PROJECT REPORT

INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

submitted by

PNT2022TMID42092

Reni P

Vanteru srividya

Alice bala rose marry

Revathi S

TABLE OF CONTENTS

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

    7.1 Feature 1

    7.2 Feature 2

    7.3 Database Schema (if Applicable)

8. **TESTING**

    8.1 Test Cases

    8.2 User Acceptance Testing

9. **RESULTS**

    9.1 Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

    Source Code, GitHub & Project Demo Link

1. INTRODUCTION

1.1 PROJECT OVERVIEW

With IBM Cloud IaaS, organizations can deploy and access virtualized IT resources -- such as compute power, storage and networking -- over the internet. For compute, organizations can choose between bare-metal or virtual servers.

With IBM Cloud PaaS -- which is based on the open source cloud platform Cloud Foundry -- developers can use IBM services to create, manage, run and deploy various types of applications for the public cloud, as well as for local or on-premises environments. IBM Cloud supports various programming languages, such as Java, Node.js, PHP and Python and extends to support other languages.

1.2 PURPOSE

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.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Inventory Management System is an integral part of all organizations to manage the information about availability of items in stock and its issues and returns. In this post we will learn how to create a simple online Inventory Management System that allows you to add items, accept requests from employees and Issue items against their requests. The non-consumable items can be returned thus updating the stock.

2.2 REFERENCES

The relationship of financial and inventory performance of manufacturing firms in Indian context.

Gaur and Bhattacharya (2011)

Attempted to study the linkage between the performance of the components of inventory such as raw material, work in progress and finished goods and financial performance of Indian manufacturing firms. The study revealed that finished goods inventory as inversely associated with business performance while raw material inventory and work in progress did not have much effect on same. They emphasised that instead of focusing on total inventory, an attempt should be made to concentrate on individual components of inventory so as to adequately manage the same. They concluded that managers not paying heed to inventory performance may become weak in combating competitors.

Inventory management practices and business performance for small scale enterprises in Kenya.

Nyabwanga and Ojera (2012)

They Highlighted the association between inventory management practices and business performance of smallscale enterprises (SSEs), in Kisii Municipality, Kisii County, Kenya. They used a cross-sectional survey study based on a small sample size of 79 SSEs. The empirical results disclosed that a positive significant relationship existed between business performance and inventory management practices with inventory budgeting having the maximum influence on business performance.

Impact of inventory management on the profitability of SMEs in Tanzania

Madishetti, Srinivas & Kibona, Deogratias. (2013).

A survey conducted on all the eight (8) sugar manufacturing firms in Kenya established that there is generally positive correlation between each of inventory management practices.
Specific performance indicators were proved to depend on the level of inventory management practices.
They established that Return on Equity had a strong correlation with lean inventory system and strategic supplier partnerships.
As such, they concluded that the performance of sugar firms could therefore be stated as being a function of their inventory

International Journal of Engineering Research

Srinivas Rao Kasisomayajula(2014)

An analytical study was conducted on" Inventory Management in Commercial Vehicle Industry In India". A sample of five companies' was selected for study. The study concluded that all the units in the commercial vehicle industry have significant relationship between Inventory and Sales. Proper management of inventory is important to maintain and improve the health of an organization. Efficient management of inventories will improve the profitability of the organization.

2.1 PROBLEM STATEMENT DEFINITION

Irrespective of the size of the business, inventory management is one of the most challenging processes in the retail sector.
In this industry, the efficiency of inventory management directly impacts customer satisfaction. As retail is a fast-paced, and customer-facing sector, customer satisfaction is core to its business growth.
The inventory process involves multiple intricate aspects that drive accurate product delivery. Even a single error in the process can have expensive and long-term consequences. This will eventually affect the company's growth and reputation.
Thus, retail companies need to understand and analyze the risks involved in inventory management. Only then can companies find proactive solutions to the problems.

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

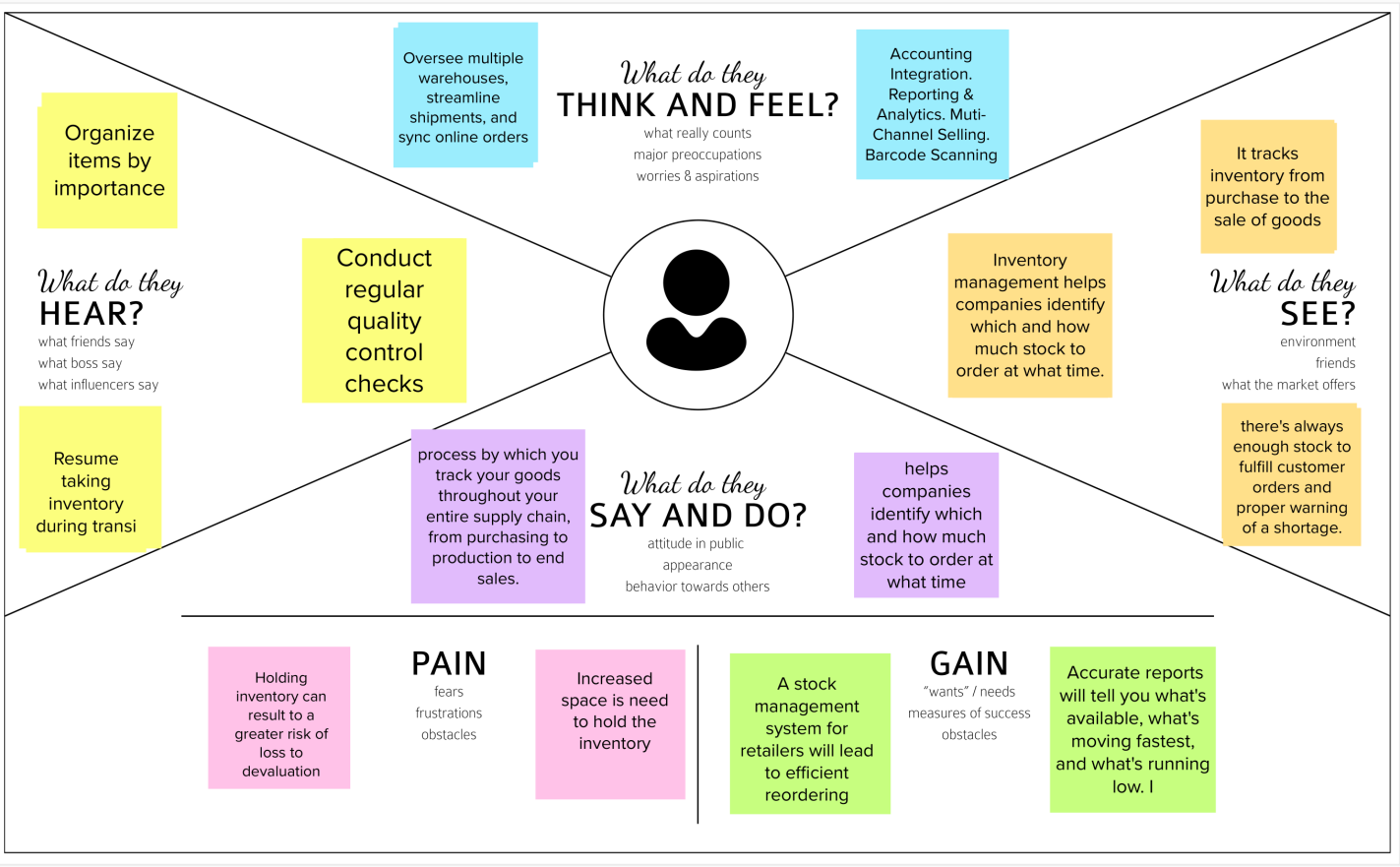
Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

