# PROJECT REPORT

# INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

## SNS COLLEGE OF ENGINEERING

## COMPUTER SCIENCE AND ENGINEERING

**TEAM ID**: PNT2022TMID07720

**TEAM LEAD -BHAWINJASPER E -713519CECS004**

**TEAM MEMBER -1 -SNEHA G -713519CECS041**

**TEAM MEMBER -2 -SADHANANDAN S -713519CECS033**

**TEAM MEMBER -3 -NANDHAKUMAR G -713519CECS020**

**TABLE OF CONTENT:**

# INTRODUCTION

## 1.1 PROJECT OVERVIEW:

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

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

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

## 1.2 PURPOSE:

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

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

# LITERATURE SURVEY

## 2.1 EXISTING PROBLEM:

1. OVER STOCKING - Keeping too much stock on hand can be as problematic as having too little. Overstock impacts business cash flow and leads to inventory-related problems, such as storage and loss.

2. INSUFFICIENT ORDER MANAGEMENT - One of the most common challenges to sound inventory management is preventing the overselling of products and running out of inventory.

3. INCONSISTENCY OF DEMAND – The demand of product is not stable and it varies because of different reasons like season, trend, region's etc.

4. CUSTOMER SATISFACTION – Can't able to deliver the product to the customer in a disclosed time and also not able to satisfy the customer demand.

5. PROPER COMMUNICATION – There is a lack of communication regarding the stock details to the inventory management and to the production unit.

## 2.2 REFERENCES:

1. **A secure and efficient inventory management system for disasters**

**PROPOSED APPROACH:**

An efficient humanitarian inventory control model and emergency logistics system plays a crucial role in maintaining reliable flow of vital supplies to the victims located in the shelters and minimizing the impacts of the unforeseen disruptions that can occur. This system should not only allow the efficient usage and distribution of emergency supplies but should also offer the ability to be integrated with technologies such as Radio Frequency Identification Devices (RFIDs) for commodity tracking and logistics

**ADVANTAGES:**

- This system used RFIDs for tracking the commodity and maintaining its stocks
- model-free on-line control methodology is used to understand the efficiency and practicality of both algorithms in terms of computational times and accuracy of results.

**DISADVANTAGES:**

- This system just tracks the commodity and does not track whether they are satisfied or not.

## 2. Inventory Management Information System in Blood Transfusion Unit

**PROPOSED APPROACH:**

There are several blood components at the Blood Transfusion Unit to improve health services in Indonesia including Whole Blood, Packet Red Cell, Liquid Plasma, Fresh Frozen Plasma, Thrombocyte Concentrate, Kriopresipitat and Washed Erythrocyte. To provide services to consumers, this unit faces problem in the form of unbalance blood supply information and consumer demand. Consequently, management of this unit was difficult to manage the blood inventory. Aims of this study is to build an information system model using the system development life cycle approach in order to manage blood demand. Furthermore, this case adopted continuous review model to conduct the inventory policies involving safety stock, reorder point, and order quantity on each blood components. This study is able to provide benefits for Blood Transfusion Unit in order to increase service level to the customer. Further study is suggested to consider blood inventory simulation in developing several scenarios to manage blood demand.

**ADVANTAGES:**

- Blood bank is made online so we can easily track the availability
- Many lives had been saved.

**DISADVANTAGES:**

- The inventory doesn’t meet the exact need.
- Many still suffer because of lack of stock.

## 3. A Material Management in Construction Project Using Inventory Management System

**PROPOSED APPROACH:**

In this project we have prepare scheme of material management in the construction industry for building project also conducting survey of industry and determine the various format for construction material management. As well as talk over the tracking system of material management in the industry and also discuss the software development for proper management

**ADVANTAGES:**

- Fastest construction due to management of stocks.
- Easy to maintain stock details.
- Better customer experience.

**DISADVANTAGES:**

- Lack of proper analysis of material need.

- Leads to loss due to excessive amount of stock materials.
- Excessive inventory can lead to poor quality goods and degradation.

### 4. A Cloud-Based Inventory Management System Using a Smart Trolley for Automated Billing and Theft Detection

**PROPOSED APPROACH:**

Currently, self-checkout counters contribute about 90% of the supermarkets worldwide. However, there is no system to provide product details while shopping to customers and monitor theft which has led to loss in supermarkets and customer dissatisfaction, respectively. Nowadays, billing is a tedious task as customers have to stand in long queues to get their products barcode scanned. This paper proposes an architecture of a smart trolley which provides an automated billing, anti-theft system, and facilitates inventory management through a web application. This is achieved on a cloud-based platform using RFID and Wi-Fi technology. The goal of this proposed smart trolley system is to provide easily scalable, economical, and technology-oriented shopping system thereby reducing queuing time, anti-theft, and labour cost. Results show that there is a 26% reduction in time spent during shopping as compared to conventional shopping methods

**ADVANTAGES:**

- Reduces theft and improves customer experience.
- Easy to maintain inventory.

**DISADVANTAGES:**

- Its practically difficult because it uses RFIDs so wrong detection is possible.
- Some customers will not prefer self-service.

### 5. An IoT Quality Global Enterprise Inventory Management Model for Automation and Demand Forecasting Based on Cloud

**PROPOSED APPROACH:**

Industrial Internet of Things (IoT) is the applied Internet of Things (IoT) to the manufacturing industry also termed as the Industrial Internet or Industry 4.0. IoT is the next big thing that will be revolutionizing enterprises and factories with focus on return on investment in IoT, all machines in an enterprise as well as the factory are connected to a network and data are collected from the machines. Every machine has so many sensors attached to it, the sensor data can be sent in real time to a cloud storage system through a communication network. It helps to monitor the machines and make machines work efficiently. Data can be stored in a cloud storage service permanently; the collected data can be used for analysing the enterprise inventory management system. Useful information can be taken out of the stored data which can be used for improving the enterprise’s inventory performance globally. It can help the enterprise to reduce losses and increase profits by finding where the performance of the inventory can be improved.

6. **Inventory management system**

**PROPOSED APPROACH:**

This project is aimed at developing a desktop-based application named Inventory Management System for managing the inventory system of any organization. The Inventory Management System (IMS) refers to the system and processes to manage the stock of organization with the involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update theinventory based on the sales details, generate sales and inventory report daily or weekly based. This project is categorized individual aspects for the sales and inventory management system. In this system we are solving different problem affecting to direct sales management and purchase management. Inventory Management System is important to ensure quality control in businesses that handle transactions resolving around consumer goods. Without proper inventory control, a large retail store may runout of stock on an important item. A good inventory management system will alert the wholesaler when it is time to record. Inventory Management System is also on important means of automatically tracking large shipment. An automated Inventory Management System helps to minimize the errors while recording the stock.

**DISADVANTAGES:**

- This application is not suitable for those organization where there is large quantityof product and different level of warehouses

7. **Inventory management system**

**PEOPOSED APPROACH:**

The project has been developed to keep track of details regarding the equipment. The current project is a window based. To provide the basic services related to the supply of the equipment. The project will take care of all supply order.

**DISADVANTAGES:**

- Manual Errors at the time of entering the data can’t be check, only the validation required w.r.t proposed system is checked

8. **Inventory management system**

**PEOPOSED APPROACH:**

Inventory management system is an application which is helpful for business operate. Inventory management is a challenging problem area in supply chain management. Companies need to have inventories in warehouses in order to fulfil customer demand, meanwhile these inventories have holding costs and this is frozen fund that can be lost. Therefore, the task of inventory management is to find the quantity of inventories that will fulfil the demand, avoiding overstocks. This paper presents a case study for the assembling company on inventory management. It is proposed to use inventory management in order to decrease stock levels and toapply an agent system for automation of inventory management processes. Inventory management system (IMS) use for a departmental store.

**DISADVANTAGES:**

- It is difficult to found records due file management system.

## REFERENCE:

**A secure and efficient inventory management system for disasters** | 19 October 2011 | Published by Elsevier Ltd | REFERENCE LINK

**Inventory Management Information System in Blood Transfusion Unit**| 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)| REFERENCE LINK

**A Material Management in Construction Project Using Inventory Management System |** NOV 2019 | IRE Journals | REFERENCE LINK

**A Cloud-Based Inventory Management System Using a Smart Trolley for Automated Billing and Theft Detection| 08 February 2019 | REFERENCE LINK**

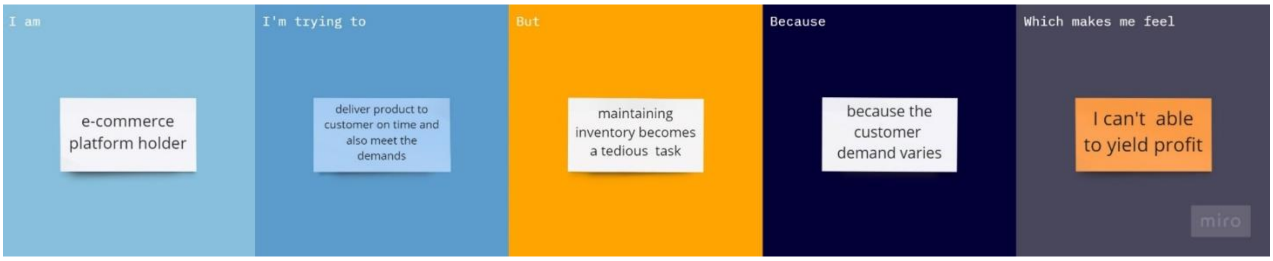
**An IoT Quality Global Enterprise Inventory Management Model for Automation and Demand Forecasting Based on Cloud|** 21 December 2017| IEEE | REFERENCE LINK

**Inventory management system** | Anish Singh, Maharjan, Mandip, Humagain

**Inventory management system** | MS. Dhruvika Patel

**Inventory management system** | Rajkumar,Neelesh kumar singh

## 2.3 PROBLEM STATEMENT DEFINITIONS:

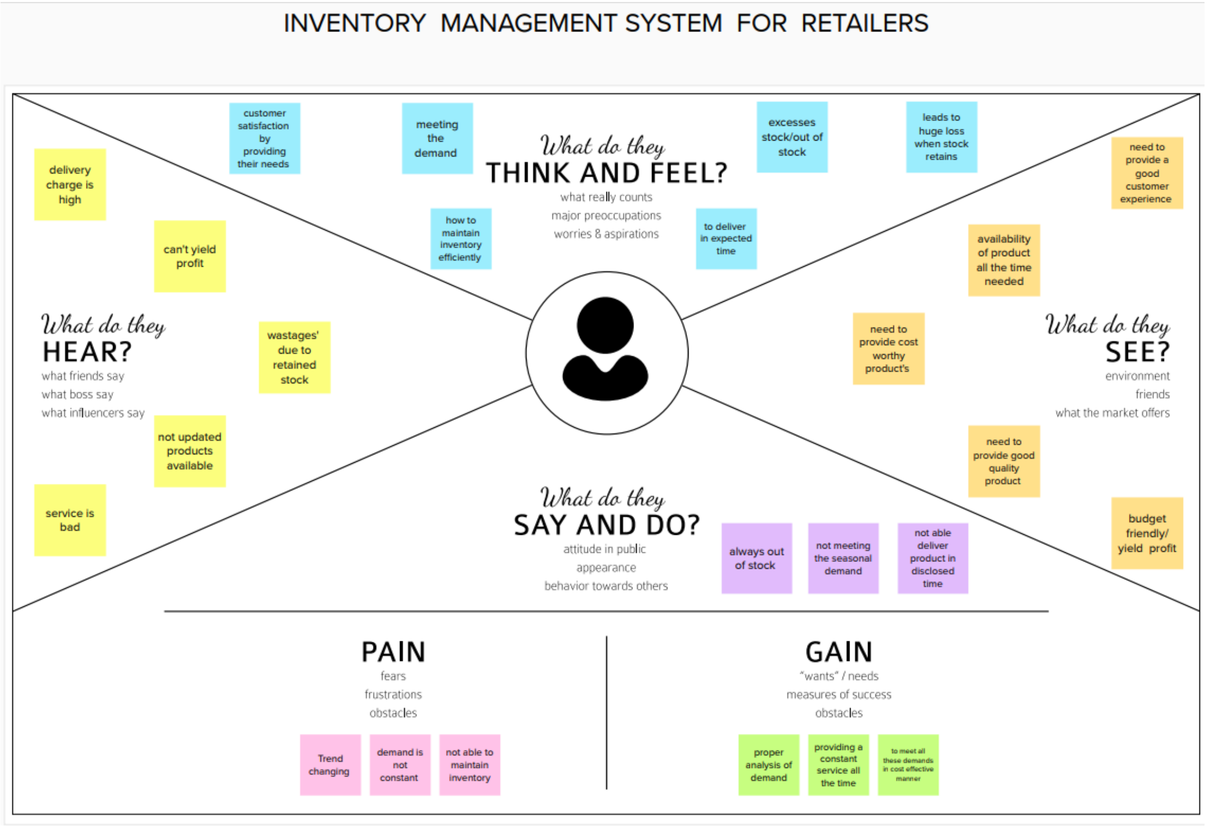


| I am | I'm trying to | But | Because | Which makes me feel |
|---|---|---|---|---|
| e-commerce platform holder | deliver product to customer on time and also meet the demands | maintaining inventory becomes a tedious task | because the customer demand varies | I can't able to yield profit |

| Problem Statement (PS) | I am (Customer) | I'm trying to | But | Because | Which makes me feel |
|---|---|---|---|---|---|
| Over stocking | e-kart | Deliver fresh fruits and vegetable to customer on time | I can't able to deliver fresh products | Because of over stocking and With out a proper stocking we can't able to deliver product on time | I can't able to satisfy customers |
| Proper communication | e-commerce | I try to sell All kind of products including dress, electronic etc. | I can't able to maintain Inventory properly | Because I sell various types of products | I don't have a proper account of my inventory. |
| Locating Hub's | e-commerce | I try sell my products in different region | I can't able to deliver products on time | Because I don't have proper hub location. | I can't able satisfy my customer' |

# IDEATION & PROPOSED SOLUTION

## 3.1 EMPATHY MAP CANVAS:



INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

customer satisfaction by providing their needs

meeting the demand

how to maintain inventory efficiently

to deliver in expected time

excesses stock/out of stock

leads to huge loss when stock retains

What do they
HEAR?
what friends say
what boss say
what influencers say

delivery charge is high

can't yield profit

wastages' due to retained stock

not updated products available

service is bad

What do they
SEE?
environment
friends
what the market offers

need to provide a good customer experience

availability of product all the time needed

need to provide cost worthy product's

need to provide good quality product

budget friendly/ yield profit

What do they
SAY AND DO?
attitude in public
appearance
behavior towards others

always out of stock

not meeting the seasonal demand

not able deliver product in disclosed time

PAIN
fears
frustrations
obstacles

Trend changing

demand is not constant

not able to maintain inventory

GAIN
“wants” / needs
measures of success
obstacles

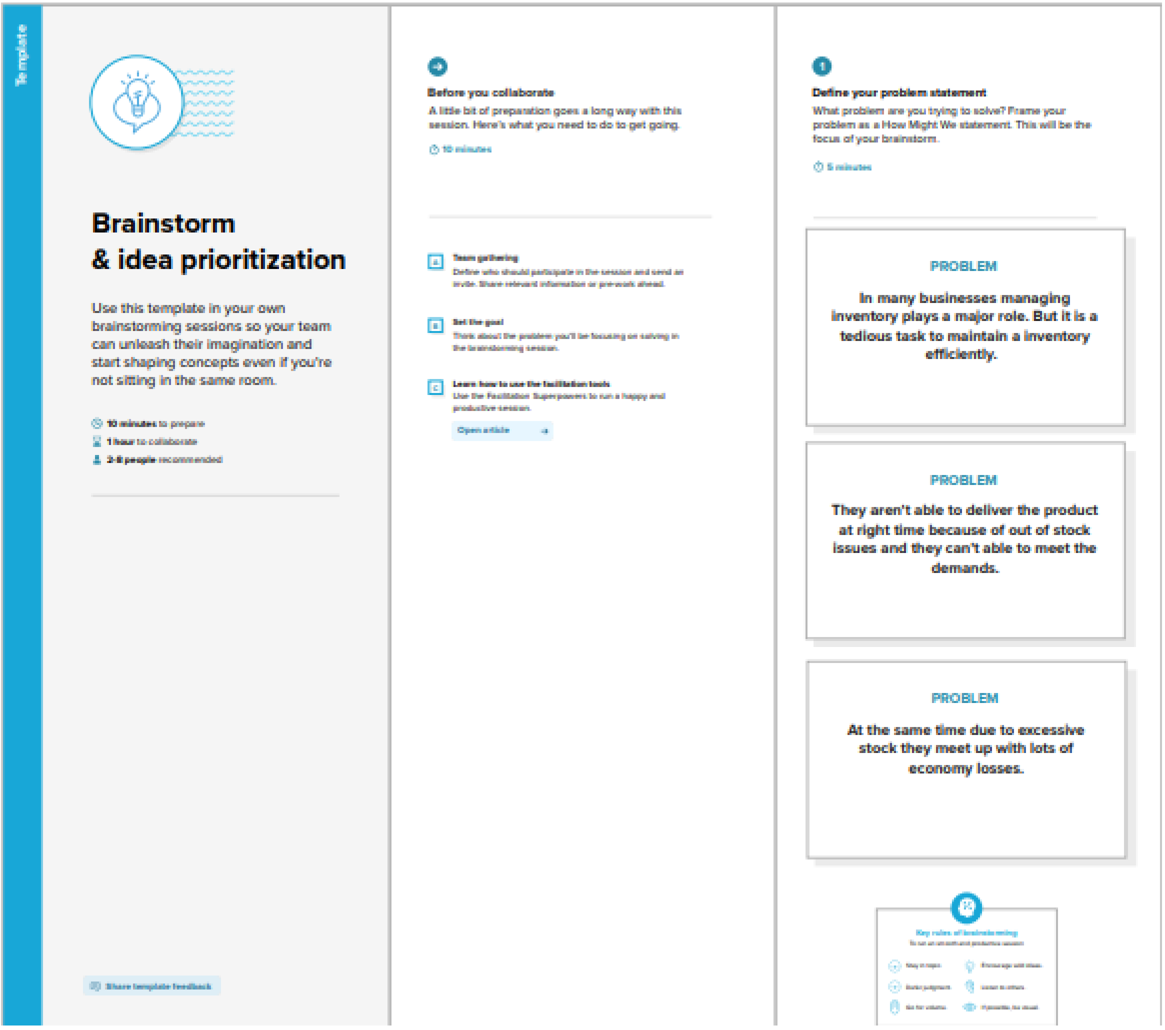
proper analysis of demand

providing a constant service all the time

to meet all these demands in cost effective manner

## 3.2 IDEATION & BRAINSTORMING:

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



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

**In many businesses managing inventory plays a major role. But it is a tedious task to maintain a inventory efficiently.**

PROBLEM

**They aren't able to deliver the product at right time because of out of stock issues and they can't able to meet the demands.**

