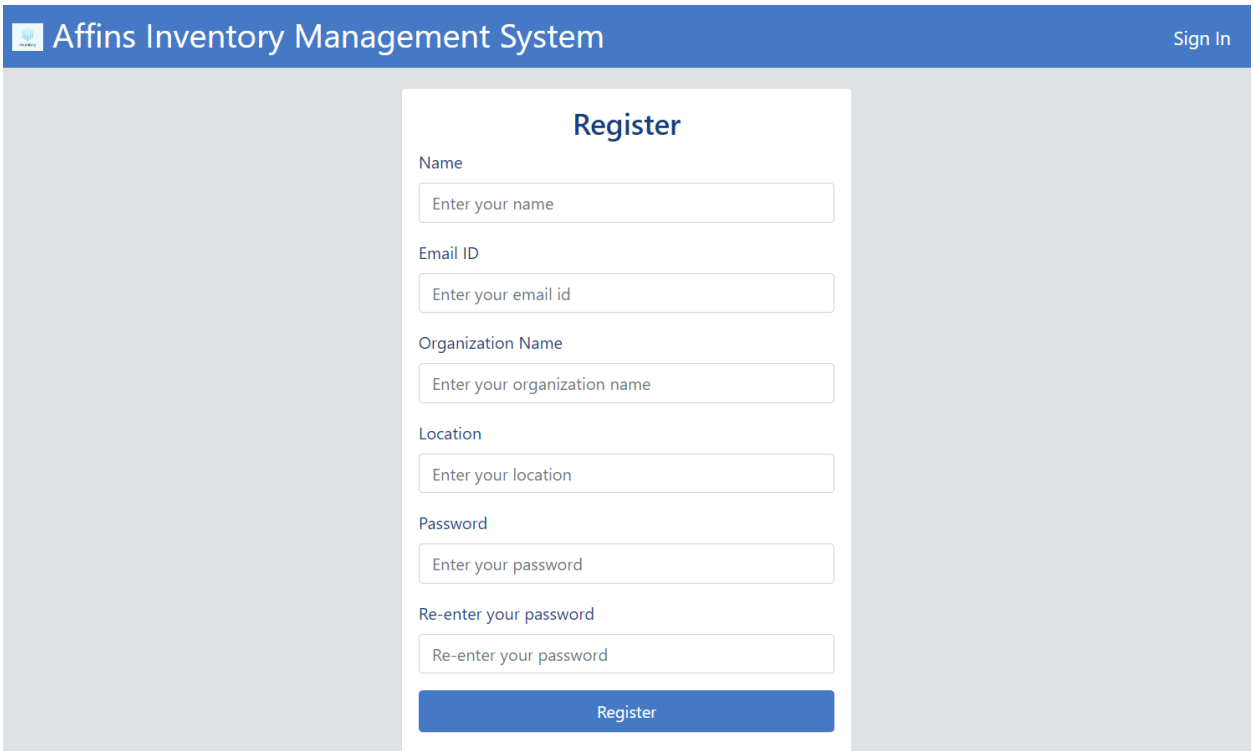
RESULTS

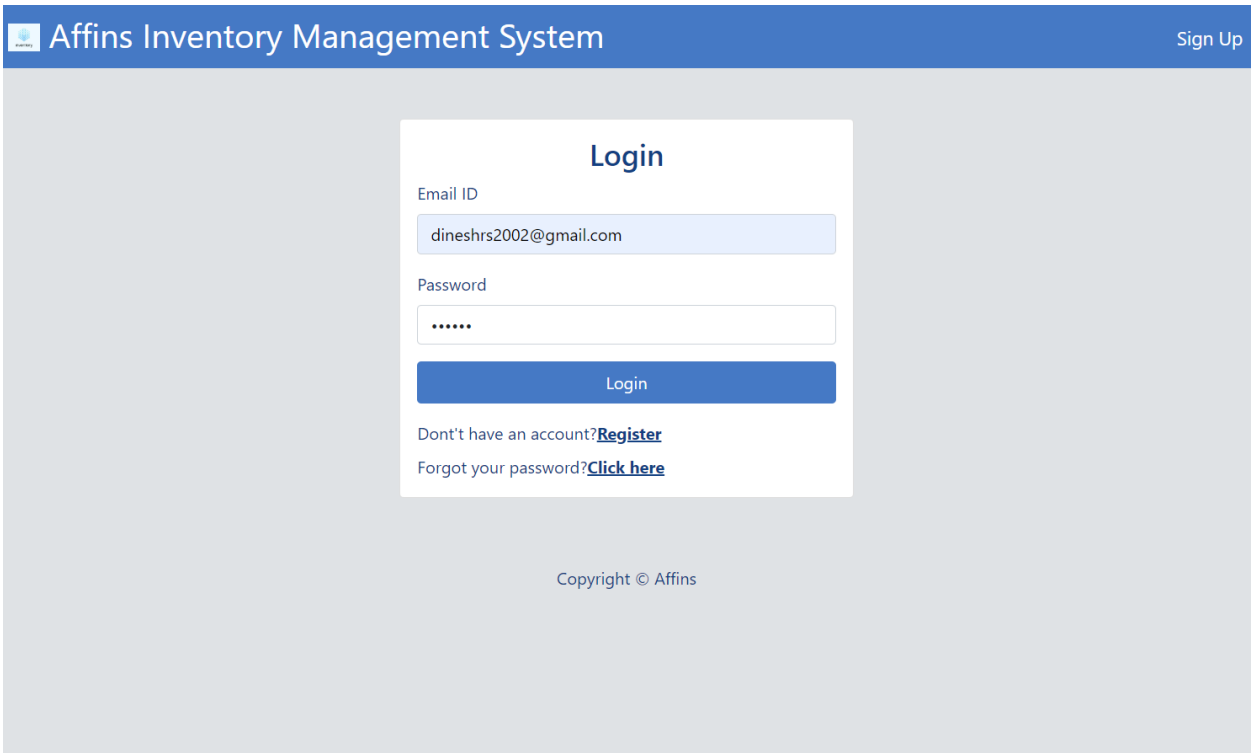
9.1 Performance Metrics :

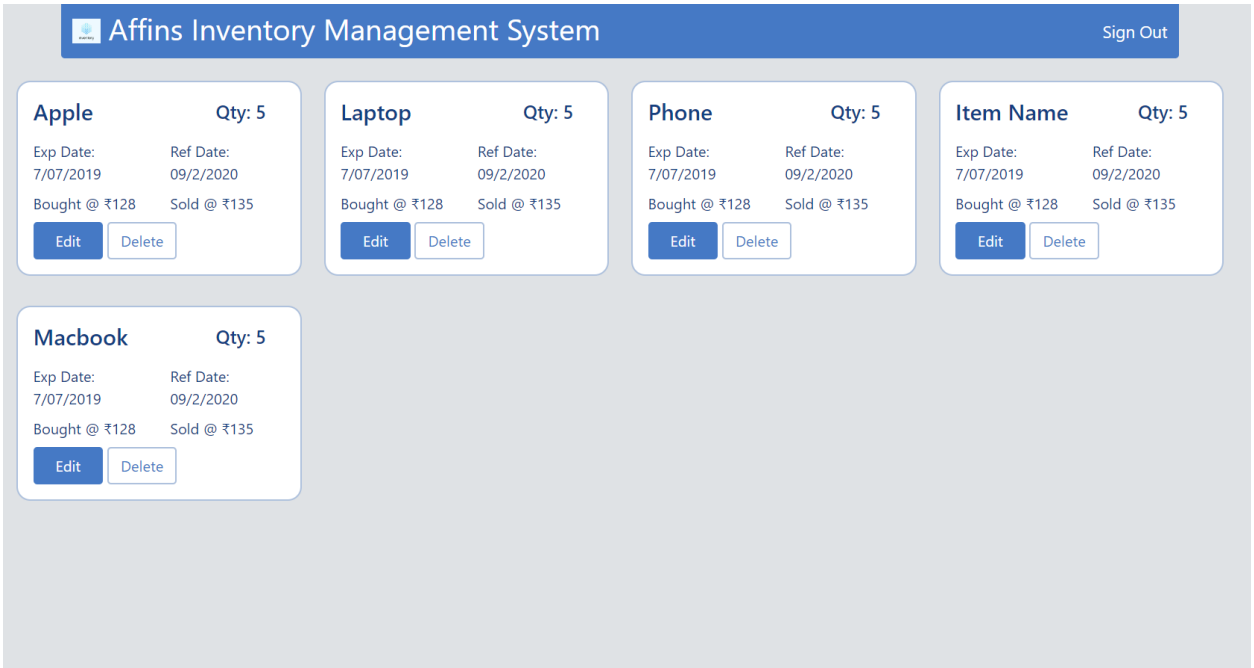
- Project metrics are used to track the progress and performance of a project.
- Monitoring parts of a project like productivity, scheduling, and scope make it easier for team leaders to see what's on track.
- As a project evolves, managers need access to changing
- deadlines or budgets to meet their client's expectations