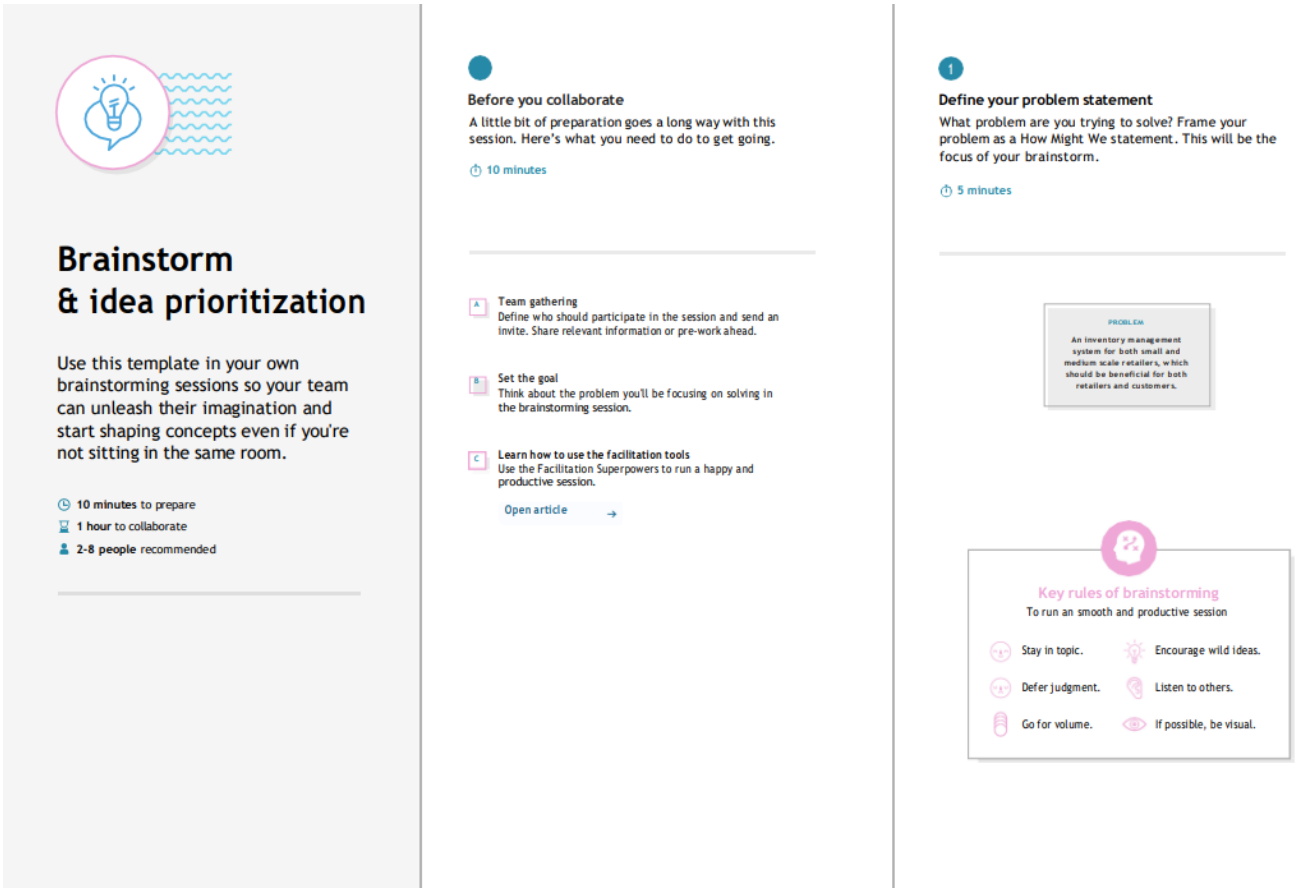
Build empathy and keep your focus on the user by putting yourself in their shoes.