PROBLEM

**At the same time due to excessive stock they meet up with lots of economy losses.**

Key rules of brainstorming

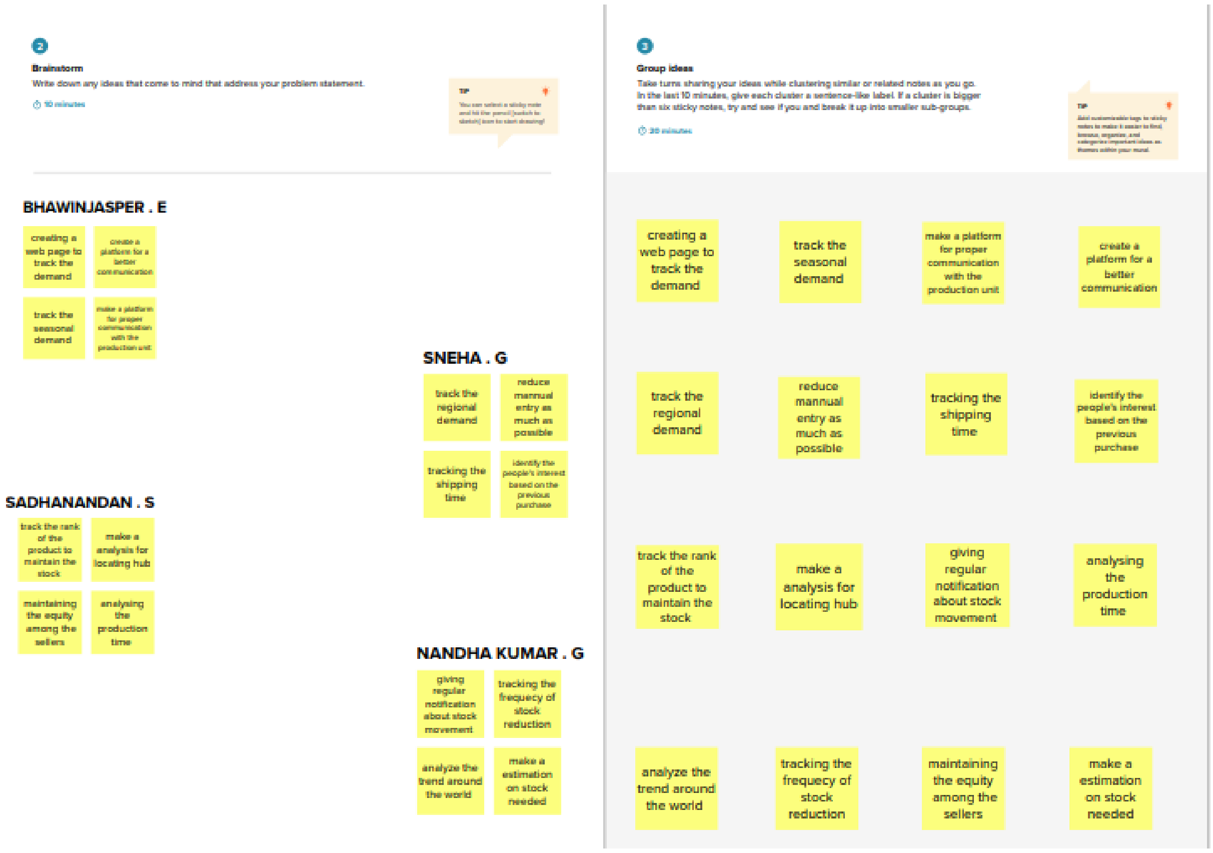
To run an smooth and productive session

Stay in topic. | Encourage wild ideas.

Defer judgment. | Listen to others.

Go for volume. | If possible, be visual.

## Step-2: Brainstorm, Idea Listing and Grouping



2

**Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

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.

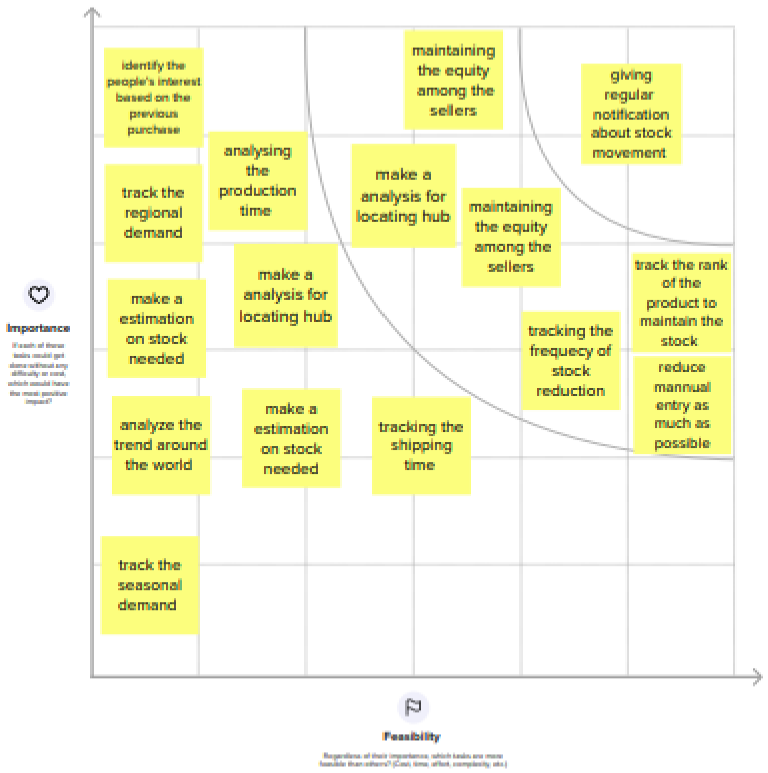
20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

BHAWINJASPER . E
creating a web page to track the demand
create a platform for a better communication
track the seasonal demand
make a platform for proper communication with the production unit
SNEHA . G
track the regional demand
reduce mannual entry as much as possible
tracking the shipping time
identify the people's interest based on the previous purchase
SADHANANDAN . S
track the rank of the product to maintain the stock
make a analysis for locating hub
maintaining the equity among the sellers
analysing the production time
NANDHA KUMAR . G
giving regular notification about stock movement
tracking the frequecy of stock reduction
analyze the trend around the world
make a estimation on stock needed
creating a web page to track the demand
track the seasonal demand
make a platform for proper communication with the production unit
create a platform for a better communication
track the regional demand
reduce mannual entry as much as possible
tracking the shipping time
identify the people's interest based on the previous purchase
track the rank of the product to maintain the stock
make a analysis for locating hub
giving regular notification about stock movement
analysing the production time
analyze the trend around the world
tracking the frequecy of stock reduction
maintaining the equity among the sellers
make a estimation on stock needed

## Step-3: Idea Prioritization



identify the people's interest based on the previous purchase
maintaining the equity among the sellers
giving regular notification about stock movement
analysing the production time
track the regional demand
make a analysis for locating hub
maintaining the equity among the sellers
track the rank of the product to maintain the stock
make a analysis for locating hub
make a estimation on stock needed
tracking the frequecy of stock reduction
reduce mannual entry as much as possible
analyze the trend around the world
make a estimation on stock needed
tracking the shipping time
track the seasonal demand
Importance
Feasibility

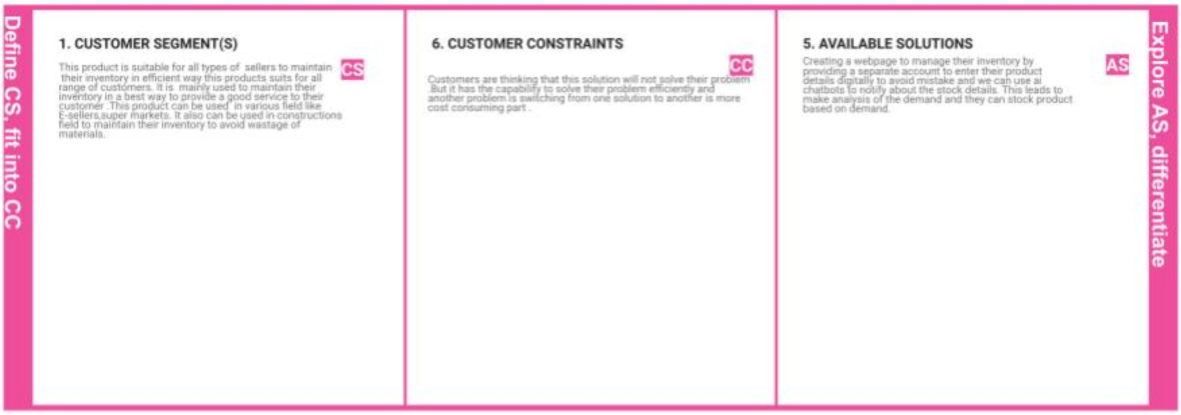
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

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

## 3.3 PROPOSED SOLUTION:

| S.No | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | Many online sellers are facing issue's in maintaining inventory in a efficient way they don't have proper account of the product and the selling ratio of the product and these factors leads to a huge loss. |
| 2. | Idea / Solution description | By providing a platform to maintain a proper account of the product details using that we can analyze the demand of the product and we can give a regular notification regarding the inventory details and we can rank the product according to the sales ratio. |
| 3. | Novelty / Uniqueness | In existing systems they have only the details of the product in inventory and they don't have any proper communication system. But in this we are going to provide a proper communication system regarding the inventory as well as we are going to rank the product based on sales and production time. Through that we are going to give more importance to some products to increase the sales ratio. |
| 4. | Social Impact / Customer Satisfaction | Through this we can able to deliver a product to the customer at disclosed time and we can avoid out of stock issues as well as over stocking issues also. The seller can able to yield more profit. In this we can able to determine the hub locations as well as the demand of the product so that we can reduce the capital invested in inventories. |
| 5. | Business Model (Revenue Model) | Through this solution we can improve customer satisfaction which leads to more profit as well as proper inventory maintenance will reduce the investment cost as well as the wastage of the product will be avoided. Through this we are going to stock only the demanded products. |
| 6. | Scalability of the Solution | This solution is feasible because we have enough technology to implement this solution. We have cloud computing for remote accessing and various message system to notify. |

## 3.4 PROBLEM SOLUTION FIT:


Define CS, fit into CC
CS
CC
AS
Explore AS, differentiate

1. CUSTOMER SEGMENT(S)

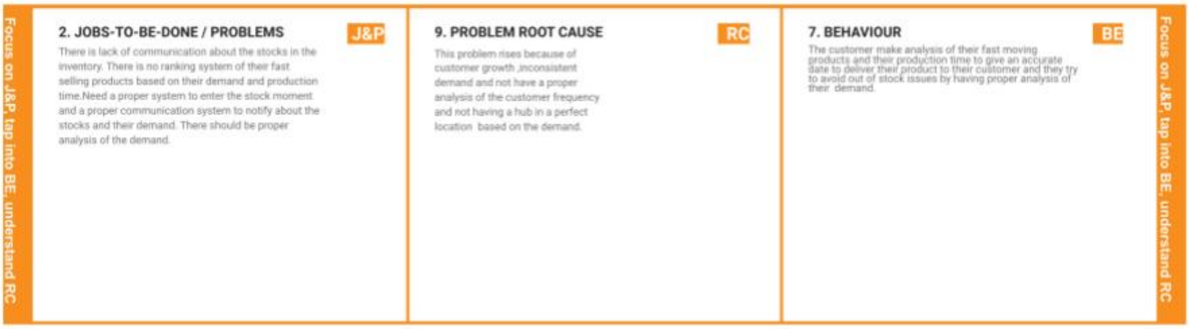
This product is suitable for all types of sellers to maintain their inventory in efficient way this products suits for all range of customers. It is mainly used to maintain their inventory in a best way to provide a good service to their customer .This product can be used in various field like E-sellers,super markets. It also can be used in constructions field to maintain their inventory to avoid wastage of materials.

6. CUSTOMER CONSTRAINTS

Customers are thinking that this solution will not solve their problem .But it has the capabilify to solve their problem efficiently and another problem is switching from one solution to another is more cost consuming part .

5. AVAILABLE SOLUTIONS

Creating a webpage to manage their inventory by providing a separate account to enter their product details digitally to avoid mistake and we can use ai chatbots to notify about the stock details. This leads to make analysis of the demand and they can stock product based on demand.


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

2. JOBS-TO-BE-DONE / PROBLEMS

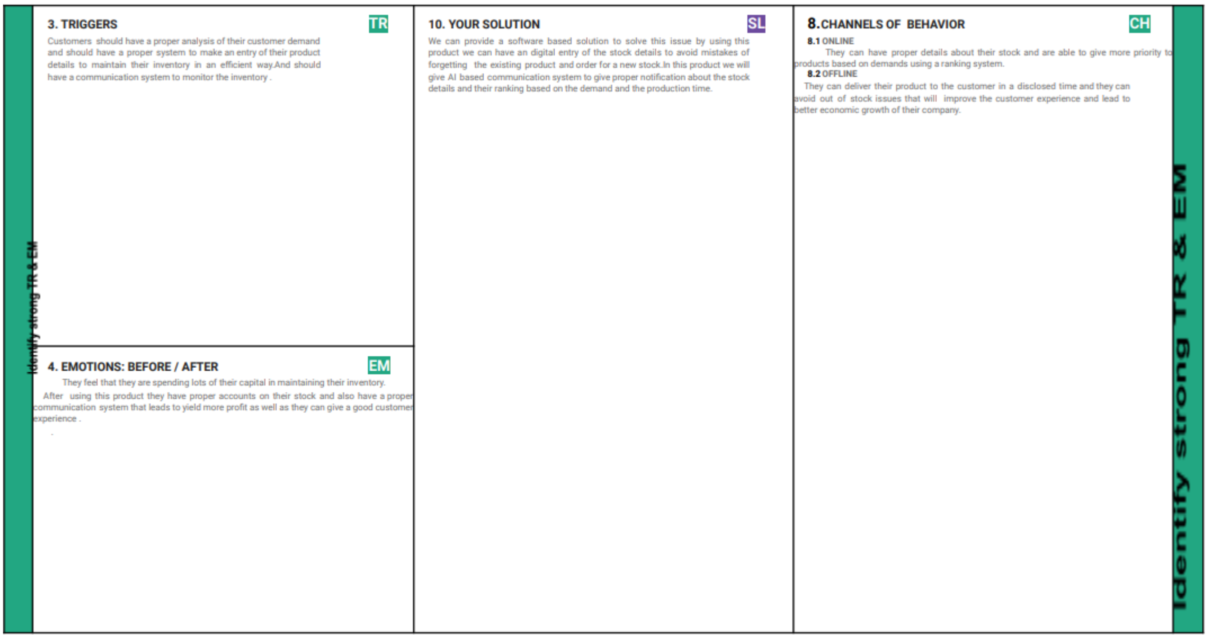
There is lack of communication about the stocks in the inventory. There is no ranking system of their fast selling products based on their demand and production time.Need a proper system to enter the stock moment and a proper communication system to notify about the stocks and their demand. There should be proper analysis of the demand.

9. PROBLEM ROOT CAUSE

This problem rises because of customer growth ,inconsistent demand and not have a proper analysis of the customer frequency and not having a hub in a perfect location based on the demand.

7. BEHAVIOUR

The customer make analysis of their fast moving products and their production time to give an accurate date to deliver their product to their customer and they try to avoid out of stock issues by having proper analysis of their demand.


Identify strong TR & EM
TR
EM
SL
CH
Identify strong TR & EM

3. TRIGGERS

Customers should have a proper analysis of their customer demand and should have a proper system to make an entry of their product details to maintain their inventory in an efficient way.And should have a communication system to monitor the inventory .

4. EMOTIONS: BEFORE / AFTER

They feel that they are spending lots of their capital in maintaining their inventory.

After using this product they have proper accounts on their stock and also have a proper communication system that leads to yield more profit as well as they can give a good customer experience .

10. YOUR SOLUTION

We can provide a software based solution to solve this issue by using this product we can have an digital entry of the stock details to avoid mistakes of forgetting the existing product and order for a new stock.In this product we will give AI based communication system to give proper notification about the stock details and their ranking based on the demand and the production time.

8.CHANNELS OF BEHAVIOR

8.1 ONLINE

They can have proper details about their stock and are able to give more priority to products based on demands using a ranking system.

8.2 OFFLINE

They can deliver their product to the customer in a disclosed time and they can avoid out of stock issues that will improve the customer experience and lead to better economic growth of their company.

# REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENT:

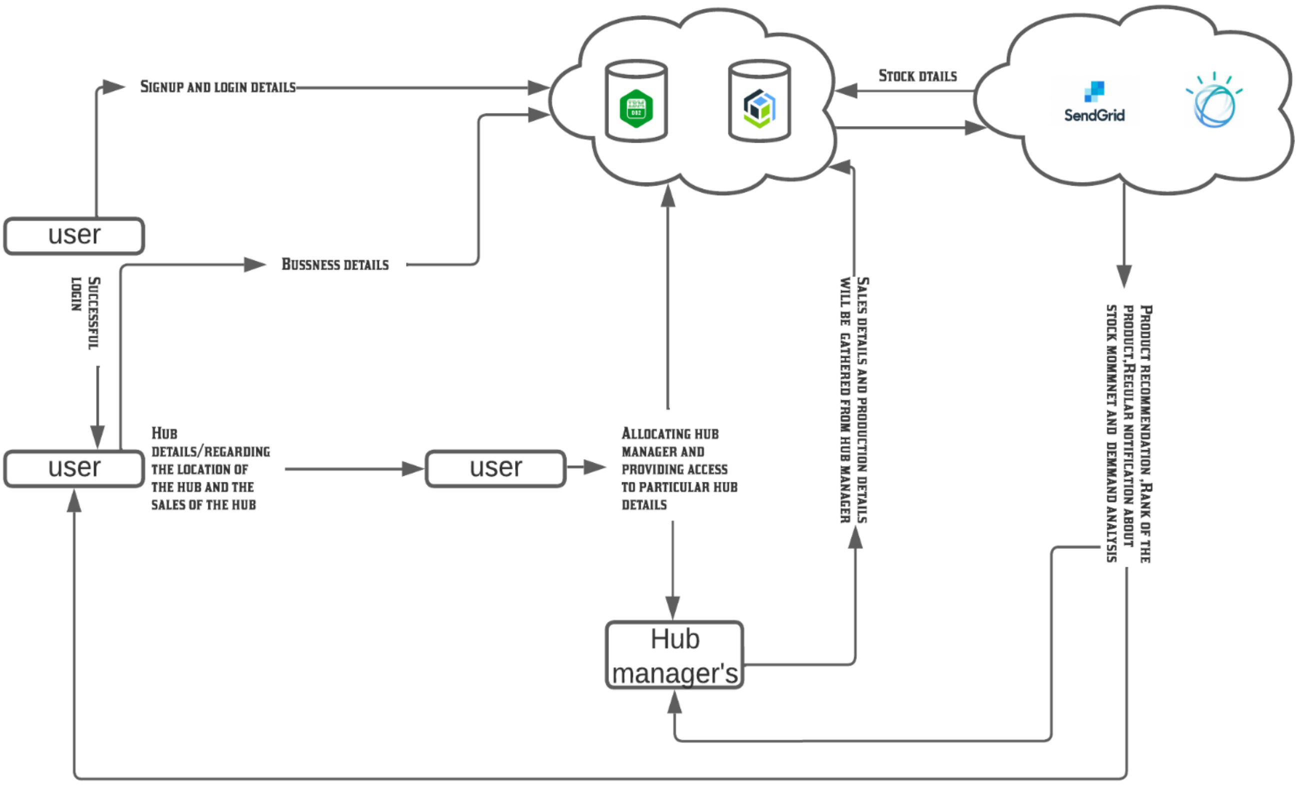
| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration using Business e-mail |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | Account Completion | Filling the details required to maintain an inventory |
| FR-4 | Allocating hub manager | Providing restricted access to the hub managers to maintain entire hub's in a single login using access policy. |
| FR-5 | Analysis of demand /stock moment | Regular communication regarding stock moment to the user. Using sendgrid mail service and also using IBM Watson Ai Chat Bot. |

## 4.2 NON FUNCTIONALREQUIREMENT:

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | It avoid manual entry so mistakes can be avoided and it also gives us regular notification about the stock moment because of that we can able to maintain inventory in an efficient way. |
| NFR-2 | **Security** | Here we are providing two step authentication to provide a better security as well the owner of the account has full access to their accounts he/she can restrict other user this will provide better security because each and every action will be monitored. |
| NFR-3 | **Reliability** | The product is going to be developed in microservice architecture so the complexity of the software will be reduced and it will improve the performance. each and every module will be loosely coupled so failure of one module will not lead to failure of entire system. |
| NFR-4 | **Performance** | Hence we are going to develop the product in microservice architecture each and every module is going to have their own environment so traffic in one module will not affect other because of that we can able to provide a smooth service to our customer. |

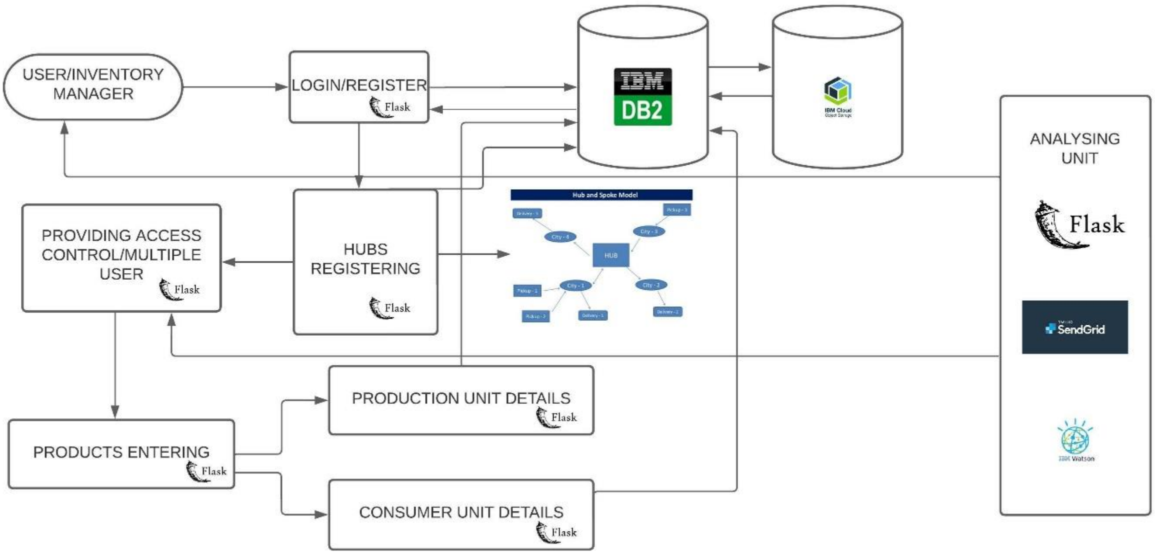
# PROJECT DESIGN

## 5.1 DATA FLOW DIAGRAMS:

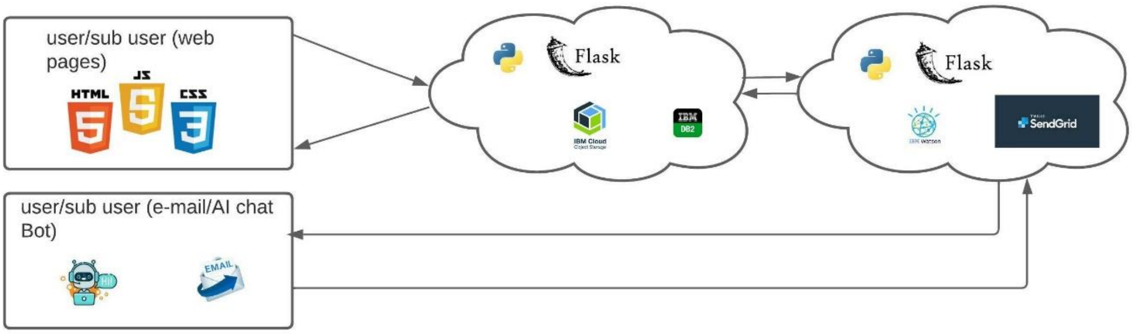

Signup and login details
user
Bussness details
Successful login
Stock dtails
SendGrid
user
Hub details/regarding the location of the hub and the sales of the hub
user
Allocating hub manager and providing access to particular hub details
Sales details and production details will be gathered from hub manager
Product recommendation ,Rank of the product,Regular notification about stock mommnet and demmand analysis
Hub manager's

## 5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

### SOLUTION ARCHITECTURE:



### TECHNICAL ARCHITECTURE:



## 5.3 USER STORIES

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| User (web browser) | Registration | USN -1 | User can create an account by providing business mail id and password | I can access my account for maintaining my inventory | High | Sprint -1 |
| User (web browser) | Registration /Login | USN -2 | Two step authentication using one time password to provide mail id or phone number | I can enhance security level of my account | High | Sprint -2 |
| User (web browser) | Login | USN -3 | Using registered mail Id | To access the dashboard | High | Sprint-1 |
| User (web browser) | Main dashboard | USN -4 | User need to complete account settings like giving the details about their inventory and their branches | I can able to split and access the inventories and I don't have any restrictions to change my account details | High | Sprint -1 |
| User (web browser) | Hub maintenance | USN -5 | User can able to create a separate account for individual hub and he can able to create access policy to share their account with their hub managers | I can allot hub mangers using this features to access particular data from my account | High | Sprint -2 |
| Sub user (web browser) | Hub dashboard login | USN -6 | Hub mangers can able to login to the account to access their allotted hub details | I can able to manage Entire hub's at one login | High | Sprint -3 |
| Sub user (web browser) | Hub dashboard | USN -7 | Hub mangers can able to add product details and production details. They can also provide access to their allotted space to others. | I can able to manage entire system at one place. | High | Sprint -3 |
| User & Sub user (web browser, AI chat bot , e-mail) | Communication system | USN -8 | User and hub mangers can get the details of the stock moment via mail or chat bot . | I can able to maintain my inventory efficiently without facing much economic loss | Medium | Sprint -4 |

# PROJECT PLANNING & SCHEDULING

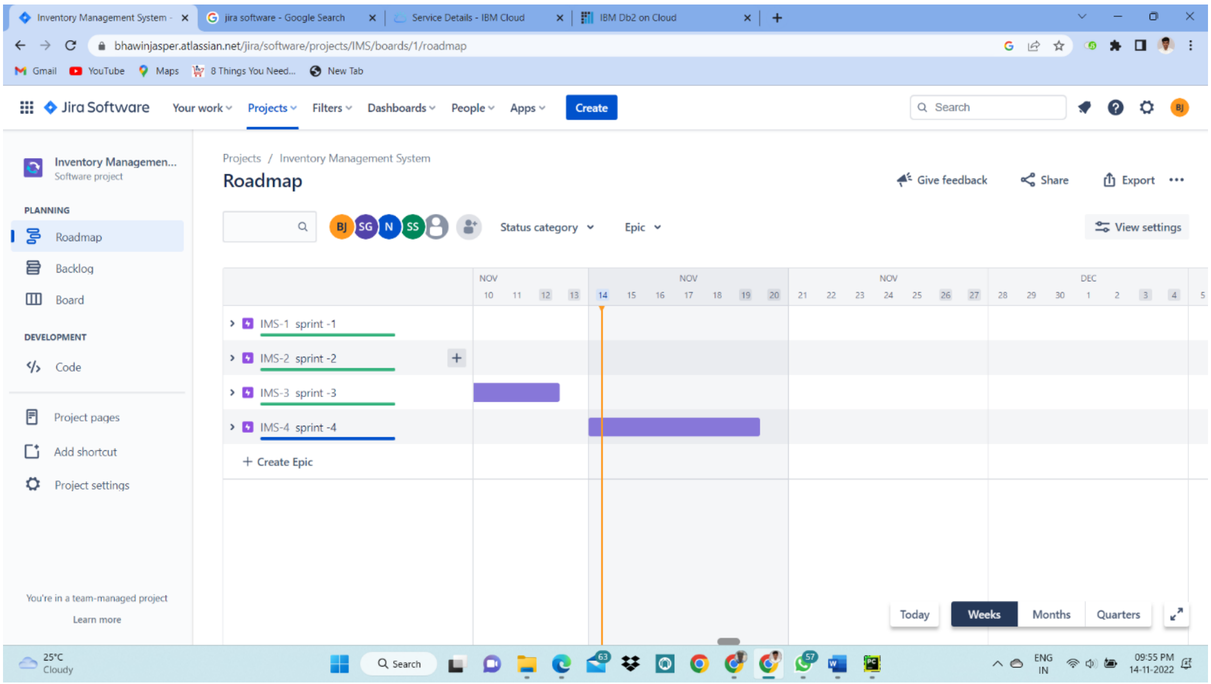
## 6.1 SPRINT PLANNING & ESTIMATION:

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Registration | USN -1 | User can create an account by providing business mail id and password | 5 | High | BHAWINJASPER E,NANDHA KUMAR G,SNEHA G ,SADHANANDAN S |
| Sprint-2 | Registration /Login | USN -2 | Two step authentication using one time password to provide mail id or phone number | 10 | High | BHAWINJASPER E,NANDHA KUMAR G,SNEHA G ,SADHANANDAN S |
| Sprint-1 | Login | USN -3 | Using registered mail Id | 5 | High | BHAWINJASPER E,NANDHA KUMAR G,SNEHA G ,SADHANANDAN S |
| Sprint-1 | Main dashboard | USN -4 | User need to complete account settings like giving the details about their inventory and their branches | 10 | High | BHAWINJASPER E,NANDHA KUMAR G,SNEHA G ,SADHANANDAN S |
| Sprint-2 | Hub maintenance | USN -5 | User can able to create a separate account for individual hub and he can able to create access policy to share their account with their hub managers | 10 | High | BHAWINJASPER E,NANDHA KUMAR G,SNEHA G ,SADHANANDAN S |
| Sprint-3 | Hub dashboard login | USN -6 | Hub mangers can able to login to the account to access their allotted hub details | 10 | High | BHAWINJASPER E,NANDHA KUMAR G,SNEHA G ,SADHANANDAN S |
| Sprint-3 | Hub dashboard | USN -7 | Hub mangers can able to add product details and production details. They can also provide access to their allotted space to others. | 10 | High | BHAWINJASPER E,NANDHA KUMAR G,SNEHA G ,SADHANANDAN S |
| Sprint-4 | Communication system | USN -8 | User and hub mangers can get the details of the stock moment via mail or chat bot . | 20 | Medium | BHAWINJASPER E,NANDHA KUMAR G,SNEHA G ,SADHANANDAN S |

## 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 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

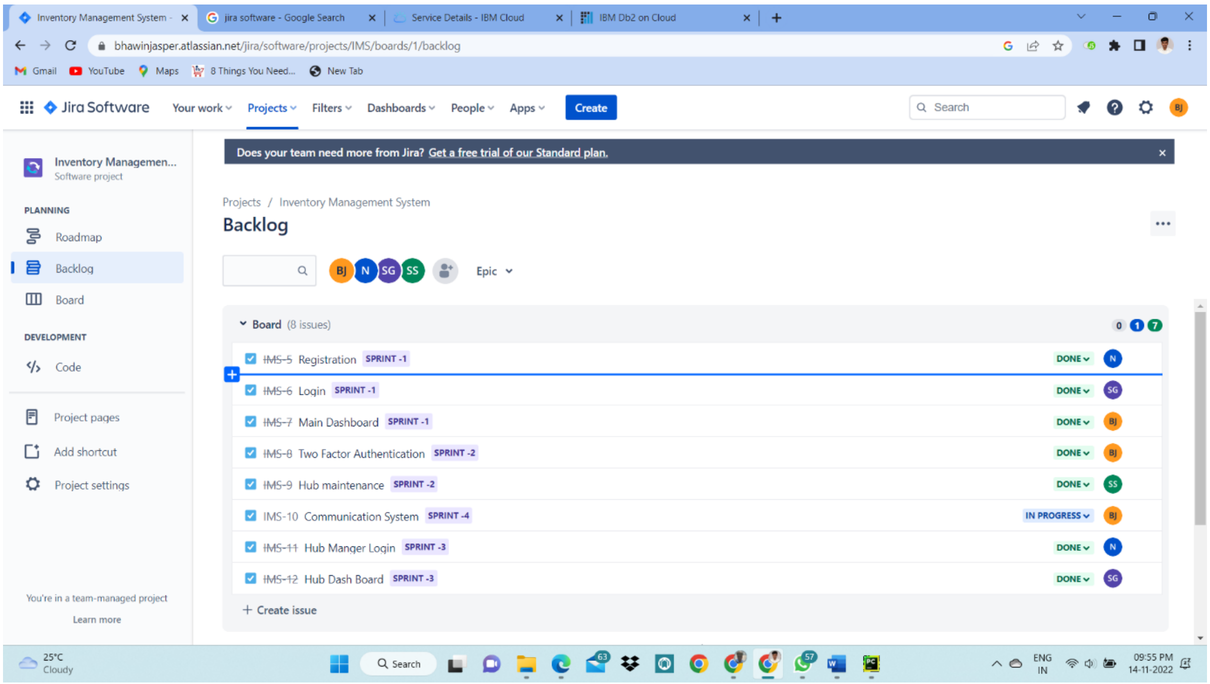
## 6.3 REPORTS FROM JIRA:


bhawinjasper.atlassian.net/jira/software/projects/IMS/boards/1/roadmap

Projects / Inventory Management System

Roadmap

- IMS-1 sprint -1
- IMS-2 sprint -2
- IMS-3 sprint -3
- IMS-4 sprint -4



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / Inventory Management System

Backlog

BJ N SG SS Epic

Board (8 issues) 0 1 7

☑ ~~IMS-5~~ Registration SPRINT -1 DONE N

☑ ~~IMS-6~~ Login SPRINT -1 DONE SG

☑ ~~IMS-7~~ Main Dashboard SPRINT -1 DONE BJ

☑ ~~IMS-8~~ Two Factor Authentication SPRINT -2 DONE BJ

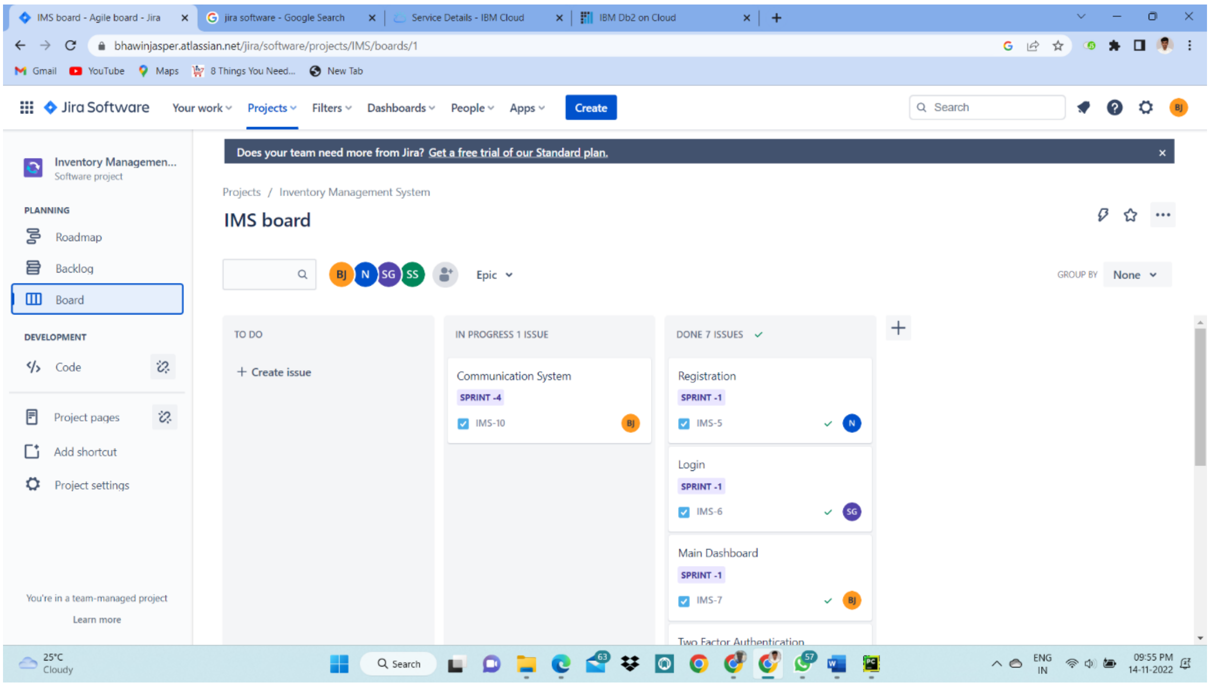
☑ ~~IMS-9~~ Hub maintenance SPRINT -2 DONE SS

☑ IMS-10 Communication System SPRINT -4 IN PROGRESS BJ

☑ ~~IMS-11~~ Hub Manger Login SPRINT -3 DONE N

☑ ~~IMS-12~~ Hub Dash Board SPRINT -3 DONE SG

+ Create issue



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / Inventory Management System

IMS board

BJ N SG SS Epic GROUP BY None

TO DO

+ Create issue

IN PROGRESS 1 ISSUE

Communication System
SPRINT -4
☑ IMS-10 BJ

DONE 7 ISSUES ✓

Registration
SPRINT -1
☑ IMS-5 ✓ N

Login
SPRINT -1
☑ IMS-6 ✓ SG

Main Dashboard
SPRINT -1
☑ IMS-7 ✓ BJ

Two Factor Authentication

# CODING & SOLUTIONING

## 7.1 FEATURE 1:

In this we created login page and registration page for creating a new account for managing their inventory. Fields for registration are username, mail id, phone number, otp ,password, re-enter the password .The username should be unique and with one mail id we can able to create only one account. Fields for login page username, mail id, otp, hub user check box.