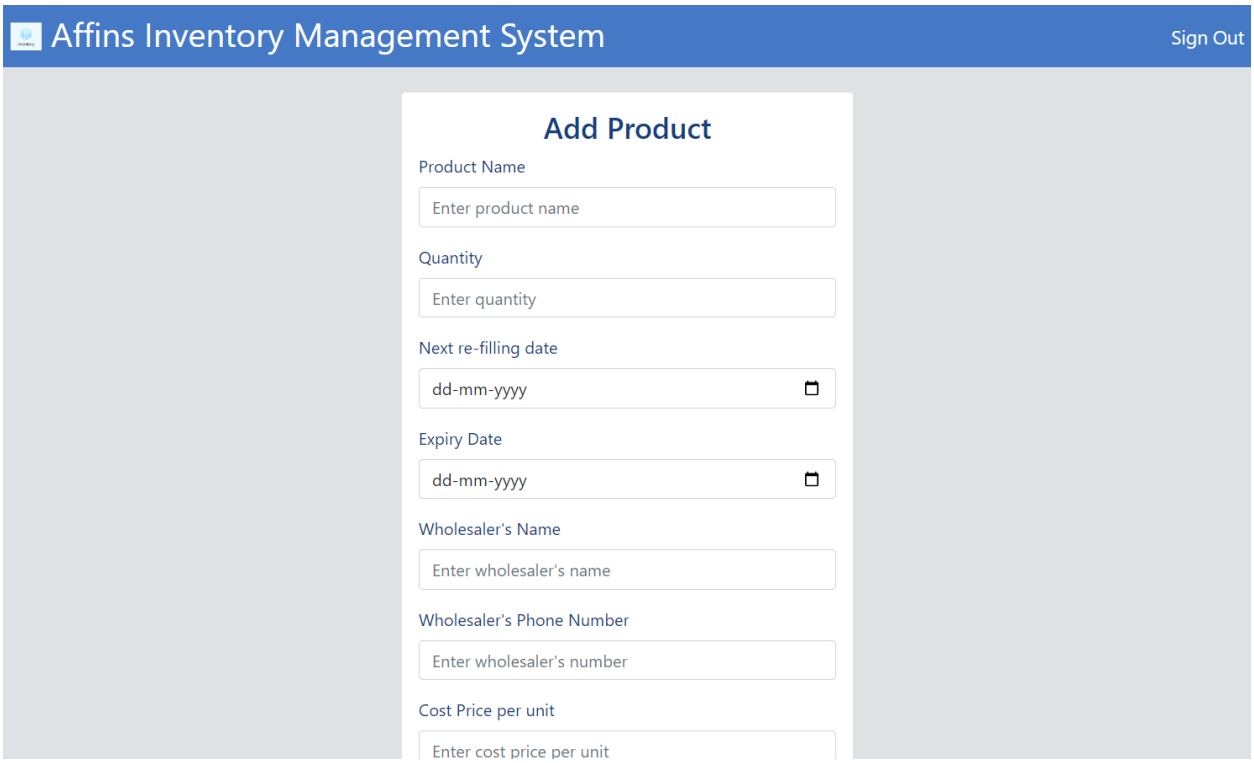



Affins Inventory Management System
Sign In
Sign Up
Services
Maintain your product details
Inventory management facilitates the smooth functioning of your business and enhances sales, promotes cost-effectiveness, and improves customer experience. It helps businesses understand which products are more valuable in sales and earning profits. It provides a complete detailing of the inventory and its location. It eases the process of managing more than one shop/store or warehouse.
Sign Up
Get daily alerts and many more...
Are you worried that you could not keep track of the quantity of stock? Or are you worried that you could not know the profits of each stock? or could you not keep track of the wholesaler's details? Or do you feel that maintaining your inventory is itself a big headache? Don't worry we have got you!!! Get an all-in-one inventory management system here... Use our product and make your things easier.
Sign Up
About Us


Affins Inventory Management System
Sign In
Register
Name
Enter your name
Email ID
Enter your email id
Organization Name
Enter your organization name
Location
Enter your location
Password
Enter your password
Re-enter your password
Re-enter your password
Register


Affins Inventory Management System
Sign Up
Login
Email ID
dineshrs2002@gmail.com
Password
Login
Dont't have an account?Register
Forgot your password?Click here
Copyright © Affins