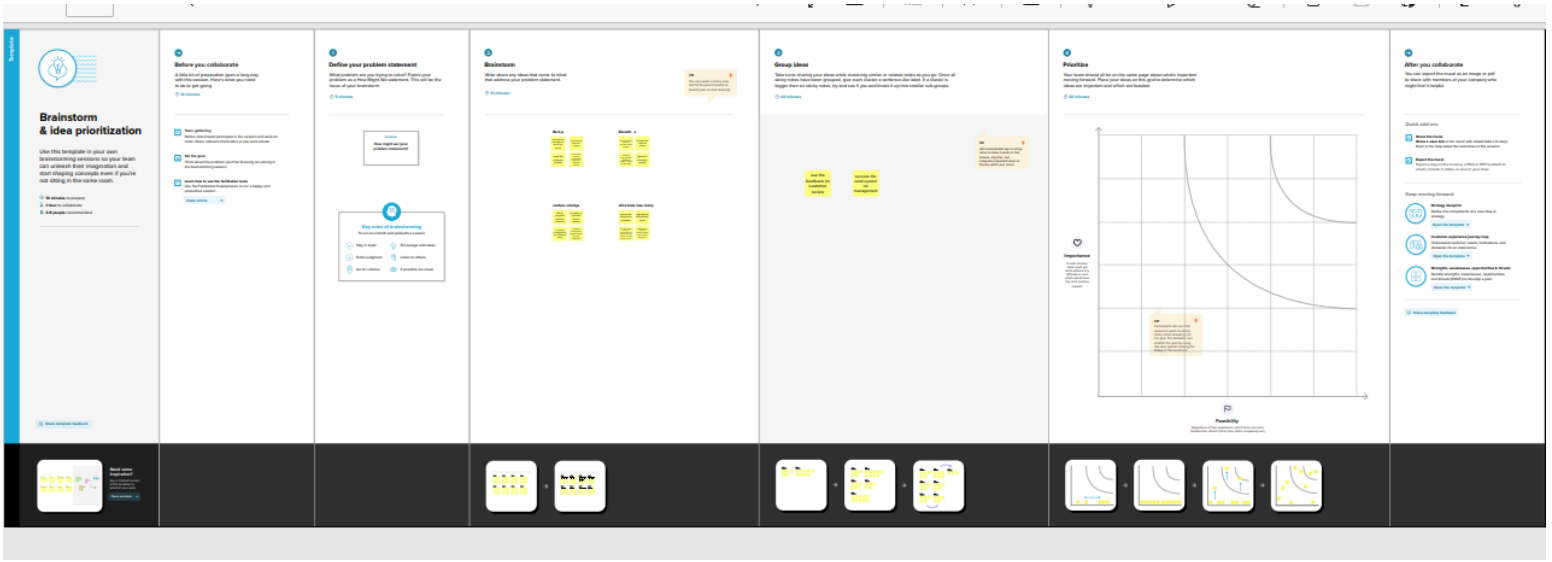
Share your feedback

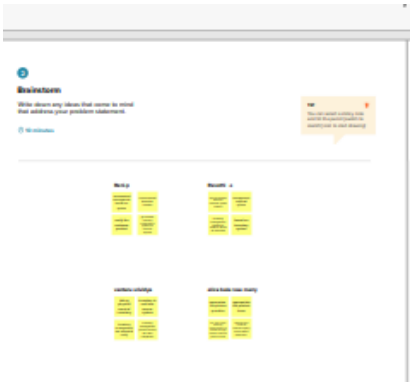
3.2 Ideation& Brainstorming

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



Step-2: Brainstorm, Idea Listing and Grouping





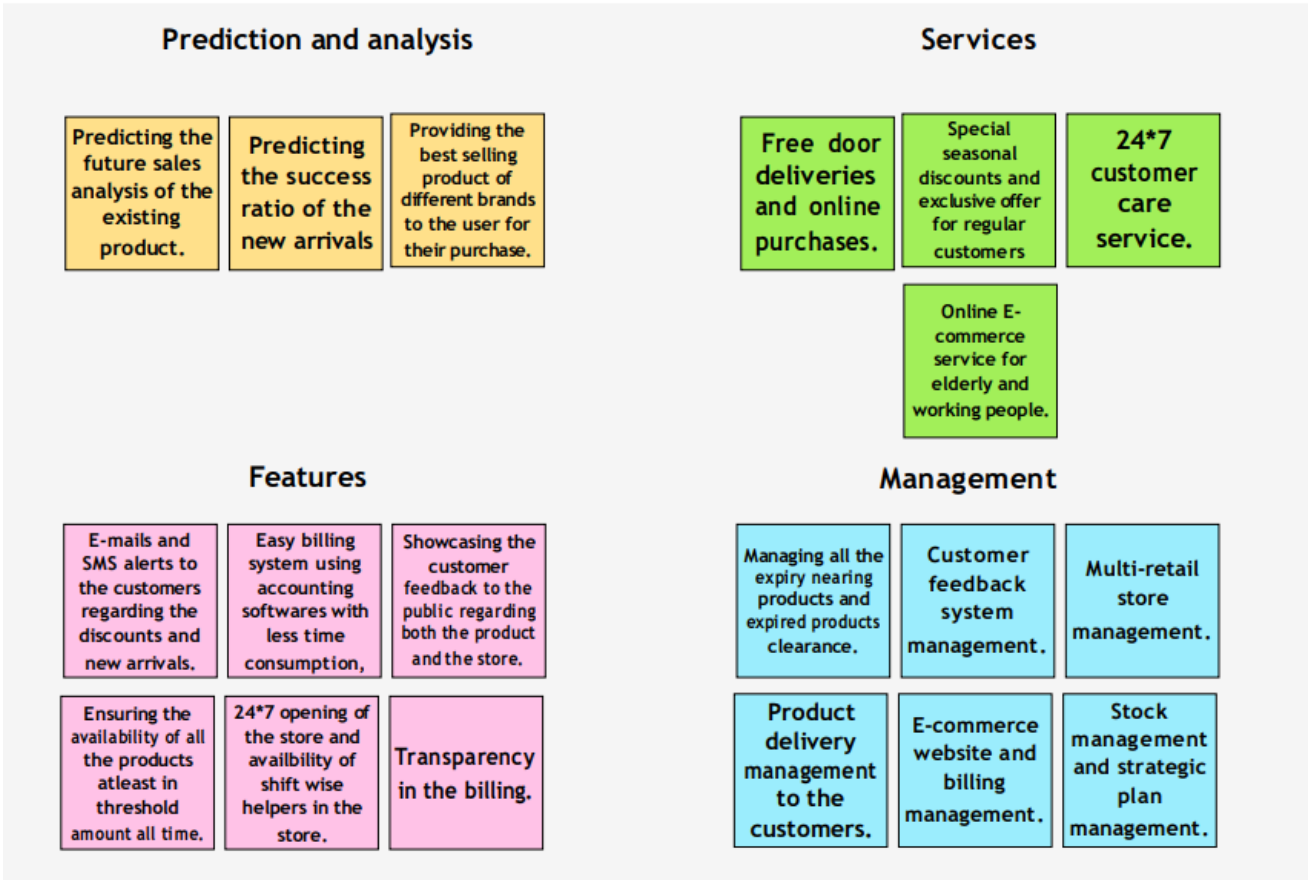
Step-3: Idea Prioritization

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes



PROPOSED SLOUTION

S.No.	Parameter	Description
❖	Problem Statement (Problem to be solved)	• The retailers generally facing issues in recording the stocks and its threshold limit available. • The customers are not satisfied with the retailers store since it doesn't have enough supplements and the deliveries were not made on time.
❖	Idea / Solution description	• This proposed system will have a daily update system whenever a product is sold or it is renewed more. • The product availability is tracked daily and an alert system in again kept on to indicate those products which falls below the threshold limit. • All the customers can register their accounts after which they will be given a login credentials which they can use whenever they feel like buying the stocks. • The application allows the customers to know all the present time available stocks and also when the new stock will be available on the store for them to buy.

❖	Novelty / Uniqueness	• Certain machine learning algorithms are used to predict the seasonal high selling products which can be made available during that time. • Prediction of the best selling brand of all certain products based on their popularity, price and customer trust and satisfaction will be implemented. • Notifications will be sent to the retailers if any product that the customers have been looking for is not available so that the product can be stocked up soon.
❖	Social Impact / Customer Satisfaction	• The customers will be highly satisfied since the wasting of time while searching for an unavailable product is reduced. • The work load of the retailers will be minimized if the system is automated every day and during every purchase. • The customer satisfaction will be improved for getting appropriate response from the retailers and that too immediately.
❖	Business Model (Revenue Model)	• Hereby we can provide a robust and most reliable inventory management system by using: 1. ML algorithms for all the prediction purposes using all the past dataset since datasets are undoubtedly available in huge amounts. 2. Can deploy the most appropriate business advertising models. 3. To establish a loss preventing strategy. 4. And to ensure the all time, any where availability of products system.

❖	Scalability of the Solution	• Implementation of anyone and anywhere using system can be helpful for even a commoner to buy the products. • Daily and Each time purchase updation of the stock for preventing inventory shrinkage.

3.3 PROBLEM SOLUTION FIT

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Our proposed model targets retailers to have a track on their stock availability.

6. CUSTOMER CC

Not having knowledge of the available and upcoming demands more over existing solutions are not so far good in intimating the retailer about the stock which is about to get over.