Hub user check box is used for manger login the manager allocation will be took place inside the hub dash board by the owner of the account.

After login user can able to see the home page with a top navigation bar. In home page user can able to see the profit growth graph, sales growth bar, customer base pie chart. In navigation bar there is profile settings, Analysis page ,Ranking page, Hub entry page, Hub dashboard page.

In analysis page user can able to see the sales growth, profit growth, customer base hub wise as well as product wise.

In ranking page user can able to rank the product as well as the hub based the analysis details.

In hub entry page we can able to add hub details, supplier details, product details.

In hub dash board we can able to create account for manager and also allocate hub.

And also we can able to list and edit the product details and supplier details.

Source code for login page: login page

Source code for registration page: registration page

## 7.2 FEATURE 2:

In this we provided a communication system to alert the user about the out of stock based on the user setting regarding the threshold, and also we provided a communication regarding the hub manger allocation.

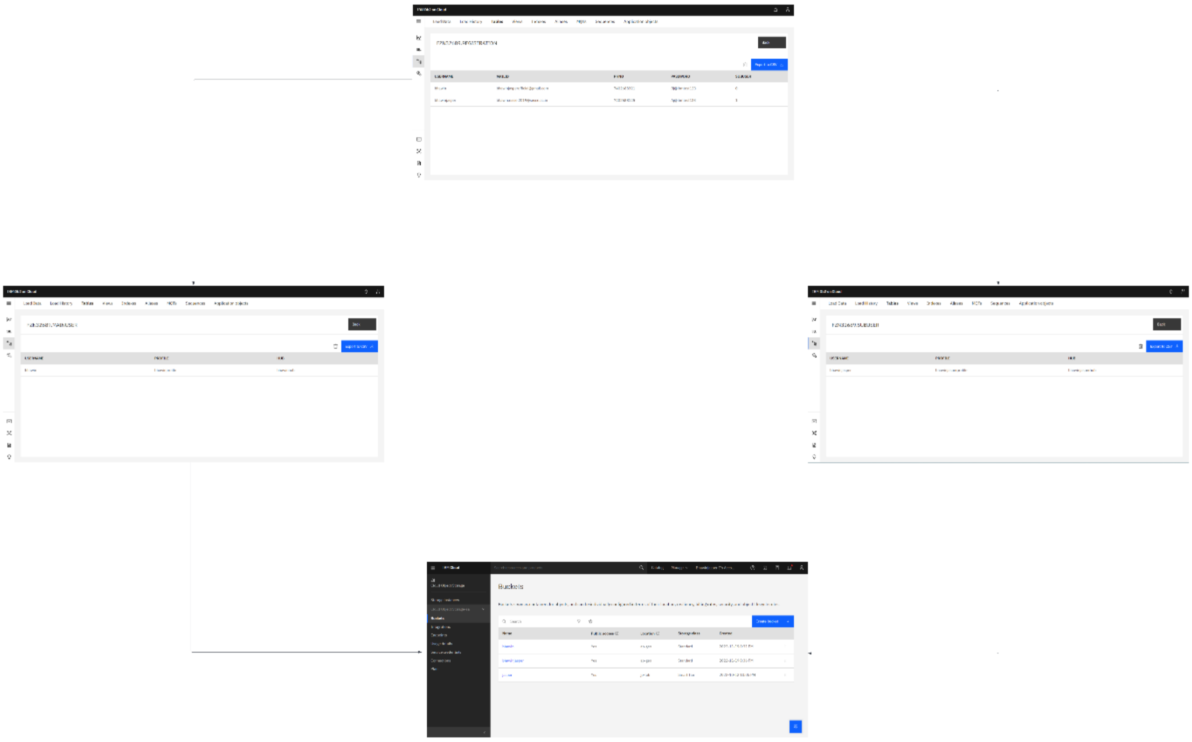
And user can able to add sales details in hub dashboard and also can able to edit the supplier details as well as the product details in the hub dash board.

Source code: hub user

Source code: main user

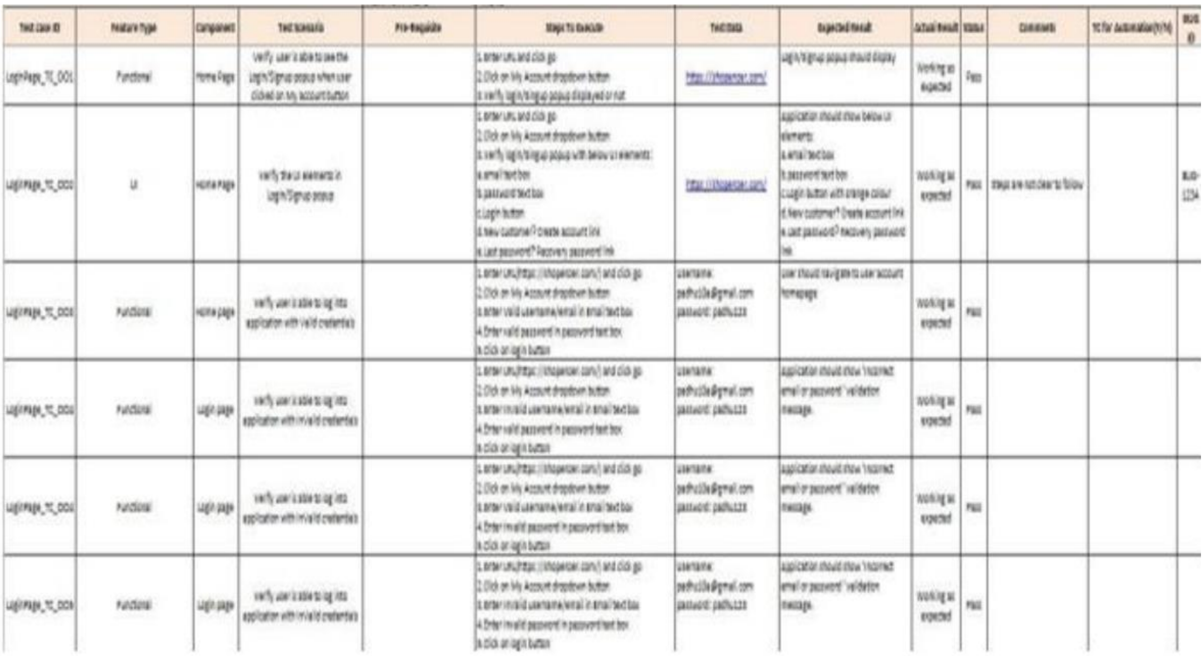
## 7.3 DATABASE SCHEMA:

Here we used IBM DB2 for storing registration details and for storing hub dash board details we used IBM OBJECT STORAGE  were we store files in json format.



# TESTING

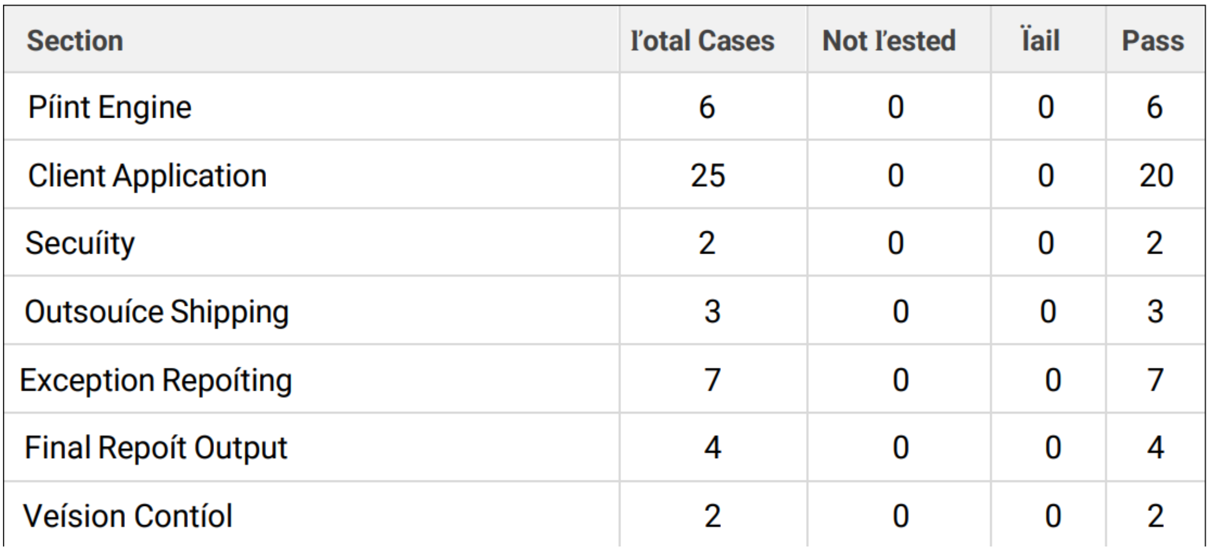
## 8.1 TEST CASES:



| Test Case ID | Feature Type | Component | Test Scenario | Pre-Requisite | Steps To Execute | Test Data | Expected Result | Actual Result | Status | Comments | TC for Automation(Y/N) | BUG ID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LoginPage_TC_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<br>2.Click on My Account dropdown button<br>3.verify login/signup popup displayed or not | https://shopenzer.com/ | Login/Signup popup should display | Working as expected | Pass |  |  |  |
| LoginPage_TC_00[illegible] | UI | Home Page | Verify the ui elements in Login/Signup popup |  | 1.enter url and click go<br>2.Click on My Account dropdown button<br>3.verify login/signup popup with below ui elements:<br>a.email text box<br>b.password text box<br>c.Login button<br>d.New customer? Create account link<br>e.Lost password? Recovery password link | https://shopenzer.com/ | application should show below ui elements:<br>a.email text box<br>b.password text box<br>c.Login button with orange colour<br>d.New customer? Create account link<br>e.Lost password? recovery password link | Working as expected | Pass | steps are not clear to follow |  | BUG-1234 |
| LoginPage_TC_00[illegible] | Functional | Home page | verify user is able to log into application with Valid credentials |  | 1.enter url(https://shopenzer.com/) and click go<br>2.Click on My Account dropdown button<br>3.enter valid username/email in email text box<br>4.Enter valid password in password text box<br>5.click on login button | username: padhu03a@gmail.com<br>password: padhu123 | user should navigate to user account homepage | Working as expected | Pass |  |  |  |
| LoginPage_TC_00[illegible] | Functional | Login page | verify user is able to log into application with invalid credentials |  | 1.enter url(https://shopenzer.com/) and click go<br>2.Click on My Account dropdown button<br>3.enter invalid username/email in email text box<br>4.Enter valid password in password text box<br>5.click on login button | username: padhu03a@gmail.com<br>password: padhu123 | application should show 'incorrect email or password ' validation message | Working as expected | Pass |  |  |  |
| LoginPage_TC_00[illegible] | Functional | Login page | verify user is able to log into application with invalid credentials |  | 1.enter url(https://shopenzer.com/) and click go<br>2.Click on My Account dropdown button<br>3.enter valid username/email in email text box<br>4.Enter invalid password in password text box<br>5.click on login button | username: padhu03a@gmail.com<br>password: padhu123 | application should show 'incorrect email or password ' validation message | Working as expected | Pass |  |  |  |
| LoginPage_TC_00[illegible] | Functional | Login page | verify user is able to log into application with invalid credentials |  | 1.enter url(https://shopenzer.com/) and click go<br>2.Click on My Account dropdown button<br>3.enter invalid username/email in email text box<br>4.Enter invalid password in password text box<br>5.click on login button | username: padhu03a@gmail.com<br>password: padhu123 | application should show 'incorrect email or password ' validation message | Working as expected | Pass |  |  |  |

## 8.2 USER ACCEPTANCE TESTING:

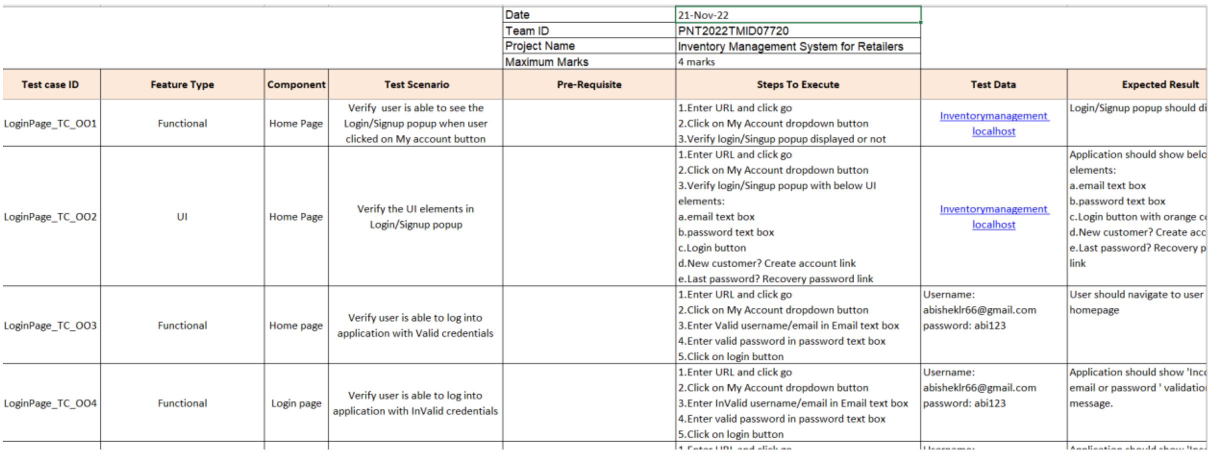
| Resolution | Seveíity 1 | Seveíity 2 | Seveíity 3 | Seveíity4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 10 | 4 | 2 | 3 | 19 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| Exteínal | 2 | 3 | 0 | 1 | 6 |
| Fixed | 11 | 2 | 4 | 20 | 37 |
| Not Repíoduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 |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



| Section | ľotal Cases | Not ľested | Ïail | Pass |
|---|---|---|---|---|
| Píint Engine | 6 | 0 | 0 | 6 |
| Client Application | 25 | 0 | 0 | 20 |
| Secuíity | 2 | 0 | 0 | 2 |
| Outsouíce Shipping | 3 | 0 | 0 | 3 |
| Exception Repoíting | 7 | 0 | 0 | 7 |
| Final Repoít Output | 4 | 0 | 0 | 4 |
| Veísion Contíol | 2 | 0 | 0 | 2 |

# RESULTS

## 9.1 PERFORMANCE METRICS:



| | |
|---|---|
| Date | 21-Nov-22 |
| Team ID | PNT2022TMID07720 |
| Project Name | Inventory Management System for Retailers |
| Maximum Marks | 4 marks |

| Test case ID | Feature Type | Component | Test Scenario | Pre-Requisite | Steps To Execute | Test Data | Expected Result |
|---|---|---|---|---|---|---|---|
| LoginPage_TC_OO1 | Functional | Home Page | Verify user is able to see the Login/Signup popup when user clicked on My account button | | 1.Enter URL and click go<br>2.Click on My Account dropdown button<br>3.Verify login/Singup popup displayed or not | Inventorymanagement localhost | Login/Signup popup should di |
| LoginPage_TC_OO2 | UI | Home Page | Verify the UI elements in Login/Signup popup | | 1.Enter URL and click go<br>2.Click on My Account dropdown button<br>3.Verify login/Singup popup with below UI elements:<br>a.email text box<br>b.password text box<br>c.Login button<br>d.New customer? Create account link<br>e.Last password? Recovery password link | Inventorymanagement localhost | Application should show belo elements:<br>a.email text box<br>b.password text box<br>c.Login button with orange c<br>d.New customer? Create acc<br>e.Last password? Recovery p link |
| LoginPage_TC_OO3 | Functional | Home page | Verify user is able to log into application with Valid credentials | | 1.Enter URL and click go<br>2.Click on My Account dropdown button<br>3.Enter Valid username/email in Email text box<br>4.Enter valid password in password text box<br>5.Click on login button | Username:<br>abisheklr66@gmail.com<br>password: abi123 | User should navigate to user homepage |
| LoginPage_TC_OO4 | Functional | Login page | Verify user is able to log into application with InValid credentials | | 1.Enter URL and click go<br>2.Click on My Account dropdown button<br>3.Enter InValid username/email in Email text box<br>4.Enter valid password in password text box<br>5.Click on login button | Username:<br>abisheklr66@gmail.com<br>password: abi123 | Application should show 'Inc email or password ' validatio message. |

# ADVANTAGES & DISADVANTAGES

**ADVANTAGES:**

1. **It helps to maintain the right amount of stocks:** contrary to the belief that is held by some people, inventory management does not seek to reduce the amount of inventory that you have in stock, however, it seeks to maintain an equilibrium point where your inventory is working at a maximum efficiency and you do not have to have many stocks or too few stocks at hand at any particular point in time. The goal is to find that zone where you are never losing money in your inventory in either direction. With the aid of an efficient inventory management strategy, it is easy to improve the accuracy of inventory order.
2. **It leads to a more organized warehouse:** with the aid of a good inventory management system, you can easily organize your warehouse. If your warehouse is not organized, you will find it very difficult to manage your inventory. A lot of businesses choose to optimize their warehouse by putting the items that have the highest sales together in a place that is easy to access in the warehouse. This ultimately helps to speed up order fulfilment and keeps clients happy.

**DISADVANTAGES:**

1. **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. In instances where inventory control is in-house, this includes the number of new hires that are not present to regulate the warehouse and facilitate transactions. In instances where the inventory management is in the hands of a third party, the cost is a subscription price and a dependence on another separate company to manage its infrastructure. No matter the choice you go for, it translates to a higher overhead cost and more layers of management between the owner and the customer. From the view point of the customer, a problem that requires senior management to handle will take a longer period of time before it will be trashed out.
2. **Impersonal touch:** another disadvantage of inventory management is a lack of personal touch. Large supply chain management systems make products more accessible across the globe and most provide customer service support in case of difficulty, but the increase in infrastructure can often mean a decrease in the personal touch that helps a company to stand out above the rest. For instance, the sales manager of a small manufacturing company that sells plumbing supplies to local plumbers can throw in an extra box of washers or elbows at no charge to the customer without raising any alarms. This is done for the sake of customer relations and often makes the customer feel like he is special. While free materials can also be provided under inventory

management, processing time and paper work make obtaining the material feel more like a chore for the customer or even an entitlement.

# CONCLUSION

Inventory management is a very complex but essential part of the supply chain. An effective inventory management system helps to reduce stock-related costs such as warehousing, carrying, and ordering costs. There are different techniques that businesses can utilize to simplify and optimize stock management processes and control systems.

# FUTURE SCOPE

The scope of an inventory system can cover many needs, including valuing the inventory, measuring the change in inventory and planning for future inventory levels. The value of the inventory at the end of each period provides a basis for financial reporting on the balance sheet. Measuring the change in inventory allows the company to determine the cost of inventory sold during the period. This allows the company to plan for future inventory needs.