Affins Inventory Management System
Sign Out
Apple
Qty: 5
Exp Date: 7/07/2019
Ref Date: 09/2/2020
Bought @ ₹128
Sold @ ₹135
Edit
Delete
Laptop
Qty: 5
Exp Date: 7/07/2019
Ref Date: 09/2/2020
Bought @ ₹128
Sold @ ₹135
Edit
Delete
Phone
Qty: 5
Exp Date: 7/07/2019
Ref Date: 09/2/2020
Bought @ ₹128
Sold @ ₹135
Edit
Delete
Item Name
Qty: 5
Exp Date: 7/07/2019
Ref Date: 09/2/2020
Bought @ ₹128
Sold @ ₹135
Edit
Delete
Macbook
Qty: 5
Exp Date: 7/07/2019
Ref Date: 09/2/2020
Bought @ ₹128
Sold @ ₹135
Edit
Delete


Affins Inventory Management System
Sign Out
Add Product
Product Name
Enter product name
Quantity
Enter quantity
Next re-filling date
dd-mm-yyyy
Expiry Date
dd-mm-yyyy
Wholesaler's Name
Enter wholesaler's name
Wholesaler's Phone Number
Enter wholesaler's number
Cost Price per unit
Enter cost price per unit

CHAPTER - 10

ADVANTAGES AND DISADVANTAGES

Advantages:

- It helps to maintain the right amount of stocks.
- It leads to a more organized warehouse.
- It saves time and money, an effective inventory management system can translate to time and money saved on the part of the business.
- Reduction in holding costs yet another benefit of an efficient management system is that it helps to save on inventory cost.
- A well-structured inventory management system leads to improved customer retention for customers to keep patronizing you, you will need to always have the goods they want, at the amount they want, and at the time they want it.

Disadvantages:

- Bureaucracy, even though inventory management allows employees at every level of the company to read and manipulate company stock and product inventory, the infrastructure required to build such a system adds a layer of bureaucracy to the whole process and the business in general.
- Impersonal touch, another disadvantage of inventory management is a lack of personal touch.
- Even though inventory management can reveal to we the amount of stock we have at hand and the amount that we have sold off, it can also hide production problems that could lead to customer service disasters.
- Increased space is need to hold the inventory, in order to hold inventory, you will need to have space so unless the goods you deal in are really small in size, then you will need a warehouse to store it.

CHAPTER - 11

CONCLUSION

To conclude, this inventory management system plays a vital role in keeping data that stores sales data for a specific desktop application. It is a simple desktop application that links to the particular distribution centre, allowing information to be refreshed and confirmed within the store. It also provides sales information on a daily, weekly, and monthly basis. This method makes inventory management a breeze. Increased income, profitability, and an overall boost in customer satisfaction are noticed as a result of the inventory management system.

CHAPTER - 12

FUTURE SCOPE

All retailers may not be able to employ these technologies due to their high cost of implementation and maintenance. To all those retailers with limited resources, cheaper software is accessible that could help with the management of their inventory like bar codes or policies as EOQ, AUD, and IQD, which will allow optimizing their stock without making considerable investment