5. AVAILABLE SOLUTIONS AS

Data from different key performance metrics, which take into account several aspects of the inventory influencing the business and Methodology (such as AUD and MDP) to forecast revenue and discount on the products. Tools (such as RFID and barcodes) to maintain correct records across digital and physical databases.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P

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

9. PROBLEM ROOT CAUSE RC

Most of the retailers buy stocks which cannot be kept in account or tracked since the stock count is more in number.

7. BEHAVIOUR BE

Feels so hard to manage the inventory information. By this inventory management system one can manage the whole inventory information and it is time saving.

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

Identify strong TR & EM

3. TRIGGERS TR

Friends and family who run whole sale shops or markets will be encouraged by this inventory management system.

4. EMOTIONS: BEFORE / AFTER EM

Before: tired, fear, forgetful

After: Stress free, confident, relief

10. YOUR SOLUTION SL

We aim to design an Inventory Management system which is used to manage the inventory details and aims to save for the future investments. User can track the stocks sold and yet to be sold and can visualize it.

8. CHANNELS of BEHAVIOUR CH

ONLINE

Use websites to gather informationon how to use it.

OFLINE

Check regularly and intimate the retailer.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR. No.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	User Registration	Registration through registration form. Registration through One-Tap Google Sign-in.
FR-2	User Authentication and Confirmation	Authentication via Google Authentication. Confirmation via Email. Confirmation via OTP.
FR-3	Product management	Quickly produce reports for single or multiple products. Track information of dead and fast-moving products. Track information of suppliers and manufacturers of the product.
FR-4	Audit Monitoring	The technique of tracking crucial data is known as audit tracking. Monitor the financial expenses carried out throughout the whole time (from receiving order of the product to delivery of the product).
FR-5	Historical Data	Data of everything should be stored for analytics and forecasting.

4.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The UI should be accessible to everybodydespite of there diversity in languages. People with some impairments should also be able to use the application with ease. (Example,integrate google assistant so that blind people can use it). .
NFR-2	Security	The security requirements deal with the primary security. Only authorized users can access the systemwith their credentials. Administrator or the concerned security team shouldbe alerted on any unauthorized access or data breaches so as to rectify it immediately.
NFR-3	Reliability	The software should be able to connect to the database in the event of the server being down due toa hardware or software failure.

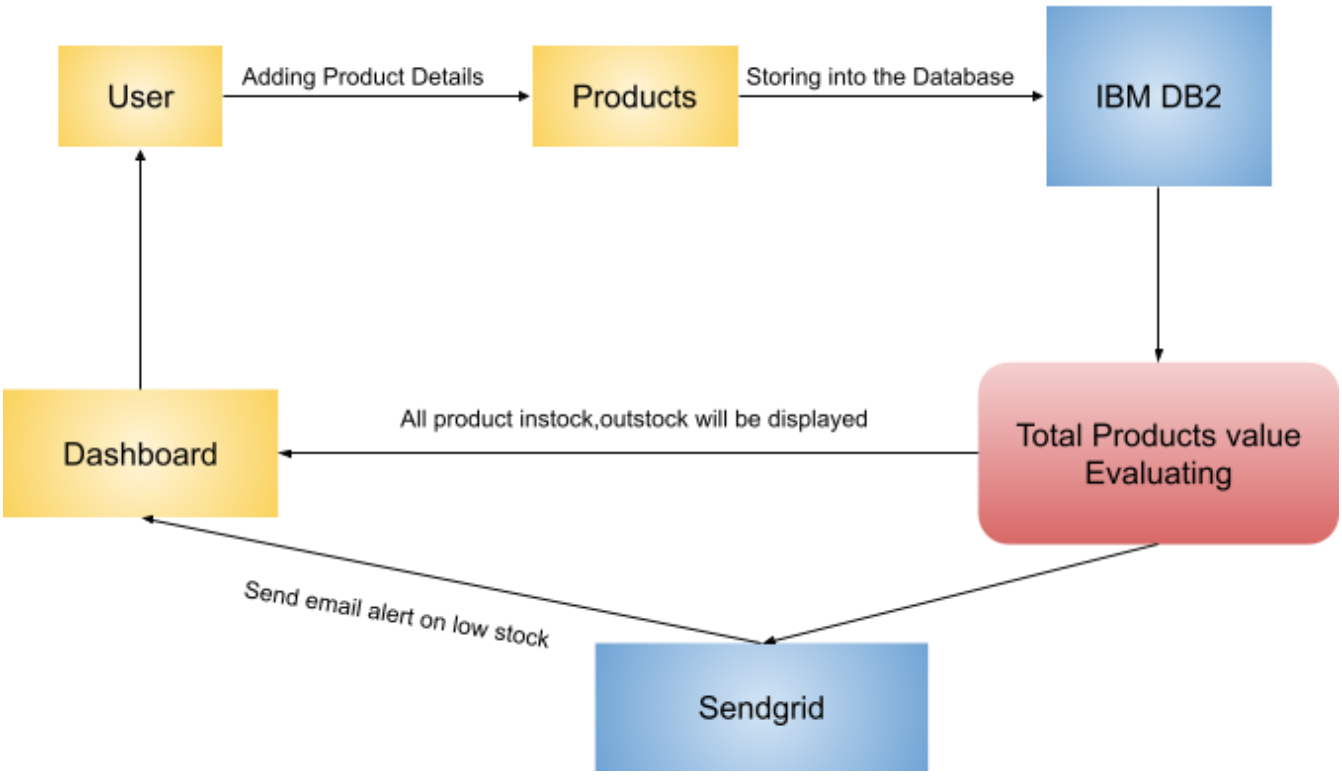
		The users must me intimated by the periodic maintenance break of the server so that they willbe aware of it.
NFR-4	Performance	Performance of the app should be reliable withhigh-end servers on which the software is running.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

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 enters and leaves the system, what changes the information, and where data is stored.

Example:(Simplified)FLOW



5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered

Solution Architecture Diagram:

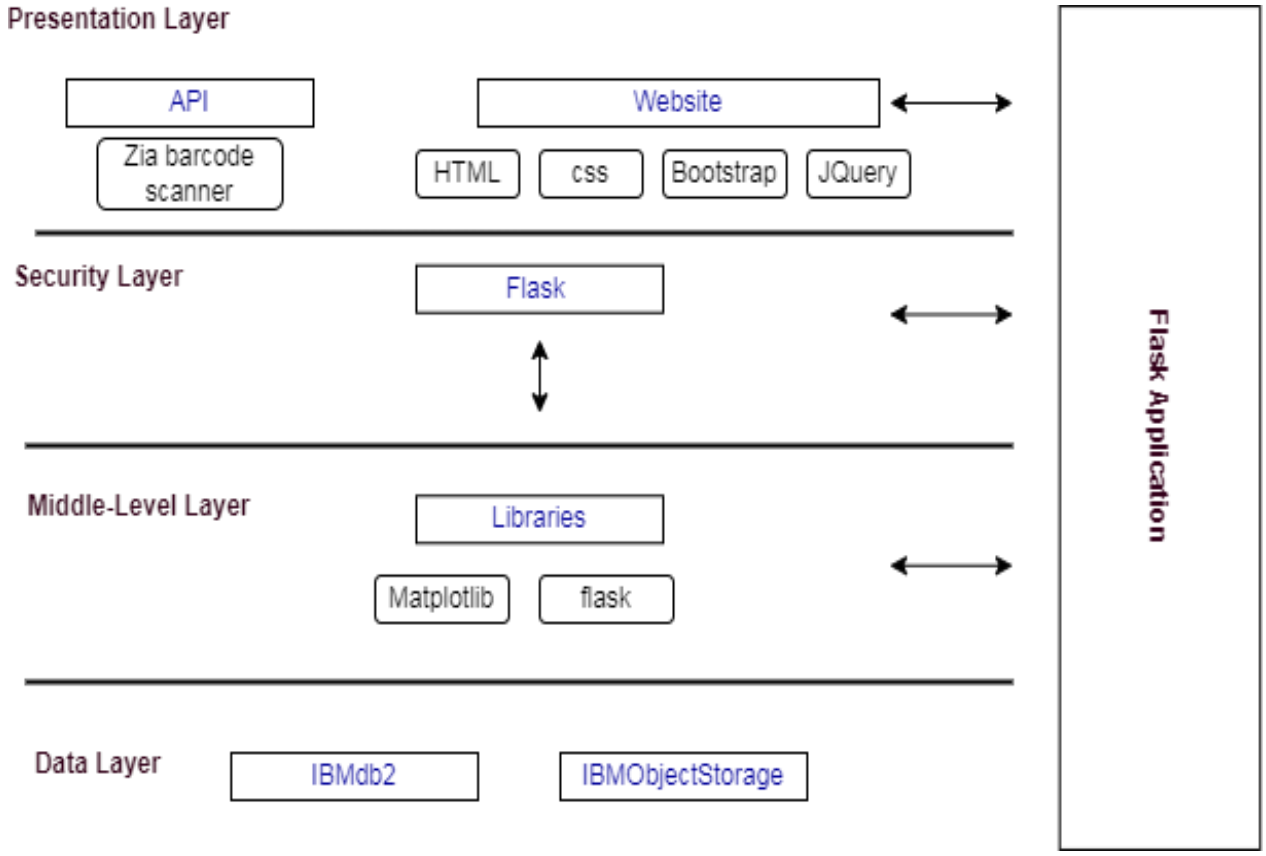
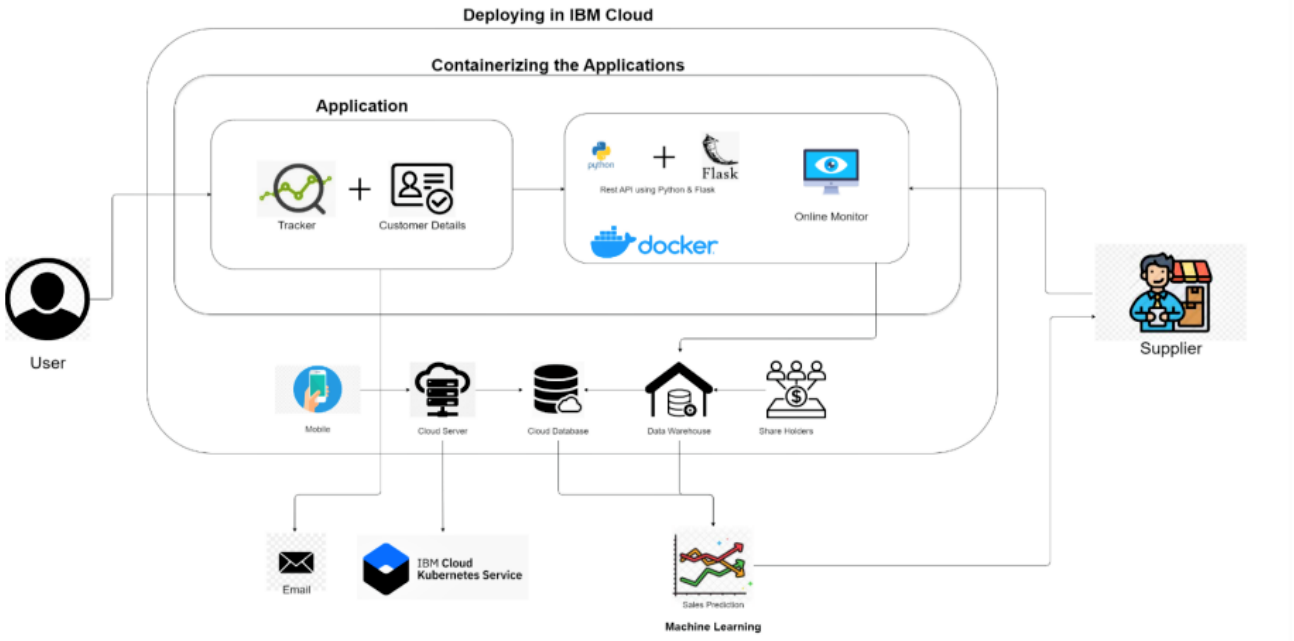


Figure 1: Model Architecture of the cloud development for retailers

Solution Architecture Diagram:



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Retailer(Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I will be redirected to login page	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can verify the OTP number	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	As a user,I can update stock in & out count details	Updation can be made through barcode scanning	High	Sprint -2
	Dashboard	USN-7	As a user,I can check the low stock details through alert message	Alert message can be received by registered mail	High	Sprint -1
		USN-8	As a user,I can check the total product details	I can view the value of total products in the stock	Medium	Sprint -2
		USN-9	As a user,I can check the high demand product details	I can update sales details of the products	High	Sprint -2
		USN-10	As a user,I can generate the invoice details	I can add incoming stock details	High	Sprint -1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Retailer(Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I will be redirected to login page	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can verify the OTP number	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	As a user I can update stock in & out count details	Updation can be made through barcode scanning	High	Sprint -2
	Dashboard	USN-7	As a user I can check the low stock details through alert message	Alert message can be received by registered mail	High	Sprint -1
		USN-8	As a user I can check the total product details	I can view the value of total products in the stock	Medium	Sprint -2
		USN-9	As a user I can check the high demand product details	I can update sales details of the products	High	Sprint -2
		USN-10	As a user I can generate the invoice details	I can add incoming stock details	High	Sprint -1

6.2 SPRINT DELIVERY SCHEDULE

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	4
Sprint-1		USN-2	As a user, I can register for the application through E-mail	1	Medium	4
Sprint-1	Confirmation	USN-3	As a user, I will receive confirmation email once I have registered for the application	2	Medium	4

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	2	High	4
Sprint-2	Dashboard	USN-5	As a user, I can view the products which are available	4	High	4
Sprint-2	Add items to cart	USN-6	As a user, I can add the products I wish to buy to the carts.	5	Medium	4
Sprint-3	Stock Update	USN-7	As a user, I can add products which are not available in the dashboard to the stock list.	5	Medium	4
Sprint-4	Request to Customer Care	USN-8	As a user, I can contact the Customer Care Executive and request any services I want from the customer care.	5	Low	4
Sprint-4	Contact Administrator	USN-9	I can be able to report any difficulties I experience as a report	5	Medium	4

7. CODING & SOLUTIONING

7.1 FEATURE 1

- track raw material and finished goods for manufacturers
- track lot numbers, FDA, and recall
- support for kitting and costing of kits from components and labor

7.2 FEATURE 2

- Business owners manage the inventory well with the help of inventory software. Managers balance the demand and supply of the company products efficiently.
- This is why businesses are capable of generating a huge amount of revenue on an annual basis.