The limitations of the periodic system include not knowing an exact inventory count in the middle of the period and running the risk of stockouts. With the periodic system, the company knows the inventory level with certainty only when it physically counts the inventory at the end of each period. Throughout the period, the company takes customer orders without knowing the exact inventory count or whether enough products are available to meet customer demand.

# APPENDIX

**SOURCE CODE:**

Source code: source code

Git Hub: https://github.com/IBM-EPBL/IBM-Project-27288-1660053196

Project Demo Link: https://youtu.be/4uRBmTwVlzE

## LOGIN PAGE FOR MAIN USER & HUB USER:

This page will provide login page for **owner** of the account. As well as for the **hub user** that is manager for certain hub’s.

### TEMPLATE:

### LOGIN.HTML

This is for UI.

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head>
    <meta charset="UTF-8">
    <title>Inventory management login page</title>
      <meta name="viewport" content="width=device-width, initial-scale=1">
  <link              href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>
  <script>
     var ur="";
       function checker1(value) {

      const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
          document.getElementById("usr").innerHTML=this.responseText;
       }
```

```
xhttp.open("POST", "/uservalidate");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
   }
   function checker2(value) {
     ur=document.getElementById("user").value;
    const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("psw").innerHTML=this.responseText;
        if( document.getElementById("psw").innerHTML==""){

         var x = document.createElement("INPUT");
        var y=document.createElement("br")
        var z=document.createElement("p")
        const node = document.createTextNode(".");
         z.appendChild(node);
          z.setAttribute("id","otpp")
          x.setAttribute("type", "password");
          x.setAttribute("placeholder","enter 4 digit otp")
          x.setAttribute("class","form-control")
          x.setAttribute("id","otp")
         document.getElementById("mail").appendChild(x);
         document.getElementById("mail").appendChild(y);
         document.getElementById("mail").appendChild(y);
         document.getElementById("mail").appendChild(z);
         document.getElementById("mail").appendChild(y);
          var z1=document.getElementById("otp")
          z1.addEventListener("focusout",function(){
           const xhttp = new XMLHttpRequest();
     xhttp.onload = function() {
        document.getElementById("otpp").innerHTML=this.responseText;
        if( document.getElementById("otpp").innerHTML=="otp matched"){
          document.getElementById("otpp").style.color="green";
        }else{
          document.getElementById("otpp").style.color="red";
        }
      }
  xhttp.open("POST", "/verifyotp");
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

        xhttp.send("mail="+ur+"&"+"otp="+this.value);
          });
        }


      }
     xhttp.open("POST", "/passwordvalidate");
     xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
     xhttp.send("pass="+value+"&name="+ur);
    }

    function ok(){

      if( document.getElementById("usr").innerHTML!=""){

        document.getElementById("usr").focus();

        return false;
        }
        if( document.getElementById("psw").innerHTML!=""){

        document.getElementById("psw").focus();

          return false;
        }
        if( document.getElementById("otpp").innerHTML!="otp matched"){

        document.getElementById("otpp").focus();

          return false;
        }

        return true;
      }
  </script>
<body>
<div class="container mt-3" >
   <h3>LOGIN</h3>
<form  onsubmit="return ok()" action="/validate" method="post" >
   <input id="user" class="form-control" type="text" placeholder="username"
onfocusout="checker1(this.value)" name="usernm"required ><p id="usr"
style="color:red">.</p><br><br>
   <div id="mail">
     <input class="form-control" type="password" placeholder="Enter your password"
onfocusout="checker2(this.value)" name="pass" required><p id="psw"
style="color:red">.</p><br><br>
   </div>
   <input  type="checkbox" value="1" name="sub"> HUB USER </input> <br><br>
   <input  class="btn btn-primary" type="submit" value="submit"><br><br>
```

```
    <a href="http://127.0.0.1:5001/mainregister">Don't have an account ? register here...</a>
</form>
</div>


</body>
</html>
```

**BUSINESS LOGIC:**

**This part provide business logic for login. It includes FLASK API, IBM DB API, SENDGRID API.**

**MAIN.PY:**

```
from flask import Flask,redirect,render_template,request


import ibm_db
import TwoStepAuthenticator
con=True
app=Flask(__name__)


otp={}
@app.route("/")
def Login():
    return render_template("Login.html")
@app.route("/validate" ,methods=["POST"])
def validate():
    print(type(request.form))
    usernm=request.form["usernm"]
    password=request.form["pass"]


    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7Yk
TR3uKK3a",
            '', '')
        try:
         qry=f"SELECT  * FROM FZN32689.REGISTRATION"
         stmt=ibm_db.exec_immediate(con,qry)
         result=ibm_db.fetch_both(stmt)
         while result!=False:
            if request.form.__contains__("sub"):
                if result["USERNAME"] == usernm and result["PASSWORD"] == password and
result["SUBUSER"] == 1:
                    return redirect("http://127.0.0.1:5012/dashboard/"+usernm)
                else:
                    result = ibm_db.fetch_both(stmt)
            else:
             if  result["USERNAME"]==usernm  and   result["PASSWORD"]==password  and
result["SUBUSER"] == 0:
                return redirect("http://127.0.0.1:5002/dashboard/"+usernm)
             else:
                 result=ibm_db.fetch_both(stmt)


         return redirect("http://127.0.0.1:5000")


        except Exception as e:
            print(e)
            return redirect("http://127.0.0.1:5000")
    except Exception as e:
        print(e)
        return render_template("sorry.html")


@app.route("/uservalidate",methods=["POST"])
def validateusername():
    usr=request.form["fname"]
    print(usr )
    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7YkTR3uKK3a",
            '', '')
        try:
            qry = f"SELECT  * FROM FZN32689.REGISTRATION"
            stmt = ibm_db.exec_immediate(con, qry)
            result = ibm_db.fetch_both(stmt)
            while result!=False:
                if(result["USERNAME"]==usr):
                    return ""
                else:
                    result = ibm_db.fetch_both(stmt)
            return "user not found"


        except:
            return "user not found"
    except:
        pass
@app.route("/passwordvalidate",methods=["POST"])
def validatepassword():
    password=request.form["pass"]
    username=request.form["name"]
    print(password)
    print(username)
    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7Yk
TR3uKK3a",
            '', '')
        try:
            qry = f"SELECT  * FROM FZN32689.REGISTRATION"
            stmt = ibm_db.exec_immediate(con, qry)
            result = ibm_db.fetch_both(stmt)
            while result!=False:
                if(result["USERNAME"]==username and result["PASSWORD"]==password):
                    otp[result["USERNAME"]]=TwoStepAuthenticator.generateOTP()
                    TwoStepAuthenticator.send_otp(result["MAILID"],otp[result["USERNAME"]])
                    return ""
                else:
                    result = ibm_db.fetch_both(stmt)
            return "incorrect password"


        except:
            return "incorrect password"
    except:
        pass
@app.route("/verifyotp",methods=["POST"])
def verifyotp():
    print("hii fro verifier of otp")
    mail=request.form["mail"]
    ot=request.form["otp"]
    print(mail)
    print(ot)
    if otp[mail]== ot:
        print(ot)
        print("matched")
        return "otp matched"
    print("mismatch")
    return "otp mismatch"
if __name__=="__main__":
    app.run(debug=True)
```

## Two Step Authenticator:

This part will provide **OTP** system to enhance the security of the web page.

**TwoStepAuthenticator.py**

```
import sendgrid
from python_http_client.exceptions import HTTPError
import math,random
import apikey

def send_otp(mailId,otp):
    API_KEY = apikey.api_key
    sg = sendgrid.SendGridAPIClient(API_KEY)
    data = {
        "personalizations": [
            {
                "to": [
                    {
                        "email": mailId
                    }
                ],
                "subject": "OTP FROM INVENTORY MANAGEMENT SYSTEM IBM PROJECT"
            }
        ],
        "from": {
            "email": "bhawinjasperbj@gmail.com"
        },
        "content": [
            {
                "type": "text/plain",
                "value": "your otp don't share with any one " + otp
            }
        ]
    }
    try:
        response = sg.client.mail.send.post(request_body=data)
        print(response.status_code)
        print(response.body)
        print(response.headers)
        return ""
    except HTTPError as e:
        print(e.to_dict)
        return "INVALID MAILID"



def generateOTP():
    # Declare a digits variable
    # which stores all digits
    digits = "0123456789"
    OTP = ""

    # length of password can be changed
    # by changing value in range
    for i in range(4):
        OTP += digits[math.floor(random.random() * 10)]
    print(OTP)
    return OTP
```

## REGISTRATION PAGE FOR MAIN USER & HUB USER:

This part will provide registration page for main user and hub user. but **hub user registration** page will be available only inside the **dash board** we can access it only after the creation of main user account.

### TEMPLATE:

### REGISTRATION PAGE:

**REGISTER.HTML:**

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Inventory management register page</title>
     <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" 
rel="stylesheet">
  <script 
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>


   <script>


     var username=""
     var mailid=""



       function loadDoc(value) {
        username=value;
     const xhttp = new XMLHttpRequest();
     xhttp.onload = function() {
         document.getElementById("vl1").innerHTML=this.responseText;
      }
  xhttp.open("POST", "/validationusername");
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhttp.send("fname="+value);
    }
      function loadDoc1(value) {
           mailid=value
```

```
const xhttp = new XMLHttpRequest();
xhttp.onload = function() {
    document.getElementById("vl2").innerHTML=this.responseText;
    if( document.getElementById("vl2").innerHTML==""){
     var x = document.createElement("INPUT");
    var y=document.createElement("br")
    var z=document.createElement("p")
    var z1=
      z.setAttribute("id","otpp")
      x.setAttribute("type", "password");
      x.setAttribute("placeholder","enter 4 digit otp")
      x.setAttribute("class","form-control")
      x.setAttribute("id","otp")
     document.getElementById("mail").appendChild(x);
     document.getElementById("mail").appendChild(y);
     document.getElementById("mail").appendChild(y);
     document.getElementById("mail").appendChild(z);
     document.getElementById("mail").appendChild(y);
      var z1=document.getElementById("otp")
      z1.addEventListener("focusout",function(){

        const xhttp = new XMLHttpRequest();
xhttp.onload = function() {
    document.getElementById("otpp").innerHTML=this.responseText;
    if( document.getElementById("otpp").innerHTML=="otp matched"){
      document.getElementById("otpp").style.color="green";
    }else{
      document.getElementById("otpp").style.color="red";
```

```
        }
     }
xhttp.open("POST", "/verifyotp");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

        xhttp.send("mail="+mailid+"&"+"otp="+this.value);
          });
        }
     }
xhttp.open("POST", "/validationmailid");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
   }

     function loadDoc2(value) {

    const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("vl3").innerHTML=this.responseText;
     }
xhttp.open("POST", "/validationphno");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
   }
   var ps=true
   var pasw=""
   function ok(){
     alert("your data is processing don't click submit button simultaneously");
```

```
      if(document.getElementById("vl1").textContent!=""){
        ps=false;
      }
      if(document.getElementById("vl2").textContent!=""){
        ps=false;
      }
      if(document.getElementById("vl3").textContent!=""){
        ps=false;
      }
      if(document.getElementById("vl4").textContent!=""){
        ps=false;
      }
      if(document.getElementById("otpp").innerHTML!="otp matched"){
        ps=false;
      }
     return ps;
   }
   function passwordck(value){
            const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("vl4").innerHTML=this.responseText;
     }
xhttp.open("POST", "/psck");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
           if(document.getElementById("vl4").textContent==""){
              pasw=value;
               document.getElementById("vl4").innerHTML="";
           }
```

```
          if(pasw==value){
              document.getElementById("vl5").innerHTML=""
              ps=true;
            }else{
               document.getElementById("vl5").innerHTML="PASSWORD MISMATCH";
               ps=false;
            }



     }

</script>

</head>
<body >
<div class="container mt-3" >
    <h3>REGISTER</h3>
<form action="/mainvalidate" method="post"  onsubmit="return ok()">
    <input class="form-control" type="text" placeholder="username" name="usernm" onfocusout="loadDoc(this.value)" required><p id="vl1" style="color:red">username should contain only lower case letter and numeric values.</p><br><br>
    <div id="mail">
    <input class="form-control"  type="email" placeholder="eneter your maild" name="mail" onfocusout="loadDoc1(this.value)"required><p id="vl2" style="color:red">.</p><br><br>
    </div>
    <div id="ph">
     <input class="form-control"  type="tel" placeholder="enete your phone number " name="phno" onfocusout="loadDoc2(this.value)"required><p id="vl3" style="color:red">.</p><br><br>
    </div>
```

```
        <input class="form-control" type="password" placeholder="Enter your password" name="pass" onfocusout="passwordck(this.value)" required><p id="vl4" style="color:red">.</p><br><br>

    <input class="form-control" type="password" placeholder="confirm password" onfocusout="passwordck(this.value)" required><p id="vl5" style="color:red"></p><br><br>

    <input  class="btn btn-primary" type="submit" value="submit"><br><br>

    <a href="http://127.0.0.1:5000/">Allready have an account ? login here...</a>

</form>

</div>


</body>

</html>
```

**SUB REGISTER.HTML:**

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Inventory management register page</title>

      <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet">

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>


    <script>


      var username="";

      var mailid="";

      var phno="";

      var psw="";
      function loadDoc(value) {
       username=value;
    const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("vl1").innerHTML=this.responseText;
     }
xhttp.open("POST", "/validationusername");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
   }
     function loadDoc1(value) {
            mailid=value

    const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("vl2").innerHTML=this.responseText;
        if( document.getElementById("vl2").innerHTML==""){
         var x = document.createElement("INPUT");
        var y=document.createElement("br")
        var z=document.createElement("p")
        var z1=
          z.setAttribute("id","otpp")
          x.setAttribute("type", "password");
          x.setAttribute("placeholder","enter 4 digit otp")
          x.setAttribute("class","form-control")
          x.setAttribute("id","otp")
         document.getElementById("mail").appendChild(x);
         document.getElementById("mail").appendChild(y);
```

```
          document.getElementById("mail").appendChild(y);
          document.getElementById("mail").appendChild(z);
          document.getElementById("mail").appendChild(y);
           var z1=document.getElementById("otp")
           z1.addEventListener("focusout",function(){

             const xhttp = new XMLHttpRequest();
     xhttp.onload = function() {
         document.getElementById("otpp").innerHTML=this.responseText;
         if( document.getElementById("otpp").innerHTML=="otp matched"){
           document.getElementById("otpp").style.color="green";
         }else{
           document.getElementById("otpp").style.color="red";
         }
      }
   xhttp.open("POST", "/verifyotp");
   xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

         xhttp.send("mail="+mailid+"&"+"otp="+this.value);
           });
         }
      }
   xhttp.open("POST", "/validationmailid");
   xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
   xhttp.send("fname="+value);
     }

      function loadDoc2(value) {
            phno=value;
```

```
    const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("vl3").innerHTML=this.responseText;
     }
xhttp.open("POST", "/validationphno");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
   }
   var ps=true
   var pasw=""
   function ok(){
     alert("your data is processing don't click submit button simultaneously");

      if(document.getElementById("vl1").textContent!=""){
        ps=false;
      }
      if(document.getElementById("vl2").textContent!=""){
        ps=false;
      }
      if(document.getElementById("vl3").textContent!=""){
        ps=false;
      }
      if(document.getElementById("vl4").textContent!=""){
        ps=false;
      }
      if(document.getElementById("otpp").innerHTML!="otp matched"){
        ps=false;
      }
      if(ps){
```

```
    const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
      alert(this.responseText);
      history.back();
     }
xhttp.open("POST", "{{' http://127.0.0.1:5001/subvalidate/%s/%s'%(name,hubname)}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
alert("request maded")
xhttp.send("usernm="+username+"&mail="+mailid+"&phno="+phno+"&pass="+psw);
return false;
     }else{
       return ps;
     }


   }
   function passwordck(value){
    psw=value;
           const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
       document.getElementById("vl4").innerHTML=this.responseText;
     }
xhttp.open("POST", "/psck");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
          if(document.getElementById("vl4").textContent==""){
             pasw=value;
              document.getElementById("vl4").innerHTML="";
          }
```

```
         if(pasw==value){
            document.getElementById("vl5").innerHTML=""
            ps=true;
          }else{
             document.getElementById("vl5").innerHTML="PASSWORD MISMATCH";
             ps=false;
          }



    }

</script>

</head>
<body >
<div class="container mt-3" >
   <h3>MANAGER REGISTER </h3>
<form   onsubmit="return ok()">
   <input     class="form-control"     type="text"  placeholder="username"  name="usernm" onfocusout="loadDoc(this.value)" required><p id="vl1" style="color:red">username should contain only lower case letter and numeric values.</p><br><br>
   <div id="mail">
   <input  class="form-control"  type="email" placeholder="eneter your maild" name="mail" onfocusout="loadDoc1(this.value)"required><p id="vl2" style="color:red">.</p><br><br>
   </div>
   <div id="ph">
    <input     class="form-control"    type="tel"  placeholder="enete  your  phone  number  " name="phno"        onfocusout="loadDoc2(this.value)"required><p        id="vl3" style="color:red">.</p><br><br>
   </div>
```

```
<input class="form-control" type="password" placeholder="Enter your password" name="pass" onfocusout="passwordck(this.value)" required><p id="vl4" style="color:red">.</p><br><br>
<input class="form-control" type="password" placeholder="confirm password" onfocusout="passwordck(this.value)" required><p id="vl5" style="color:red"></p><br><br>
<input  class="btn btn-primary" type="submit" value="submit"><br><br>
<a href="http://127.0.0.1:5000/">Allready have an account ? login here...</a>
</form>
</div>


</body>
</html>
```

**<u>BUSINESS LOGIC:</u>**

**This part provide business logic for registration. It includes FLASK API, IBM DB API, SENDGRID API, IBM COS SDK API FOR OBJECT STORAGE.**

**<u>MAIN.PY:</u>**