CHAPTER - 13

APPENDIX

Source code:

```
from flask import Flask, render_template, request,redirect, url_for,escape
from flask_session import Session
import sendgrid
from sendgrid.helpers.mail import Mail, Email, To, Content
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=2d46b6b4-cbf6-40eb-bbce-
6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32328;SE
CURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=cpx90924;PWD=
Wq27hQ1Veq7bFkKx","","")
app=Flask(_name_)
@app.route('/')
def index():
    return render_template('Homepage.html')
@app.route('/signup',methods=['GET','POST'])
def register():
    if request.method=='POST':
         name=request.form['name']
        email=request.form['email']
        org=request.form['organization']
        location=request.form['location']
        pwd=request.form['password']
        stmt=ibm_db.prepare(conn,"Insert into
RetailersInformation(Name,EmailId,OrganisationName,Location,Password)
```

```
values(?,?,?,?,?)")
        ibm_db.bind_param(stmt,1,name)
        ibm_db.bind_param(stmt,2,email)
        ibm_db.bind_param(stmt,3,org)
        ibm_db.bind_param(stmt,4,location)
        ibm_db.bind_param(stmt,5,pwd)
        ibm_db.execute(stmt)
        return render_template('Signin.html')
    else:
         return render_template('Signup.html')
@app.route('/signin',methods=['GET','POST'])
def signin():
    if request.method=="POST":
        email=request.form["email"]
        pwd=request.form["password"]
        sql=f"select * from RetailersInformation where EmailId='{email}' and
Password='{pwd}'"
        stmt=ibm_db.exec_immediate(conn,sql)
        flag=ibm_db.fetch_row(stmt)
        if flag:
            return redirect(url_for("home"))
        else :
              return "Invalid id or password"
    else:
        return render_template("Signin.html")

@app.route('/forgot_password',methods=['GET','POST'])
def forgot_password():
    if request.method=="GET":
         return render_template("get_email.html")
```

```
    elif request.method=="POST":
        email=request.form['email']
        sg =
sendgrid.SendGridAPIClient(api_key="SG.loccRQHCRomwrWTcjZmnfA.DWmwtnUCi
xyx1Ng0ojgp3lIzU_1BeT-ZpXSbu-lOMc4")
        from_email = Email("dineshrs2002@gmail.com")  # Change to your verified
sender
        to_email = To(email)  # Change to your recipient
        subject = "[Inventory Management] Please reset your password"
        content = Content("text/html", f"We heard that you lost your Inventory
Management Portal password. Sorry about that!<br>But don’t worry! You can use the
following button to reset your password:<br> <button><a
href='http://127.0.0.1:5000/reset?email={email}'> reset</a></button>")
        mail = Mail(from_email, to_email, subject, content)
        mail_json = mail.get()
        response = sg.client.mail.send.post(request_body=mail_json)
        if response.status_code!=202:
            return "alert(Invalid mail)"
        else:
            return "<h1>Mail Sent Sucessfully</h1>"

@app.route('/reset', methods=["GET","POST"])
def reset_html():
    if request.method=="GET":
        email=request.args.get('email')
        pwd=request.form['password']
        sql=f"update RetailersInformation set password='{escape(pwd)}' where
emailid='{email}'"
        stmt=ibm_db.exec_immediate(conn,sql)
        return redirect(url_for("signin"))
```

```
    else:
        return render_template("ResetPassword.html")

@app.route('/addproduct', methods=["GET","POST"])
def addproduct():
    if request.method=="POST":
        name=request.form['productName']
        stock=request.form['qty']
        expiryDate=request.form['expiryDate']
        wholesalerName=request.form['wholesalerName']
        wholesalerNumber=request.form['wholesalerNumber']
        costPrice=request.form['costPrice']
        sellingPrice=request.form['sellingPrice']
        Retailer_email=request.form['Retailer_email']
        stmt=ibm_db.prepare(conn,"Insert into
ProductsInformation(Name,stock,wholesalername,wholesalernumber,costprice,selling
price,retaileremail) values(?,?,?,?,?,?,?)")
        ibm_db.bind_param(stmt,1,name)
        ibm_db.bind_param(stmt,2,stock)
        ibm_db.bind_param(stmt,3,wholesalerName)
        ibm_db.bind_param(stmt,4,wholesalerNumber)
        ibm_db.bind_param(stmt,5,costPrice)
        ibm_db.bind_param(stmt,6,sellingPrice)
        ibm_db.bind_param(stmt,7,Retailer_email)
        ibm_db.execute(stmt)
        return redirect(url_for("home"))
    else:
        return render_template("AddProduct.html")

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

```
def home():
    return render_template("Dashboard.html")

if _name=="main_":
    app.run(debug=True)
@app.route('/editproduct', methods=["POST"])
def editproduct():
    pid=request.args.get("id")
    name=request.form['productName']
    stock=request.form['qty']
    expiryDate=request.form['expiryDate']
    wholesalerName=request.form['wholesalerName']
    wholesalerNumber=request.form['wholesalerNumber']
    costPrice=request.form['costPrice']
    sellingPrice=request.form['sellingPrice']
    Retailer_email=request.form['Retailer_email']
    sql=f"update ProductsInformation set
productname='{name}',stock='{stock}',wholesalerName='{wholesalerName}',wholesal
erNumber='{wholesalerNumber}',costPrice='{costPrice}',sellingPrice='{sellingPrice}',R
etailer_email='{Retailer_email}'  where productid='{pid}'")
    stmt=ibm_db.exec-immediate(conn,sql)
    return redirect(url_for("home"))

@app.route('/deleteproduct', methods=["POST"])
def deleteproduct():
    pid=request.args.get('productid')
    sql=f"Delete from ProductsInformation where productid='{pid}'"
    stmt=ibm_db.exec_immediate(conn,sql)
    return redirect(url_for("home"))
```

Github link:

https://github.com/IBM-EPBL/IBM-Project-21229-1659775361

Demo Video Link:

https://vimeo.com/772988937