7.3 DATABASE SCHEMA (IF APPLICABLE)

- The shop has an inventory of products. Each product has a price, but this price should vary depending on sales.
- Customers can make orders for multiple products at a time, and should be able to see their order history.
- When the order has been completed, there should be a track and trace number.

```
import os

import numpy as np

from flask import Flask, render_template, request, send_from_directory,
url_for

#from gevent.pywsgi import WSGIServer

from keras.models import load_model

from keras.preprocessing import image

from PIL import Image

from werkzeug.utils import redirect, secure_filename


UPLOAD_FOLDER = 'D:/NalaiyaThiran/projFiles/data'


app = Flask(_name_)

app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER


model = load_model("./model/mnist_digit_recog_cnn.h5")


@app.route('/')

def index():

    return render_template('index.html')


@app.route('/web', methods=['GET', 'POST'])

def web():
```

```
    if request.method == "POST":
        f = request.files["image"]
        basepath = os.path.dirname(__file__)
        filepath = os.path.join(basepath, 'data', f.filename)
        f.save(filepath)
        # img = image.load_img(filepath, target_size=(64, 64))
        # x = image.img_to_array(img)
        # x = np.expand_dims(x, axis=0)

        # filepath = secure_filename(f.filename)
        # f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        # upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(filepath).convert("L")     # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image

        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement

        pred = model.predict(im2arr)
```

```
        num = np.argmax(pred, axis=1)  # printing our Labels


        return render_template('web.html', num=str(num[0]))
    return render_template('web.html')


if __name__== '__main_':
    app.run(debug=True, threaded=False)
```

8. TESTING

8.1 TEST CASE

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on selected project and gathering information by referring the project's related technical papers, research publications, etc.	28 SEPTEMBER 2022
Prepare Empathy Map	Prepare empathy map canvas to capture the user's pains & gains and prepare the list of problem statements.	24 SEPTEMBER 2022
Ideation	To list by the organizing brainstorm sessions and prioritize the top three ideas based on the feasibility and importance.	25 SEPTEMBER 2022
Proposed Solution	To prepare the proposed solution documents, which includes the novelty, feasibility of ideas, business model, social impact, scalability of the solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Preparing the problem solution fit document.	30 SEPTEMBER 2022

Solution Architecture	To prepare the solution architecture document	28 SEPTEMBER 2022
Customer Journey	Prepare the customers journey map help the customers understand the user interaction and experiences with the application from the beginning to the end.	20 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for the review.	9 OCTOBER 2022
Technology Architecture	Prepare technical architecture diagram.	10 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones and activity of the project.	30 OCTOBER 2022
Project Development – Delivery of Sprint-1, 2, 3 &	Develop and submit the developed code by testing it and having no errors.	IN PROGRESS...

8.2 USER ACCEPTANCE TESTING

1. 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	0	0	0	0	0
Duplicate	0	0	0	0	0
External	0	0	0	0	0
Fixed	0	0	0	0	0
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	0	0	0	0	0

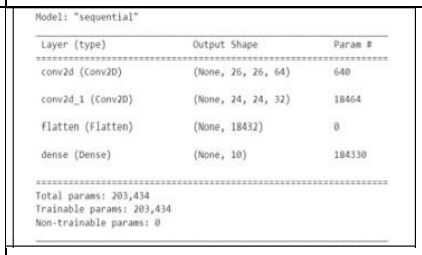
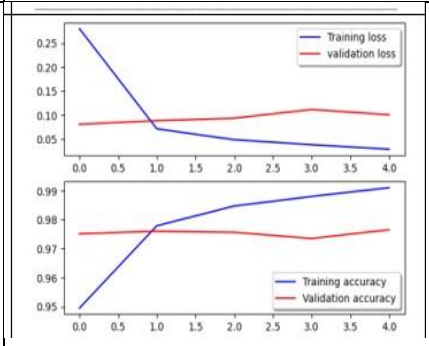
2. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Client Application	5	0	0	5
Security	5	0	0	5
Final Report Output	5	0	0	5
Version Control	5	0	0	5

9. RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	
2.	Accuracy	Training Accuracy – 99% Validation Accuracy – 97%	
3.	Confidence Score (OnlyYolo Projects)	Class Detected – Confidence Score –	

10. ADVANTAGES & DISADVANTAGES

ADVANTAGE

- Real-time inventory tracking helps you improve inventory management and ensures that you have optimal stock available to fulfill orders.
- However, for most retail businesses, the inventory accuracy is merely 63%. With accurate inventory tracking, you can eliminate over-stocking, and in turn, reduce the cost and manual efforts required in holding it.
- There are many ways to improve inventory efficiency. Some of the most proven methods include:

.

- **DISADVANTAGE**

  - Not 100% accurate, there are likely to be some mistakes made during the method.

11. CONCLUSION

Feedback offers retailers a valuable tool to improve the way they engage with their customers.

It lets customers know their opinion matters.

It shows that retailers listen to their customers, value their opinion, and care about their experience.

12. FUTURE SCOPE

Inventory may be seen as the bloodstream of any competitive business. Its unobstructed flow is critical

Like the life-sustaining “oxygen” that is carried by the blood to various parts of the human body,

Like the doctor who extracts a few samples of blood from a patient’s arm in order to evaluate health.

Accumulation of unnecessary inventory is frequently symptomatic of bigger problems that lurk

Its significance cannot be overstated.