```
from flask import  Flask,render_template,redirect,request
import ibm_db
import re
import ObjectStorage
import json
import os
import MailVerfication
app = Flask(__name__)
profile={
 "profileIamge":"",
  "userName": "",
   "mailId" :"",
    "phNo" :"",
    "settings" : ""
}
Settings={
    "productranking":"",
    "hubranking":"",
    "productalertkl":"",
     "productalertcnt":""
}
hub={
"listofhubs":[]
}
hubmanger={
  "ownername" : "",
  "hubname" : ""
}
otp={}
@app.route("/mainregister")
def Register():
    return render_template("Register.html")
@app.route("/subregister/<name>/<hubname>")
def Subregister(name,hubname):
    print(name)
    print(hubname)

    return render_template("subregister.html",name=name,hubname=hubname)
@app.route("/mainvalidate",methods=["POST"])
def mainregister():
     usernm=request.form["usernm"]
     mailid=request.form["mail"]
    phno=request.form["phno"]
    password=request.form["pass"]
    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7YkTR3uKK3a",
            '', '')
        qry=f"INSERT INTO FZN32689.REGISTRATION (USERNAME,MAILID,PHNO,PASSWORD,SUBUSER) VALUES('{usernm}','{mailid}','{phno}','{password}',0);"
        print(qry)

        rs=ObjectStorage.create_bucket(usernm)
        print(rs)
        if(rs != None):
            return "Registration failed due to invalid username please try again later"
        profile["userName"]=usernm
        profile["mailId"] = mailid
        profile["phNo"] = phno
        jsn=json.dumps(Settings)
        file=open(usernm+"settings","w")
        file.write(jsn)
        file.close()
        ObjectStorage.multi_part_upload(usernm,usernm+"settings",os.path.abspath(usernm+"settings"))
        os.remove(os.path.abspath(usernm+"settings"))
        profile["settings"]=usernm+"settings"
        jsn1=json.dumps(profile)
        jsn2=json.dumps(hub)
        col1=usernm+"profile"
        col2=usernm+"hub"
        file=open(usernm+"profile","w")
        file.write(jsn1)
        file.close()

ObjectStorage.multi_part_upload(usernm,usernm+"profile",os.path.abspath(usernm+"profile
"))
        os.remove(os.path.abspath(usernm+"profile"))
        file=open(usernm+"hub","w")
        file.write(jsn2)
        file.close()

ObjectStorage.multi_part_upload(usernm,usernm+"hub",os.path.abspath(usernm+"hub"))
        os.remove(os.path.abspath(usernm+"hub"))


        qry1= f"INSERT INTO  FZN32689.MAINUSER (USERNAME,PROFILE,HUB)
VALUES('{usernm}','{col1}','{col2}');"
        print(qry1)
        stm = ibm_db.exec_immediate(con, qry)
        stm=ibm_db.exec_immediate(con,qry1)
        return redirect("http://127.0.0.1:5000/")
    except Exception as e:
        print(e)
        return "registration failed"
@app.route("/subvalidate/<name>/<hubname>",methods=["POST"])
def subregister(name,hubname):
    usernm=request.form["usernm"]
    mailid=request.form["mail"]
    phno=request.form["phno"]
    password=request.form["pass"]
    print(usernm)
    print(mailid)
    print(phno)
    print(password)
    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7Yk
TR3uKK3a",
            '', '')
        qry=f"INSERT INTO FZN32689.REGISTRATION
(USERNAME,MAILID,PHNO,PASSWORD,SUBUSER)
VALUES('{usernm}','{mailid}','{phno}','{password}',1);"
        print(qry)

        rs = ObjectStorage.create_bucket(usernm)
        print(rs)
        if (rs != None):
            return "Registration failed due to invalid username please try again later"
        profile["userName"] = usernm
        profile["mailId"] = mailid
        profile["phNo"] = phno
        hubdetails=ObjectStorage.get_item(name,name+"hub")
        print(hubdetails)
        hubdetails=hubdetails.decode("UTF -8")
        data=json.loads(hubdetails)
        for x in data["listofhubs"]:
            if x["HubName"]==hubname:
                hubmanger["hubname"]=hubname
                hubmanger["ownername"]=name
            hub["listofhubs"].append(hubmanger)
            print(x["HubManager"]+"manager")
            if x["HubManager"] != "":
                print(x["HubManager"])
                hubmanagerd = ObjectStorage.get_item(x["HubManager"], x["HubManager"] +
"hub")
                print(hubmanagerd)
                hubmanagerd = hubmanagerd.decode("UTF -8")
                hubmanagerd = json.loads(hubmanagerd)
                for y in hubmanagerd["listofhubs"]:
                    if y["hubname"] == hubname:
                        hubmanagerd["listofhubs"].remove(y)
                        break
                jsonm = json.dumps(hubmanagerd)
                filem = open(x["HubManager"] + "hub", "w")
                filem.write(jsonm)
                filem.close()
                ObjectStorage.multi_part_upload(x["HubManager"], x["HubManager"] + "hub",
                                                os.path.abspath(x["HubManager"] + "hub"))
                os.remove(os.path.abspath(x["HubManager"] + "hub"))
                break
            else:
                x["HubManager"] = usernm
                break
            break

        jsn = json.dumps(Settings)
        file = open(usernm + "settings", "w")
        file.write(jsn)
        file.close()
        ObjectStorage.multi_part_upload(usernm, usernm + "settings", os.path.abspath(usernm + "settings"))
        os.remove(os.path.abspath(usernm + "settings"))
        fl = open(name + "hub", "w")
        fl.write(json.dumps(data))
        fl.close()
        ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
        os.remove(os.path.abspath(name + "hub"))
        profile["settings"] = usernm + "settings"
        jsn1 = json.dumps(profile)
        jsn2 = json.dumps(hub)
        col1 = usernm + "profile"
        col2 = usernm + "hub"
        file = open(usernm + "profile", "w")
        file.write(jsn1)
        file.close()
        ObjectStorage.multi_part_upload(usernm, usernm + "profile", os.path.abspath(usernm + "profile"))
        os.remove(os.path.abspath(usernm + "profile"))
        file = open(usernm + "hub", "w")
        file.write(jsn2)
        file.close()
        ObjectStorage.multi_part_upload(usernm, usernm + "hub", os.path.abspath(usernm + "hub"))
        os.remove(os.path.abspath(usernm + "hub"))

        qry1 = f"INSERT INTO  FZN32689.SUBUSER (USERNAME,PROFILE,HUB) VALUES('{usernm}','{col1}','{col2}');"
        print(qry1)
        stm = ibm_db.exec_immediate(con, qry)
```

```
         stm = ibm_db.exec_immediate(con, qry1)
         MailVerfication.message(mailid,"Congratulations you account for managing the hub has been created ","Your username = "+usernm+" password = "+password)
         return "Registered Successfully"
     except Exception as e:
         print(e)
         return "registration failed"
@app.route("/validationusername",methods=["POST"])
def validateusername():
   unm=request.form["fname"]
   try:
    con = ibm_db.connect(
        "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7YkTR3uKK3a",
        '', '')
    qry = f"SELECT  * FROM FZN32689.REGISTRATION"
    stmt = ibm_db.exec_immediate(con, qry)
    result = ibm_db.fetch_both(stmt)
    while result!=False:
        if(result["USERNAME"]==unm):
            return "USER ALREADY EXIST"
        else:
            result = ibm_db.fetch_both(stmt)
            print(unm)


   except:
       print("something went wrong")
   return ""
```

```
@app.route("/validationmailid",methods=["POST"])
def validatemailid():
    unm = request.form["fname"]
    if(unm == ""):
        return "INVALID MAIL ID"
    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7YkTR3uKK3a",
            "", "")
        qry = f"SELECT  * FROM FZN32689.REGISTRATION"
        stmt = ibm_db.exec_immediate(con, qry)
        result = ibm_db.fetch_both(stmt)
        while result != False:
            if (result["MAILID"] == unm):
                return "USER ALREADY EXIST"
            else:
                result = ibm_db.fetch_both(stmt)
        otp[unm]=MailVerfication.generateOTP()
        print(otp)
        return MailVerfication.send_otp(unm,otp[unm])


    except:
        print("something went wrong")
    return ""
@app.route("/validationphno",methods=["POST"])
def validatephno():
    unm = request.form["fname"]
    if (len(unm) != 10):
        return "INVALID PHONE NUMBER"

    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7Yk
TR3uKK3a",
            "", "")
        qry = f"SELECT  * FROM FZN32689.REGISTRATION"
        stmt = ibm_db.exec_immediate(con, qry)
        result = ibm_db.fetch_both(stmt)

        while result != False:
            if (result["PHNO"] == unm):
                return "USER ALREADY EXIST"
            else:
                result = ibm_db.fetch_both(stmt)

    except:
        print("something went wrong")
    return ""
@app.route("/psck",methods=["POST"])
def passwordchecker():
    passwd = request.form["fname"]
    reg = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!#%*?&]{6,20}$"

    # compiling regex
    pat = re.compile(reg)
    # searching regex
    mat = re.search(pat, passwd)

    # validating conditions
    if mat:
        return ""
    else:
        return "password should contain 6 to 20 characters,one special symbol,at least one uppercase and one lowercase character, at least one number"

@app.route("/verifyotp",methods=["POST"])
def verifyotp():
    print("hii fro verifier of otp")
    mail=request.form["mail"]
    ot=request.form["otp"]
    print(mail)
    print(ot)
    if otp[mail]== ot:
        print(ot)
        print("matched")
        return "otp matched"
    print("mismatch")
    return "otp mismatch"

if __name__=="__main__":
    app.run(port=5001,debug=True)
```

## **Two Step Authenticator:**

This part will provide **OTP** system to verify the mail id.

This part also includes message system to share login credentials to hub manager at the time of allocating hub manager to hubs.

**MailVerfication.py**

```
import sendgrid
from python_http_client.exceptions import HTTPError
import math,random
import apikey

def send_otp(mailId,otp):
    API_KEY = apikey.api_key

    sg = sendgrid.SendGridAPIClient(API_KEY)
    data = {
        "personalizations": [
            {
                "to": [
                    {
                        "email": mailId
                    }
                ],
                "subject": "OTP FROM INVENTORY MANAGEMENT SYSTEM IBM PROJECT"
            }
        ],
        "from": {
            "email": "bhawinjasperbj@gmail.com"
        },
        "content": [
            {
                "type": "text/plain",
                "value": "your otp don't share with any one " + otp
            }
        ]
    }
    try:
        response = sg.client.mail.send.post(request_body=data)
        print(response.status_code)
        print(response.body)
        print(response.headers)
        return ""
    except HTTPError as e:
        print(e.to_dict)
        return "INVALID MAILID"



def generateOTP():
    # Declare a digits variable
    # which stores all digits
    digits = "0123456789"
    OTP = ""

    # length of password can be changed
    # by changing value in range
    for i in range(4):
        OTP += digits[math.floor(random.random() * 10)]
    print(OTP)
    return OTP
def message(mailId,subject,message):
```

```
API_KEY = apikey.api_key

sg = sendgrid.SendGridAPIClient(API_KEY)
data = {
    "personalizations": [
        {
            "to": [
                {
                    "email": mailId
                }
            ],
            "subject": subject
        }
    ],
    "from": {
        "email": "bhawinjasperbj@gmail.com"
    },
    "content": [
        {
            "type": "text/plain",
            "value": message
        }
    ]
}
try:
    response = sg.client.mail.send.post(request_body=data)
    print(response.status_code)
    print(response.body)
    print(response.headers)
```

```
        return ""
    except HTTPError as e:
        print(e.to_dict)
        return "INVALID MAILID"
```

**ObjectStorage.py**

This part will provide access to work with IBM CLOUD OBJECT STORAGE .This project mostly store data in JSON format only three sql table had been used in this project.

```
import ibm_boto3
from ibm_botocore.client import Config, ClientError





COS_ENDPOINT="https://s3.tok.ap.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID="dRpfBDLhp5Y2FqwqaZHEq6cWeinyufVjZLRz0VNl7Hnj"
COS_INSTANCE_CRN="crn:v1:bluemix:public:cloud-object-storage:global:a/702af44240f54d66ba7adebefb61dd74:21d01580-e4e2-41a3-8589-ef29aaacb70d::"
COS_BUCKET_LOCATION="jp-tok-smart"

cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)
def get_buckets():
    print("Retrieving list of buckets")
    try:
        buckets = cos.buckets.all()
        print(buckets)
        for bucket in buckets:
            print("Bucket Name: {0}".format(bucket.name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve list buckets: {0}".format(e))



def create_bucket(bucket_name):
    print("Creating new bucket: {0}".format(bucket_name))
    try:
        cos.Bucket(bucket_name).create()
        print("Bucket: {0} created!".format(bucket_name))
    except ClientError as be:
      return "CLIENT ERROR: {0}\n".format(be)
    except Exception as e:
        print("Unable to create bucket: {0}".format(e))

def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15
        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB

        with open(file_path, "rb") as file_data:
             cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))



def get_item(bucket_name, item_name):
    print("Retrieving item from bucket: {0}, key: {1}".format(bucket_name, item_name))
    try:
        file = cos.Object(bucket_name, item_name).get()

        by=file["Body"].read()
        return  by




    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve file contents: {0}".format(e))




def delete_bucket(bucket_name):
    print("Deleting bucket: {0}".format(bucket_name))
    try:
        cos.Bucket(bucket_name).delete()
        print("Bucket: {0} deleted!".format(bucket_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to delete bucket: {0}".format(e))
```

## MIAN DASHBOARD:

This part includes dashboard for main user and also for hub user.

In that dash board it contains Profile, Settings, Home page, Analysis page, Ranking Page,Hub entry page ,Hub dashboard.

## TEMPLATES:

**MainBoard.HTML:**

```
import ibm_boto3
from ibm_botocore.client import Config, ClientError




COS_ENDPOINT="https://s3.tok.ap.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID="dRpfBDLhp5Y2FqwqaZHEq6cWeinyufVjZLRz0VNl7Hnj"
COS_INSTANCE_CRN="crn:v1:bluemix:public:cloud-object-storage:global:a/702af44240f54d66ba7adebefb61dd74:21d01580-e4e2-41a3-8589-ef29aaacb70d::"
COS_BUCKET_LOCATION="jp-tok-smart"


cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)
def get_buckets():
    print("Retrieving list of buckets")
    try:
        buckets = cos.buckets.all()
        print(buckets)
        for bucket in buckets:
            print("Bucket Name: {0}".format(bucket.name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve list buckets: {0}".format(e))


def create_bucket(bucket_name):
    print("Creating new bucket: {0}".format(bucket_name))
    try:
        cos.Bucket(bucket_name).create()
        print("Bucket: {0} created!".format(bucket_name))
    except ClientError as be:
      return "CLIENT ERROR: {0}\n".format(be)
    except Exception as e:
        print("Unable to create bucket: {0}".format(e))

def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )
        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB

        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))



def get_item(bucket_name, item_name):
    print("Retrieving item from bucket: {0}, key: {1}".format(bucket_name, item_name))
    try:
        file = cos.Object(bucket_name, item_name).get()

        by=file["Body"].read()
        return  by




    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve file contents: {0}".format(e))



def delete_bucket(bucket_name):
    print("Deleting bucket: {0}".format(bucket_name))
    try:
        cos.Bucket(bucket_name).delete()
        print("Bucket: {0} deleted!".format(bucket_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to delete bucket: {0}".format(e))
```

**Profile.HTML:**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{{name}}</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>
<style>
```

```
#dp{
 border-radius:8 px;
}
#profile{
  border-style: solid;
  border-color: rgb(68, 183, 215);
  border-width: 1px;
  width :50%;
  float :left;
}
#settings{
  border-style: solid;
  border-width: 1px;
  border-color: rgb(68, 183, 215);
  width:50%;
  float :right;
}

/* Button used to open the contact form - fixed at the bottom of the page */
.open-button {
  background-color: #555;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  opacity: 0.8;
  position: fixed;
  bottom: 23px;
  right: 28px;
  width: 280px;
}

/* The popup form - hidden by default */
.form-popup {
  display: none;
  position: fixed;
  bottom: 0;
  right: 15px;
  border: 3px solid #f1f1f1;
  z-index: 9;
}

/* Add styles to the form container */
.form-container {
  max-width: 300px;
  padding: 10px;
  background-color: white;
}

/* Full-width input fields */
.form-container input[type=text], .form-container input[type=password] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  border: none;
  background: #f1f1f1;
}
```

```
/* When the inputs get focus, do something */
.form-container input[type=text]:focus, .form-container input[type=password]:focus {
  background-color: #ddd;
  outline: none;
}

/* Set a style for the submit/login button */
.form-container .btn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  width: 100%;
  margin-bottom:10px;
  opacity: 0.8;
}

/* Add a red background color to the cancel button */
.form-container .cancel {
  background-color: red;
}

/* Add some hover effects to buttons */
.form-container .btn:hover, .open-button:hover {
  opacity: 1;
}
</style>
<script>
```

```
  function openForm() {
  document.getElementById("myForm").style.display = "block";
}

function closeForm() {
  document.getElementById("myForm").style.display = "none";
}
function openForm1() {
  document.getElementById("myForm1").style.display = "block";
}

function closeForm1() {
  document.getElementById("myForm1").style.display = "none";
}
</script>
</head>
<body>
<div id="menubar">
    <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;" >
         <div class="container-fluid">
    <div class="navbar-brand" onclick="history.back()">
      <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30" height="24" class="d-inline-block align-text-top">
    </div>
  </div>
        <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
        <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>
        <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>
        <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>
        <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>


    </nav>


</div>
<div>
<div class="container" id="profile">
    <img id="dp" src="MAINDASHBOARD\anojprofilepic.png" alt="{{'%s'%name}}"
style="width:200px"><hr><br>
    <form action = {{'/changeprofile/%s'%name}} method = "post" enctype="multipart/form-
data">
        <input type="file" name="file" />
        <input type = "submit" value="Upload" class="btn btn-primary">
    </form>  <hr><br>
    <span > {{data["mailId"]}} &emsp;   <button class="btn btn-primary"
onclick="openForm()">change</button></span><hr><br>
    <span > {{data["phNo"]}} &emsp;   <button class="btn btn-primary"
onclick="openForm1()">change</button></span>
      <hr><br>   <a  href=" {{'/changepassword/%s'%name}}">CHANGE PASSWORD</a>
      <div class="form-popup" id="myForm">
        <form action="/action_page.php" class="form-container">



          <label for="email"><b>Email Id</b></label><br><br>
          <input name="email"  type="email" placeholder="Enter new mail id" class="form-
control" ><br>



          <label for="psw"><b>VALUE</b></label>
          <input type="text" placeholder="Enter the otp" name="psw" required>
          <button type="submit" class="btn">Submit</button>

          <button type="button" class="btn cancel" onclick="closeForm()">Close</button>

        </form>

      </div>

      <div class="form-popup" id="myForm1">

        <form action="/action_page.php" class="form-container">

          <label for="psw"><b>PH NO :</b></label><br>

          <input type="tel" placeholder="Enter the number" name="psw" class="form-control" required><br><br>


          <button type="submit" class="btn">Submit</button>

          <button type="button" class="btn cancel" onclick="closeForm1()">Close</button>

        </form>

      </div><br><br><br>

      <button class="btn btn-danger" ><a href="http://127.0.0.1:5000">Log Out</a></button>



</div>

<script>

  function settings(){

     var ck1=document.getElementById("Hday");

     var ck2=document.getElementById("Hweek");

     var ck3=document.getElementById("Hmonth");

     var ck4=document.getElementById("Pday");

     var ck5=document.getElementById("Pweek");

     var ck6=document.getElementById("Pmonth");

     var hr="";

     var pr="";

     var kl=document.getElementById("kilo").value;
```

```
    }

    var cnt=document.getElementById("count").value;
    if(ck1.checked == true){
       hr=ck1.value;
    }
    else if(ck2.checked == true){
     hr=ck2.value;
    }else if(ck3.checked == true){
     hr=ck3.value;
    }
    if(ck4.checked == true){
       pr=ck4.value;
    }
    else if(ck5.checked == true){
     pr=ck5.value;
    }else if(ck6.checked == true){
     pr=ck6.value;
    }
    alert("going to make changes");
    const xhttp = new XMLHttpRequest();
     xhttp.onload = function() {
         alert(this.responseText);



      }
  xhttp.open("POST", "{{'/settings/%s'%name}}");
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhttp.send("hr="+hr+"&pr="+pr+"&kl="+kl+"&cnt="+cnt);
  }
</script>
```

```
<div  class ="container" id="settings">
  <h3>Hub Ranking</h3>
  <input type="checkbox" value="day" class="form-check-input" id="Hday"> Day</input><br><br>
  <input type="checkbox" value="week" class="form-check-input" id="Hweek"> Week</input><br><br>
  <input type="checkbox" value="month" class="form-check-input" id="Hmonth"> Month</input><br><hr>
  <h3>Product Ranking</h3>
  <input type="checkbox" value="day" class="form-check-input" id="Pday"> Day</input><br><br>
  <input type="checkbox" value="week" class="form-check-input" id="Pweek"> Week</input><br><br>
  <input type="checkbox" value="month" class="form-check-input" id="Pmonth"> Month</input><br><hr>


  <h3>Product Alert</h3>
  <input type="text" class="form-control" placeholder="For Kilogram " id="kilo"></input><br><br>
  <input type="number" class="form-control"placeholder="for countable products" id="count"></input><br><br>


  <button class="btn btn-primary" class="form-control" onclick="settings()">Save Changes</button>
</div>
</div>



</body>
</html>
```