13. APPENDIX

SOURCE CODE

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Sidebar Menu</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.
css" rel="stylesheet"
    integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTR
i" crossorigin="anonymous" />
  <link rel="stylesheet" href="static/css/style.css" />
</head>

<body>
  <div class="wrapper">
    <!-- Sidebar  -->
    <nav id="sidebar">
      <div class="sidebar-header">
        <h3>Inventory</h3>
      </div>

      <ul class="list-unstyled components">
        <li class="active">
          <a>Dashboard</a>
        </li>
        <li>
          <a href="#">Add item</a>
        </li>
        <li>
          <a href="#pageSubmenu">Pages</a>
        </li>
        <li>
          <a href="#">delete items</a>
```

```
      </li>
      <li>
        <a href="#">Help</a>
      </li>
    </ul>
  </nav>

  <!-- Page Content  -->
  <div id="content">
    <h2>Dashboard</h2>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
      eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
ad
      minim veniam,
    </p>
  </div>
 </div>
 <nav class="navbar navbar-expand-lg bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNavDropdown"
      aria-controls="navbarNavDropdown" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-
toggle="dropdown" aria-expanded="false">
            Dropdown link
          </a>
```

```
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li>
              <a class="dropdown-item" href="#">Something else here</a>
            </li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>

<nav>
  <div class="search">
    <i class="bx bx-search"></i>
    <input type="text" class="hide" placeholder="Quick Search ..." />
  </div>

  <div class="sidebar-links">
    <ul>
      <div class="active-tab"></div>
      <li class="tooltip-element" data-tooltip="0">
        <a href="#" class="active" data-active="0">
          <div class="icon">
            <i class="bx bx-tachometer"></i>
            <i class="bx bxs-tachometer"></i>
          </div>
          <span class="link hide">Dashboard</span>
        </a>
      </li>
      <li class="tooltip-element" data-tooltip="1">
        <a href="#" data-active="1">
          <div class="icon">
            <i class="bx bx-folder"></i>
            <i class="bx bxs-folder"></i>
          </div>
          <span class="link hide">Products</span>
        </a>
      </li>
      <li class="tooltip-element" data-tooltip="2">
        <a href="#" data-active="2">
          <div class="icon">
```

```
      <i class="bx bx-message-square-detail"></i>
      <i class="bx bxs-message-square-detail"></i>
    </div>
    <span class="link hide">Add items</span>
  </a>
</li>
<li class="tooltip-element" data-tooltip="3">
  <a href="#" data-active="3">
    <div class="icon">
      <i class="bx bx-bar-chart-square"></i>
      <i class="bx bxs-bar-chart-square"></i>
    </div>
    <span class="link hide">Remove items</span>
  </a>
</li>
<div class="tooltip">
  <span class="show">Dashboard</span>
  <span>Projects</span>
  <span>Messages</span>
  <span>Analytics</span>
</div>

<li class="tooltip-element" data-tooltip="1">
  <a href="#" data-active="5">
    <div class="icon">
      <i class="bx bx-help-circle"></i>
      <i class="bx bxs-help-circle"></i>
    </div>
    <span class="link hide">Help</span>
  </a>
</li>
<li class="tooltip-element" data-tooltip="2">
  <a href="#" data-active="6">
    <div class="icon">
      <i class="bx bx-cog"></i>
      <i class="bx bxs-cog"></i>
    </div>
    <span class="link hide">Settings</span>
  </a>
</li>
<div class="tooltip">
  <span class="show">Tasks</span>
  <span>Help</span>
```

```
        <span>Settings</span>
      </div>
    </ul>
  </div>

  <div class="sidebar-footer">
    <a href="#" class="account tooltip-element" data-tooltip="0">
      <i class="bx bx-user"></i>
    </a>
    <div class="admin-user tooltip-element" data-tooltip="1">
      <div class="admin-profile hide">
        <img src="./img/face-1.png" alt="" />
        <div class="admin-info">
          <h3>Ganesh</h3>
          <h5>user</h5>
        </div>
      </div>
      <a href="#" class="log-out">
        <i class="bx bx-log-out"></i>
      </a>
    </div>
    <div class="tooltip">
      <span class="show">Ganesh</span>
      <span>Logout</span>
    </div>
  </div>
 </nav>

 <main>
  <h1>My Dashboard</h1>
 </main>

 <script src="static/js/app.js"></script>
</body>

</html>
```

login.html:

```
{% extends 'base.html '%}
{% block head %}
<title>Login page</title>
{% endblock%}

{%block body%}

<main class="container ">
   <div class="mx-auto mt-5 border bg-light login-card " style="width:500px;">
      <h2 class='mx-4 mt-2'>LOGIN</h2>
      <form action="{{url_for('login') }}" method="post">
         <div class="mx-4 mt-2 text-danger">{{ msg }}</div>
         <div class="my-2 mx-4">
            <label for="username">username</label>
            <input type="text" class="form-control" placeholder="adc@gmail.com"
name="username" required />
         </div>
         <div class="my-2 mx-4">
            <label for="password_1">password</label>
            <input type="password" class="form-control" name="password_1"
required />
         </div>
         <input type="submit" value="submit" class="btn btn-primary my-4 mt-2
mx-4" />


      </form>
      <p>Don't have an account?<a href="{{ url_for('signup') }}"> Sign Up</a>
   </div>
</main>
</p>
```

```
</main>
{% endblock%}
```

App.py

```
from flask import Flask, render_template, url_for, request, redirect, session, make_response
import sqlite3 as sql
from functools import wraps
import re
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30367;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gkx49901;PWD=kvWCsySl7vApfsy2", '', '')

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



def rewrite(url):
    view_func, view_args = app.create_url_adapter(request).match(url)
    return app.view_functions[view_func](**view_args)



def login_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if "id" not in session:
            return redirect(url_for('login'))
        return f(*args, **kwargs)
```

```
    return decorated_function


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


@app.route('/user/<id>')
@login_required
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('/login', methods=['GET', 'POST'])
def login():
    global userid
    msg = ''

    if request.method == 'POST':
        un = request.form['username']
        pd = request.form['password_1']
        print(un, pd)
        sql = "SELECT * FROM register WHERE email =? 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['EMAIL']
            userid = account['EMAIL']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'

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


@app.route('/signup', methods=['POST', 'GET'])
def signup():
    mg = ''
    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:
            mg = 'Account already exits!!'

        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            mg = 'Please enter the avalid email address'
        elif not re.match(r'[A-Za-z0-9]+', username):
            ms = 'name must contain only character and number'
        else:
            insert_sql = 'INSERT INTO register
(USERNAME,FIRSTNAME,LASTNAME,EMAIL,PASSWORD) VALUES (?,?,?,?,?)'
            pstmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(pstmt, 1, username)
            ibm_db.bind_param(pstmt, 2, "firstname")
            ibm_db.bind_param(pstmt, 3, "lastname")
            # ibm_db.bind_param(pstmt,4,"123456789")
            ibm_db.bind_param(pstmt, 4, email)
            ibm_db.bind_param(pstmt, 5, pw)
            print(pstmt)
            ibm_db.execute(pstmt)
            mg = 'You have successfully registered click login!'
            return render_template("login.html", meg=mg)

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



@app.route('/dashboard', methods=['POST', 'GET'])
@login_required
def dashBoard():
    headings = ("id", "name", "order_id", "location")
```

```
    data = (
        ("1", "lorem", "ipsum", "dolor"),
        ("2", "lorem", "ipsum", "dolor"),
        ("3", "lorem", "ipsum", "dolor"),
        ("1", "lorem", "ipsum", "dolor"),
        ("2", "lorem", "ipsum", "dolor"),
        ("3", "lorem", "ipsum", "dolor"),
    )
    return render_template("dashboard.html", headings=headings, data=data)


@app.route('/orders', methods=['POST', 'GET'])
@login_required
def orders():
    return render_template("orders.html")


@app.route('/suppliers', methods=['POST', 'GET'])
@login_required
def suppliers():
    return render_template("suppliers.html")


@app.route('/profile', methods=['POST', 'GET'])
@login_required
def profile():
    return render_template("profile.html")


@app.route('/logout', methods=['GET'])
@login_required
```

```
def logout():
    print(request)
    resp = make_response(render_template("login.html"))
    session.clear()
    return resp



if __name__ == '__main__':
    app.run(debug=True)
```

css

```
/*
    DEMO STYLE
*/


@import
'https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700';
* {
  box-sizing: border-box;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
}
body {
  font-family: Helvetica;
  -webkit-font-smoothing: antialiased;
  background: rgba(71, 147, 227, 1);
}
/* h2 {
  text-align: center;
  font-size: 18px;
  text-transform: uppercase;
```

```
  letter-spacing: 1px;
  color: white;
  padding: 30px 0;
} */

/* Table Styles */

.table-wrapper {
  margin: 10px 70px 70px;
  box-shadow: 0px 35px 50px rgba(0, 0, 0, 0.2);
}

.fl-table {
  border-radius: 5px;
  font-size: 12px;
  font-weight: normal;
  border: none;
  border-collapse: collapse;
  width: 100%;
  max-width: 100%;
  white-space: nowrap;
  background-color: white;
}

.fl-table td,
.fl-table th {
  text-align: center;
  padding: 8px;
}

.fl-table td {
```

```
  border-right: 1px solid #f8f8f8;
  font-size: 12px;
}

.fl-table thead th {
  color: #ffffff;
  background: #4fc3a1;
}

.fl-table thead th:nth-child(odd) {
  color: #ffffff;
  background: #324960;
}

.fl-table tr:nth-child(even) {
  background: #f8f8f8;
}

/* Responsive */

@media (max-width: 767px) {
  .fl-table {
    display: block;
    width: 100%;
  }
  .table-wrapper:before {
    content: 'Scroll horizontally >';
    display: block;
    text-align: right;
    font-size: 11px;
    color: white;
```

```
  padding: 0 0 10px;
}
.fl-table thead,
.fl-table tbody,
.fl-table thead th {
  display: block;
}
.fl-table thead th:last-child {
  border-bottom: none;
}
.fl-table thead {
  float: left;
}
.fl-table tbody {
  width: auto;
  position: relative;
  overflow-x: auto;
}
.fl-table td,
.fl-table th {
  padding: 20px 0.625em 0.625em 0.625em;
  height: 60px;
  vertical-align: middle;
  box-sizing: border-box;
  overflow-x: hidden;
  overflow-y: auto;
  width: 120px;
  font-size: 13px;
  text-overflow: ellipsis;
}
.fl-table thead th {
```

```
    text-align: left;
    border-bottom: 1px solid #f7f7f9;
  }
  .fl-table tbody tr {
    display: table-cell;
  }
  .fl-table tbody tr:nth-child(odd) {
    background: none;
  }
  .fl-table tr:nth-child(even) {
    background: transparent;
  }
  .fl-table tr td:nth-child(odd) {
    background: #f8f8f8;
    border-right: 1px solid #e6e4e4;
  }
  .fl-table tr td:nth-child(even) {
    border-right: 1px solid #e6e4e4;
  }
  .fl-table tbody td {
    display: block;
    text-align: center;
  }
}
body {
  font-family: 'Poppins', sans-serif;
  background: #fafafa;
}

p {
  font-family: 'Poppins', sans-serif;
```

```
  font-size: 1.1em;
  font-weight: 300;
  line-height: 1.7em;
  color: #999;
}

a,
a:hover,
a:focus {
  color: inherit;
  text-decoration: none;
  transition: all 0.3s;
}

.navbar {
  padding: 15px 10px;
  background: #fff;
  border: none;
  border-radius: 0;
  margin-bottom: 40px;
  box-shadow: 1px 1px 3px rgba(0, 0, 0, 0.1);
}

.navbar-btn {
  box-shadow: none;
  outline: none !important;
  border: none;
}

.line {
  width: 100%;
```

```
  height: 1px;
  border-bottom: 1px dashed #ddd;
  margin: 40px 0;
}

/* ----------------------------------------------
    SIDEBAR STYLE
---------------------------------------------- */

.wrapper {
  display: flex;
  width: 100%;
  align-items: stretch;
}

#sidebar {
  min-width: 250px;
  max-width: 250px;
  background: #48494b;
  color: #fff;
  transition: all 0.3s;
}

#sidebar.active {
  margin-left: -250px;
}

#sidebar .sidebar-header {
  padding: 20px;
  background: #48494b;
}
```

```
#sidebar ul.components {
  padding: 20px 0;
  border-bottom: 1px solid #47748b;
}

#sidebar ul p {
  color: #fff;
  padding: 10px;
}
.project-title {
  font-size: 20px;
  padding-left: 10px;
  text-align: center;
}

#sidebar ul li a {
  padding: 10px;
  font-size: 1.1em;
  display: block;
}

#sidebar ul li a:hover {
  color: #7386d5;
  background: #fff;
}

#sidebar ul li.active > a,
a[aria-expanded='true'] {
  color: #fff;
  background: #48494b;
```

```
}

a[data-toggle='collapse'] {
  position: relative;
}

.dropdown-toggle::after {
  display: block;
  position: absolute;
  top: 50%;
  right: 20px;
  transform: translateY(-50%);
}

ul ul a {
  font-size: 0.9em !important;
  padding-left: 30px !important;
  background: #48494b;
}

ul.CTAs {
  padding: 20px;
}

ul.CTAs a {
  text-align: center;
  font-size: 0.9em !important;
  display: block;
  border-radius: 5px;
  margin-bottom: 5px;
}
```

```
a.download {
  background: #fff;
  color: #48494b;
}

a.article,
a.article:hover {
  background: #48494b !important;
  color: #fff !important;
}

.login-card {
  box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;
  border-radius: 10px;
  padding: 10px;
}
.login-card p {
  padding-left: 20px;
}
.login-card a {
  color: rgba(84, 84, 220, 0.888);
}

/* ------------------------------------------------
    CONTENT STYLE
------------------------------------------------ */

#content {
  width: 100%;
  padding: 20px;
```

```
  min-height: 100vh;
  transition: all 0.3s;
}

/* ----------------------------------------------
    MEDIAQUERIES
---------------------------------------------- */

@media (max-width: 768px) {
  #sidebar {
    margin-left: -250px;
  }
  #sidebar.active {
    margin-left: 0;
  }
  #sidebarCollapse span {
    display: none;
  }
}
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-43730-1660719039

PROJECT DEMO LINK

https://drive.google.com/drive/folders/1JZYiCXWcrBsRXor2hL1us84FsZkbJwJcZMk5CLoiC4s_QZgzRQa8UKaoPlJ7s_HQZa2ft-y4