**Analysis.HTML:**

```
<!DOCTYPE html>
```

```
<html lang="en">

<head>

   <meta charset="UTF-8">

   <title>Analysis</title>

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">

<!-- JavaScript Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>


</head>

<script>

  function getHub1(){

     getHub();

     const xhttp = new XMLHttpRequest();

       xhttp.onload = function() {

           var data=this.responseText;

           var obj=document.getElementById("hublist");

           var pr=JSON.parse(data);

          var ls=pr["listofhubs"]

           for(let x in ls){

             var opt=document.createElement("option");

             opt.setAttribute("value",ls[x]["HubName"]);

             const optionText = document.createTextNode(ls[x]["HubName"]);

             opt.appendChild(optionText);

             obj.appendChild(opt);

           }
        }
   xhttp.open("POST", "{{'/gethublist/%s'%name}}");
   xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
   xhttp.send();
   }
   function getHub(){

     const xhttp = new XMLHttpRequest();
       xhttp.onload = function() {
           var data=this.responseText;
           var obj=document.getElementById("mytable");
           var pr=JSON.parse(data);
           var ls=pr["listofhubs"];
           alert(data);
           for(let x in ls){
            var row = obj.insertRow(-1);
            var cell1 = row.insertCell(0);
            var cell2 = row.insertCell(1);
            var cell3 = row.insertCell(2);
            var cell4 = row.insertCell(3);
            var cell5 = row.insertCell(4);
            var cell6 = row.insertCell(5);
            var salesgrowth=0;
            var profit=0;
            var customergrowth=0;
             const optionText1 = document.createTextNode(ls[x]["HubName"]);
             const optionText2 = document.createTextNode(ls[x]["HubLocation"]);
             const optionText3 = document.createTextNode(x);
```

```
            const optionText4 = document.createTextNode(salesgrowth);
            const optionText5 = document.createTextNode(profit);
            const optionText6 = document.createTextNode(customergrowth);
             cell1.appendChild(optionText3);
             cell2.appendChild(optionText1);
             cell3.appendChild(optionText2);
             cell4.appendChild(optionText4);
             cell5.appendChild(optionText5);
             cell6.appendChild(optionText6);
           }


       }
   xhttp.open("POST", "{{'/gethublist/%s'%name}}");
   xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
   xhttp.send();
   }
</script>
<body onload="getHub1()">
<div id="menubar">
    <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;" >
        <div class="navbar-brand" onclick="history.back()">
      <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30" height="24" class="d-inline-block align-text-top">
    </div>
         <div class="container-fluid">
    <a class="navbar-brand" href="{{'/profile/%s'%name}}">
      <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">
      {{name}}
    </a>
```

```
  </div>
        <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
        <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>
        <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>
        <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>

    </nav>
    <h3>Select Hub For Product Analysis</h3><br><br>
<select  id="hublist" class="form-select"  name="hub" onchange="productdetails(this.value)"
>
            <option><p>Select the hub to see the productdetails</p></option>
</select><br><br>
<table class="table" id="mytable">
  <thead>
    <tr>
      <th scope="col">S.no</th>
      <th scope="col">HUB NAME</th>
      <th scope="col">HUB LOCATION</th>
      <th scope="col">SALES GROWTH</th>
      <th scope="col">PROFIT</th>
      <th scope="col">CUSTOMER GROWTH</th>

    </tr>
  </thead>
  <tbody>

  </tbody>
</table>
</div>
```

```
</body>
</html>
```

**Ranking.HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Ranking</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
<script>
  function getHub1(){
    getHub();
    const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
          var data=this.responseText;
          var obj=document.getElementById("hublist");
          var pr=JSON.parse(data);
         var ls=pr["listofhubs"]
          for(let x in ls){
            var opt=document.createElement("option");
            opt.setAttribute("value",ls[x]["HubName"]);
            const optionText = document.createTextNode(ls[x]["HubName"]);
            opt.appendChild(optionText);
            obj.appendChild(opt);
          }

       }
  xhttp.open("POST", "{{'/gethublist/%s'%name}}");
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhttp.send();
  }
  function getHub(){

    const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
          var data=this.responseText;
          var obj=document.getElementById("mytable");
          var pr=JSON.parse(data);
          var ls=pr["listofhubs"];
          alert(data);
          for(let x in ls){
           var row = obj.insertRow(-1);
           var cell1 = row.insertCell(0);
           var cell2 = row.insertCell(1);
           var cell3 = row.insertCell(2);
           var cell4 = row.insertCell(3);

           var rank=1;

            const optionText1 = document.createTextNode(ls[x]["HubName"]);
```

```
            const optionText2 = document.createTextNode(ls[x]["HubLocation"]);
            const optionText3 = document.createTextNode(x);
            const optionText4 = document.createTextNode(rank);

             cell1.appendChild(optionText3);
             cell2.appendChild(optionText1);
             cell3.appendChild(optionText2);
             cell4.appendChild(optionText4);

           }

         }
    xhttp.open("POST", "{{'/gethublist/%s'%name}}");
    xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhttp.send();
    }
</script>
</head>
<body onload="getHub1()">
<div id="menubar">
    <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;" >
        <div class="navbar-brand" onclick="history.back()">
      <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30" height="24" class="d-inline-block align-text-top">
    </div>
         <div class="container-fluid">
    <a class="navbar-brand" href="{{'/profile/%s'%name}}">
      <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">
      {{name}}
    </a>
  </div>
        <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
        <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>
        <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>
        <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>

    </nav>
    <h3>Select Hub For Product Ranking</h3><br><br>
<select  id="hublist" class="form-select"  name="hub" onchange="productdetails(this.value)"
>
            <option><p>Select the hub to see the productdetails</p></option>
</select><br><br>
<table class="table" id="mytable">
  <thead>
    <tr>
      <th scope="col">S.no</th>
      <th scope="col">HUB NAME</th>
      <th scope="col">HUB LOCATION</th>
      <th scope="col">RANK</th>



    </tr>
  </thead>
  <tbody>

  </tbody>
</table>
```

```
</div>
</body>
</html>
```

**HubEntry.HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>HubEntry</title>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>
<style>
.sidenav  {
  font-family: "Lato", sans-serif;
}

.sidenav {
  position: relative;
  height: 100%;
  width: 15%;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  padding-top: 20px;
  float: left;
}

.sidenav a {
  padding: 6px 6px 6px 32px;
  text-decoration: none;
  font-size: 20px;
  color: #b7afaf;
  display: block;
}

.sidenav a:hover {
  color: #f1f1f1;
}

#main{
  float: right;
  width: 80%;
}

@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}
</style>
<script>
```

```
function changenav(value){
  const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("main").innerHTML=this.responseText;
        if(value=="AddProductDetails" || value=="AddSupplierDetails"){
          getHub();
        }
      }
xhttp.open("POST", "{{'/navforhubentry/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
}
 function getHub(){

  const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        var data=this.responseText;
        var obj=document.getElementById("hublist");
        var pr=JSON.parse(data);
       var ls=pr["listofhubs"]
        for(let x in ls){
          var opt=document.createElement("option");
          opt.setAttribute("value",ls[x]["HubName"]);
          const optionText = document.createTextNode(ls[x]["HubName"]);
          opt.appendChild(optionText);
          obj.appendChild(opt);
        }

      }
  xhttp.open("POST", "{{'/gethublist/%s'%name}}");
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhttp.send();
  }
</script>
</head>
<body>
<div id="menubar">
    <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;" >
        <div class="navbar-brand" onclick="history.back()">
      <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30" height="24" class="d-inline-block align-text-top">
    </div>
         <div class="container-fluid">
    <a class="navbar-brand" href="{{'/profile/%s'%name}}">
      <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">
      {{name}}
    </a>
  </div>
        <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
        <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>
        <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>
        <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>

    </nav>

</div>
<div class="sidenav">
  <a href="#" onclick="changenav(this.textContent)">Add Hub</a>
  <a href="#" onclick="changenav(this.textContent)">AddProductDetails</a>
  <a href="#" onclick="changenav(this.textContent)">AddSupplierDetails</a>
</div>
<div id="main">


</div>
</body>
</html>
```

**AddHub.HTML**

```
<form action={{'/hubentry/%s'%name}} method="post">
   <input name="hname" type="text" placeholder="Hub Name" class="form-control"><br><br>
   <input name="hloc" type="text" placeholder="Hub Location" class="form-control"><br><br>
   <input type="submit" class= "btn btn-primary">
</form>
```

**AddProduct.HTML**

```
<div>
   <h4>Product Details</h4><br><br>



   <form action={{'/addproduct/%s'%name}} method="post">
      <input name="pname" type="text" placeholder="Product Name" class="form-control"><br><br>
      <input name="sname" type="text" placeholder="Supplier Name" class="form-control"><br><br>
      <input name="price" type="text" placeholder="Product price" class="form-control"><br><br>
      <input name="sprice" type="text" placeholder="Selling price" class="form-control"><br><br>
```

```
        <input name="qty" type="text" placeholder="supplied quantity" class="form-control"><br><br>

        <input name="date" type="date" placeholder="Date of supplied" class="form-control"><br><br>

        <select  id="hublist" class="form-select"  name="hub" >

            <option><p>NONE</p></option>

        </select><br><br>

        <input type="submit" class= "btn btn-primary">

    </form>

</div>
```

**AddSupplier.HTML**

```
<div>

    <h4>Supplier Details</h4><br><br>

    <form action={{'/addsupplier/%s'%name}} method="post">

        <input name="sname" type="text" placeholder="Supplier Name" class="form-control"><br><br>

        <input name="sloc" type="text" placeholder="Supplied From" class="form-control"><br><br>

        <input name="qty" type="number" placeholder="Supplied No Of Products" class="form-control"><br><br>

        <select name="hub" id="hublist" class="form-select">

            <option><p>NONE</p></option>

        </select><br><br>

        <input type="submit" class= "btn btn-primary">

    </form>

</div>
```

**HubDashBoard.HTML**

```
<!DOCTYPE html>

<html lang="en">

<head>
```

```
   <meta charset="UTF-8">
   <title>HubDashBoard</title>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
<style>
.sidenav  {
  font-family: "Lato", sans-serif;
}

.sidenav {
  position: relative;
  height: 100%;
  width: 15%;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  padding-top: 20px;
  float: left;
}

.sidenav a {
  padding: 6px 6px 6px 32px;
  text-decoration: none;
  font-size: 20px;
  color: #b7afaf;
  display: block;
}

.sidenav a:hover {
  color: #f1f1f1;
}

#main{
  float: right;
  width: 80%;
}

@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}

/* Button used to open the contact form - fixed at the bottom of the page */
.open-button {
  background-color: #555;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  opacity: 0.8;
  position: fixed;
  bottom: 23px;
  right: 28px;
  width: 280px;
}

/* The popup form - hidden by default */
.form-popup {
  display: none;
  position: fixed;
  bottom: 0;
  right: 15px;
  border: 3px solid #f1f1f1;
  z-index: 9;
}

/* Add styles to the form container */
.form-container {
  max-width: 300px;
  padding: 10px;
  background-color: white;
}

/* Full-width input fields */
.form-container input[type=text], .form-container input[type=password] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  border: none;
  background: #f1f1f1;
}

/* When the inputs get focus, do something */
.form-container input[type=text]:focus, .form-container input[type=password]:focus {
  background-color: #ddd;
  outline: none;
}

/* Set a style for the submit/login button */
.form-container .btn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  width: 100%;
  margin-bottom:10px;
  opacity: 0.8;
}

/* Add a red background color to the cancel button */
.form-container .cancel {
  background-color: red;
}

/* Add some hover effects to buttons */
.form-container .btn:hover, .open-button:hover {
  opacity: 1;
}

</style>
<script>
  var index;
  function openForm(value) {
    index=Number(value);
    document.getElementById("myForm").style.display = "block";
}

function closeForm() {
  document.getElementById("myForm").style.display = "none";
}
function openForm1(value) {
  index=Number(value);
  document.getElementById("myForm1").style.display = "block";
}

function closeForm1() {
  document.getElementById("myForm1").style.display = "none";
}
function changeProductDetails(x){
  alert(x);
  if(x =="PRICE DETAILS"){
    var y=document.getElementById("pdform");
    var cnt=y.childElementCount;
     while(cnt>0){
```

```
   y.removeChild(y.firstElementChild);
   cnt--;
   }
  var inp=document.createElement("INPUT");
  var inp1=document.createElement("INPUT");
  var inp2=document.createElement("INPUT");
  var inp3=document.createElement("INPUT");
  inp.setAttribute("placeholder","Enter the Product Price");
  inp.setAttribute("type","text");
  inp.setAttribute("id","ppriceet");
  inp2.setAttribute("type","date");
  inp2.setAttribute("id","dateet");
  inp1.setAttribute("placeholder","Enter the Selling Price");
  inp1.setAttribute("type","text");
  inp1.setAttribute("id","spriceet");
  inp3.setAttribute("placeholder","Enter the quantity");
  inp3.setAttribute("type","text");
  inp3.setAttribute("id","qtyet");

  document.getElementById("pdform").appendChild(inp);
  document.getElementById("pdform").appendChild(inp1);
  document.getElementById("pdform").appendChild(inp2);
  document.getElementById("pdform").appendChild(inp3);
}else if(x == "SALES DETAILS"){

  var y=document.getElementById("pdform");
  var cnt=y.childElementCount;
   while(cnt>0){
      y.removeChild(y.firstElementChild);
      cnt--;
     }
    var inp=document.createElement("INPUT");
    var inp2=document.createElement("INPUT");
    inp.setAttribute("placeholder","Enter the qty");
    inp.setAttribute("type","text");
    inp.setAttribute("id",x+"qty");
    inp2.setAttribute("type","date");
    inp2.setAttribute("id",x+"date");
    y.appendChild(inp);
    y.appendChild(inp2);
   } else{
    var y=document.getElementById("pdform");
    var y=document.getElementById("pdform");
    var cnt=y.childElementCount;
     while(cnt>0){

      y.removeChild(y.firstElementChild);
      cnt--;
     }
    var inp=document.createElement("INPUT");
    inp.setAttribute("placeholder","Enter the "+x);
    inp.setAttribute("type","text");
    inp.setAttribute("id",x);
    y.appendChild(inp);
   }


  }
  function changenav(value){
  const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("main").innerHTML=this.responseText;
        if(value=="Hub's"){
          getHub();
        }else{
          getHub1();
        }
     }
xhttp.open("POST", "{{'/navforhubentry/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
}
function removeHub(hubname){
  alert("Going to Remove "+hubname);
  const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();
     }
xhttp.open("POST", "{{'/removehub/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("hub="+hubname);
}
function getHub(){


   const xhttp = new XMLHttpRequest();
     xhttp.onload = function() {
            var data=this.responseText;
            var obj=document.getElementById("mytable");
            var pr=JSON.parse(data);
            var ls=pr["listofhubs"];
            for(let x in ls){
             var row = obj.insertRow(-1);
              var cell1 = row.insertCell(0);
             var cell2 = row.insertCell(1);
             var cell3 = row.insertCell(2);
             var cell4 = row.insertCell(3);
             var cell5 = row.insertCell(4);
             var cell6 = row.insertCell(5);
              var a=document.createElement("a");

a.setAttribute("href","{{'http://127.0.0.1:5001/subregister/%s/'%(name)}}"+ls[x]["HubName"])
              if(ls[x]["HubManager"]==""){
               const optionText = document.createTextNode("ADD MANAGER");
               a.appendChild(optionText);
              }else{
               const optionText = document.createTextNode("CHANGE MANAGER");
               a.appendChild(optionText);
              }
              var bt=document.createElement("BUTTON");
              bt.setAttribute("class","btn btn-primary");
              bt.setAttribute("value",ls[x]["HubName"])
              bt.addEventListener("click",function(){
               removeHub(this.value);
             });
             var t = document.createTextNode("Remove");
```

```
          bt.appendChild(t);



          const optionText1 = document.createTextNode(ls[x]["HubName"]);
          const optionText2 = document.createTextNode(ls[x]["HubLocation"]);
          const optionText3 = document.createTextNode(ls[x]["HubManager"]);
          const optionText4 = document.createTextNode(x);
          cell1.appendChild(optionText4);
          cell2.appendChild(optionText1);
          cell3.appendChild(optionText2);
          cell4.appendChild(optionText3);
          cell5.appendChild(a);
          cell6.appendChild(bt);

        }

     }
xhttp.open("POST", "{{'/gethublist/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send();
}
function getHub1(){

  const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        var data=this.responseText;
        var obj=document.getElementById("hublist");
```

```
         var pr=JSON.parse(data);
        var ls=pr["listofhubs"]
         for(let x in ls){
           var opt=document.createElement("option");
           opt.setAttribute("value",ls[x]["HubName"]);
           const optionText = document.createTextNode(ls[x]["HubName"]);
           opt.appendChild(optionText);
           obj.appendChild(opt);
         }


      }
 xhttp.open("POST", "{{'/gethublist/%s'%name}}");
 xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
 xhttp.send();
 }
 function removeSupplier(suppliername){
   alert("removing "+suppliername);
   var hname=document.getElementById("hublist").value;


   const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();
      }
xhttp.open("POST", "{{'/removesupplier/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("sname="+suppliername+"&hname="+hname);
 }
 function supplierdetails(value){
```

```
const xhttp = new XMLHttpRequest();
  xhttp.onload = function() {
     var data=this.responseText;
       var obj=document.getElementById("mytable");
       var pr=JSON.parse(data);
       for(let x in pr){
        var row = obj.insertRow(-1);
         var cell1 = row.insertCell(0);
        var cell2 = row.insertCell(1);
        var cell3 = row.insertCell(2);
        var cell4 = row.insertCell(3);
        var cell5 = row.insertCell(4);
        var cell6 = row.insertCell(5)
         var bt=document.createElement("BUTTON");
         bt.setAttribute("class","btn btn-primary");
         bt.setAttribute("value",pr[x]["suppliername"]);

         var t = document.createTextNode("Remove");
        bt.addEventListener("click",function(){
          removeSupplier(this.value)}
          );
         bt.appendChild(t);
         var bt1=document.createElement("BUTTON");
         bt1.setAttribute("class","btn btn-primary");
         bt1.setAttribute("value",x);
         var t1 = document.createTextNode("Edit");

         bt1.appendChild(t1);
```

```
           bt1.addEventListener("click", function(){
            openForm(this.value);
           });
           const optionText1 = document.createTextNode(pr[x]["suppliername"]);
           const optionText2 = document.createTextNode(pr[x]["supplierlocation"]);
           const optionText3 = document.createTextNode(pr[x]["suppliedproducts"]);
           const optionText4 = document.createTextNode(x);
           cell1.appendChild(optionText4);
           cell2.appendChild(optionText1);
           cell3.appendChild(optionText2);
           cell4.appendChild(optionText3);
           cell5.appendChild(bt);
           cell6.appendChild(bt1)

         }
     }
xhttp.open("POST", "{{'/supplierdetails/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
 }
 function removeproductdetais(prname){
  window.confirm("Are you sure to delete");
   var prdname=prname;
   var urname="{{'%s'%name}}";
   var hubname=document.getElementById("hublist").value;

   const xhttp = new XMLHttpRequest();
     xhttp.onload = function() {
         alert(this.responseText);
```

```
        location.reload();


      }
xhttp.open("POST", "/removeproductdetails");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+prname+"&urname="+urname+"&hubname="+hubname);


}
function productdetails(value){
 const xhttp = new XMLHttpRequest();
   xhttp.onload = function() {
       var data=this.responseText;
       alert(data);


       var obj=document.getElementById("mytable");
        var pr=JSON.parse(data);
        for(let x in pr){
         var row = obj.insertRow(-1);
          var cell1 = row.insertCell(0);
         var cell2 = row.insertCell(1);
         var cell3 = row.insertCell(2);
         var cell4 = row.insertCell(3);
         var cell5 = row.insertCell(4);
         var cell6 = row.insertCell(5);
         var cell7 = row.insertCell(6);
         var cell8 = row.insertCell(7);
         var cell9 = row.insertCell(8);
         var cell10 = row.insertCell(9);
         var cell11 = row.insertCell(10);
 var bt=document.createElement("BUTTON");
 bt.setAttribute("class","btn btn-primary");
 bt.setAttribute("value",pr[x]["productname"])
 bt.addEventListener("click",function(){
    removeproductdetais(this.value);
 });

var t = document.createTextNode("Remove");

 bt.appendChild(t);
 var bt1=document.createElement("BUTTON");
 bt1.setAttribute("class","btn btn-primary");

 var t1 = document.createTextNode("Edit");
 bt1.setAttribute("value",x)
 bt1.appendChild(t1);
 bt1.addEventListener("click", function(){
  openForm1(this.value);
 });
 const optionText1 = document.createTextNode(x);
 const optionText2 = document.createTextNode(pr[x]["productname"]);
 const optionText3 = document.createTextNode(pr[x]["suppliername"]);
 cell1.appendChild(optionText1);
 cell2.appendChild(optionText2);
 cell3.appendChild(optionText3);
 cell10.appendChild(bt);
 var prprice="";
 var selprice="";
 var datec="";
var qty="";
var mnt=0;
var yr=0;
var date=0;
var saledqty="";
var saleddate="";
for(let y in pr[x]["pricedetails"]){

     var myar=pr[x]["pricedetails"][y]["date"].split("-");

     if(yr<=myar[0]){
        yr=myar[0];
       if (mnt<=myar[1]){
         mnt=myar[1];
         if (date<myar[2]){
           date=myar[2];
           prprice=pr[x]["pricedetails"][y]["productprice"];
           selprice=pr[x]["pricedetails"][y]["sellingprice"];
           datec=pr[x]["pricedetails"][y]["date"]
           qty=pr[x]["pricedetails"][y]["qty"];
         }
       }
     }
}
const optionText4 = document.createTextNode(prprice);
const optionText5 = document.createTextNode(selprice);
const optionText6 = document.createTextNode(datec);
const optionText7 = document.createTextNode(qty);
const optionText10 = document.createTextNode(saledqty);
```

```
           const optionText11 = document.createTextNode(saleddate);
           cell4.appendChild(optionText4);
           cell5.appendChild(optionText5);
           cell6.appendChild(optionText6);
           cell7.appendChild(optionText7);
           cell11.appendChild(bt1);
           cell8.appendChild(optionText10);
           cell9.appendChild(optionText11);



         }
     }
xhttp.open("POST", "{{'/productdetails/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
 }
 function changeSd(field){
  alert(field);
    if(field=="SUPPLIER NAME"){
      var pname=document.getElementById("vl").value;
      var hubname=document.getElementById("hublist").value;
      var table=document.getElementById("mytable");
      var obj= table.rows.item(index+1).cells;
      var currentproduct=obj.item(1).innerHTML;
      const xhttp = new XMLHttpRequest();
     xhttp.onload = function() {
         alert(this.responseText);
         location.reload();
      }
xhttp.open("POST", "{{'/changesdname/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);

   }
   else if(field=="SUPPLIER LOCATION"){
     var pname=document.getElementById("vl").value;
     var hubname=document.getElementById("hublist").value;
     var table=document.getElementById("mytable");
     var obj= table.rows.item(index+1).cells;
     var currentproduct=obj.item(1).innerHTML;
     const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();

     }
xhttp.open("POST", "{{'/changesdlocation/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);

   }
   else if(field=="NO OF PRODUCTS"){
     var pname=document.getElementById("vl").value;
     var hubname=document.getElementById("hublist").value;
     var table=document.getElementById("mytable");
     var obj= table.rows.item(index+1).cells;
     var currentproduct=obj.item(1).innerHTML;
     const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();

     }
xhttp.open("POST", "{{'/changesdqty/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);

   }



   }
   function changePd(field){
 alert(field);
   if(field=="PRODUCT NAME"){
     var pname=document.getElementById(field).value;
     var hubname=document.getElementById("hublist").value;
     var table=document.getElementById("mytable");
     var obj= table.rows.item(index+1).cells;
     var currentproduct=obj.item(1).innerHTML;
     const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();

     }
xhttp.open("POST", "{{'/changepname/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);

  }
  else if(field=="SUPPLIER NAME"){
    var pname=document.getElementById(field).value;
    var hubname=document.getElementById("hublist").value;
    var table=document.getElementById("mytable");
    var obj= table.rows.item(index+1).cells;
    var currentproduct=obj.item(1).innerHTML;
    const xhttp = new XMLHttpRequest();
   xhttp.onload = function() {
      alert(this.responseText);
      location.reload();

    }
xhttp.open("POST", "{{'/changesname/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);

  }
  else if(field=="PRICE DETAILS"){
    var price=document.getElementById("ppriceet").value;
    var sprice=document.getElementById("spriceet").value;
    var date=document.getElementById("dateet").value;
    var qty=document.getElementById("qtyet").value;
    var hubname=document.getElementById("hublist").value;
    var table=document.getElementById("mytable");
    var obj= table.rows.item(index+1).cells;
```

```
        var currentproduct=obj.item(1).innerHTML;
        var pname=obj.item(1).innerHTML;
        const xhttp = new XMLHttpRequest();
       xhttp.onload = function() {
          alert(this.responseText);
          location.reload();


        }
    xhttp.open("POST", "{{'/addpricedetails/%s'%name}}");
    xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname+"&
price="+price+"&sprice="+sprice+"&date="+date+"&qty="+qty);


      }
     else {
        var qty=document.getElementById(field+"qty").value;
        var date=document.getElementById(field+"date").value;
        var hubname=document.getElementById("hublist").value;
        var table=document.getElementById("mytable");
        var obj= table.rows.item(index+1).cells;
        var currentproduct=obj.item(1).innerHTML;
        var pname=obj.item(2).innerHTML;

        const xhttp = new XMLHttpRequest();
       xhttp.onload = function() {
          alert(this.responseText);
          location.reload();


        }
```

```
  xhttp.open("POST", "{{'/addsalesdetails/%s'%name}}");
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname+"&
qty="+qty+"&date="+date);


    }
   }



</script>
</head>
<body>
<div id="menubar">
   <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color:
#e3f2fd;" >
      <div class="navbar-brand" onclick="history.back()">
    <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30"
height="24" class="d-inline-block align-text-top">
   </div>
       <div class="container-fluid">
   <a class="navbar-brand" href="{{'/profile/%s'%name}}">
     <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">
     {{name}}
   </a>
  </div>
      <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
      <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>
      <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>
      <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>
    </nav>


</div>
<div class="sidenav">
  <a href="#" onclick="changenav(this.textContent)">Hub's</a>
  <a href="#" onclick="changenav(this.textContent)">ProductDetails</a>
  <a href="#" onclick="changenav(this.textContent)">SupplierDetails</a>
</div>
<div id="main">

</div>
<div class="form-popup" id="myForm">
  <form  class="form-container" onsubmit="prevent(this)">



    <label for="email"><b>Select Coloumn</b></label><br><br>
    <select name="email" class="form-select" id="opt1" required>
      <option value="" >None</option>
       <option>SUPPLIER NAME</option>
       <option>SUPPLIER LOCATION</option>
       <option>NO OF PRODUCTS</option>
    </select><br><br>

    <label for="psw"><b>VALUE</b></label>
    <input type="text" placeholder="Enter the value" name="psw" id="vl" required>

    <button type="submit" class="btn"
onclick="changeSd(document.getElementById('opt1').value)">Submit</button>
```

```
    <button type="button" class="btn cancel" onclick="closeForm()">Close</button>
  </form>
</div>
<script>
  function prevent(e){
     e.preventDefault();
  }
</script>
<div class="form-popup" id="myForm1">
  <form  class="form-container" onsubmit="prevent(this)" >


    <label for="email"><b>Select Coloumn</b></label><br><br>
    <select name="email" id="opt" class="form-select"
onchange="changeProductDetails(this.value)" required>
      <option value="" >None</option>
       <option value="PRODUCT NAME" >PRODUCT NAME</option>
       <option value="SUPPLIER NAME">SUPPLIER NAME</option>
       <option value="PRICE DETAILS">PRICE DETAILS</option>
       <option value="SALES DETAILS">SALES DETAILS</option>
    </select><br><br>

    <div id="pdform">
    <input id="first" type="password" placeholder="Enter the value" name="psw" required>
    </div><br><br>

    <button type="submit" class="btn"
onclick="changePd(document.getElementById('opt').value)">Submit</button>
    <button type="button" class="btn cancel" onclick="closeForm1()">Close</button>
  </form>
```

```
</div>
</body>
</html>
```

**Hubs.HTML**

```
<h3>Hub List</h3><br><br>
<table class="table" id="mytable">
    <thead>
      <tr>
        <th scope="col">S.no</th>
        <th scope="col">HUB NAME</th>
        <th scope="col">HUB LOCATION</th>
        <th scope="col">HUB MANAGER</th>
        <th scope="col">ACTION</th>
        <th scope="col"></th>
      </tr>
    </thead>
    <tbody>

    </tbody>
  </table>
```

**ProductsDetail.HTML**

```
<h3>Product Detail's</h3><br><br>
<select  id="hublist" class="form-select"  name="hub" onchange="productdetails(this.value)"
>
            <option><p>Select the hub to see the productdetails</p></option>
</select><br><br>
<table class="table" id="mytable">
    <thead>
```

```
      <tr>
        <th scope="col">S.no</th>
        <th scope="col">PRODUCT NAME</th>
        <th scope="col">SUPPLIER NAME</th>
        <th scope="col">PRODUCT PRICE</th>
        <th scope="col">SELLING PRICE</th>
        <th scope="col">DATE</th>
        <th scope="col">QUANTITY</th>
        <th scope="col"> SALED QUANTITY</th>
        <th scope="col"> SALED DATE</th>
        <th scope="col"></th>
        <th scope="col"></th>
      </tr>
    </thead>
    <tbody>

    </tbody>
  </table>
```

**SupplierDetails.HTML**

```
<h3>Supplier Detail's</h3><br><br>
<select id="hublist" class="form-select" name="hub"
onchange="supplierdetails(this.value)">
            <option><p>Select the hub to see the suppliersdetails</p></option>
</select><br><br>
<table class="table" id="mytable">
    <thead>
      <tr>
        <th scope="col">S.no</th>
        <th scope="col">SUPPLIER NAME</th>
        <th scope="col">SUPPLIER LOCATION</th>
        <th scope="col">NO OF PRODUCTS</th>
        <th scope="col"></th>
        <th scope="col"></th>
      </tr>
    </thead>
    <tbody>

    </tbody>
  </table>
```

**BUSINESS LOGIC:**

**This part provide business logic for dash board. It includes FLASK API, IBM DB API, SENDGRID API, IBM COS SDK API FOR OBJECT STORAGE.**

**MAIN.PY:**

```
from flask import Flask,render_template,request,redirect
import json
import PIL.Image as Image
import io
import os
import ObjectStorage
import ibm_db
import TwoStepAuthenticator
import  re

app = Flask(__name__)
Hntry={
    "HubName" :"",
}
    "HubLocation" :"",
    "HubManager":"",
    "ProductDetails":[],
    "SupplierDetails" :[]
}
Pdetails={
"productname" :"",
"suppliername" :"",
"pricedetails" :[],
"salesdetails" :[]
}
Settings={
    "productranking":"",
    "hubranking":"",
    "productalertkl":"",
     "productalertcnt":""
}
pricedetails={
    "productprice":"",
    "sellingprice":"",
    "date":"",
    "qty":""
}
sales={
    "saledqty" :"",
    "saleddate" :"",


}
```

```
Sdetails={
  "suppliername":"",
   "supplierlocation" :"",
   "suppliedproducts":""


}
otp={}
@app.route("/dashboard/<name>")
def dashboard(name):
  data={name:"you can da "+name}
  return render_template("MainBoard.html",name=name,data=json.dumps(data))
@app.route("/profile/<name>")
def profile(name):
   b=ObjectStorage.get_item(name,name+"profile")
   print(b)
   b=b.decode("UTF-8")
   data=json.loads(b)
   return  render_template("profile.html",name=name,data=data)
@app.route("/Analysis/<name>")
def Analysis(name):
   return  render_template("Analysis.html",name=name)
@app.route("/Ranking/<name>")
def Ranking(name):
   return  render_template("Ranking.html",name=name)
@app.route("/HubEntry/<name>")
def HubEntry(name):
   return  render_template("HubEntry.html",name=name)
@app.route("/HubDashBoard/<name>")
def HubDashBoard(name):
   return  render_template("HubDashBoard.html",name=name)
@app.route("/changeprofile/<name>",methods=["POST"])
def changeprofile(name):
  by=request.files['file']
  by.save(by.filename)
  ObjectStorage.multi_part_upload(name,name+"profilepic",os.path.abspath(by.filename))
  os.remove(by.filename)
  b = ObjectStorage.get_item(name, name + "profile")
  print(b)
  b = b.decode("UTF-8")
  data = json.loads(b)
  print(type(data))
  data1=data
  data1["profileImage"]=name+"profilepic"
  print(data1["profileImage"])

  print(data1)
  fl=open(name + "profile","w")
  fl.write(json.dumps(data1))
  fl.close()
  ObjectStorage.multi_part_upload(name, name + "profile", os.path.abspath(name +
"profile"))
  os.remove(os.path.abspath(name + "profile"))
  return render_template("profile.html", name=name, data=data)
@app.route("/navforhubentry/<name>",methods=["POST"])
def navforhubentry(name):
   navnm=request.form["fname"]
   if(navnm=="Add Hub"):
    return render_template("addHub.html",name=name)
   if (navnm == "AddProductDetails"):
       return render_template("addProduct.html",name=name)
   if (navnm == "AddSupplierDetails"):
       return render_template("addSupplier.html",name=name)
   if (navnm == "Hub's"):
       return render_template("Hub's.html",name=name)
   if (navnm == "ProductDetails"):
       return render_template("Productdetails.html",name=name)
   if (navnm == "SupplierDetails"):
       return render_template("Supplierdetails.html",name=name)

@app.route("/hubentry/<name>",methods=["POST"])
def hubentry(name):
   hname=request.form["hname"]
   hloc=request.form["hloc"]
   print(hname)
   print(hloc)
   by=ObjectStorage.get_item(name,name+"hub")
   print(by)
   b = by.decode("UTF-8")
   data = json.loads(b)
   print(type(data))
   Hntry["HubName"]=str(hname)
   Hntry["HubLocation"]=str(hloc)
   data["listofhubs"].append(Hntry)
   print(data)
   f=open(name+"hub","w")
   f.write(json.dumps(data))
   f.close()
   ObjectStorage.multi_part_upload(name,name+"hub",os.path.abspath(name+"hub"))
   os.remove(os.path.abspath(name+"hub"))
   return render_template("HubEntry.html", name=name)
@app.route("/gethublist/<name>",methods=["POST"])
def hublist(name):
   by = ObjectStorage.get_item(name, name + "hub")
   print(by)
   b = by.decode("UTF-8")
   return b
@app.route("/addproduct/<name>",methods=["POST"])
def addproduct(name):
   pname=request.form["pname"]
   sname=request.form["sname"]
   price=request.form["price"]
   sprice=request.form["sprice"]
   qty=request.form["qty"]
   date=request.form["date"]
   hub=request.form["hub"]
   by = ObjectStorage.get_item(name, name + "hub")
   by = by.decode("UTF-8")
   data = json.loads(by)
   print(data)

   for x in data["listofhubs"]:
       if x["HubName"] == hub:
         print(x)

         for y in  x["ProductDetails"]:
             print(y)
    if y["productname"] ==pname and  y["suppliername"]==sname:
        pricedetails["productprice"]=price
        pricedetails["sellingprice"]=sprice
        pricedetails["date"]=date
        pricedetails["qty"]=qty
        x["HubName"]["ProductDetails"]["pricedetails"].append(pricedetails)
        break
    else:
        Pdetails["productname"] = pname
        Pdetails["suppliername"] = sname
        pricedetails["productprice"] = price
        pricedetails["sellingprice"] = sprice
        pricedetails["date"] = date
        pricedetails["qty"] = qty
        Pdetails["pricedetails"].append(pricedetails)
        x["ProductDetails"].append(Pdetails)
        break
if len(x["ProductDetails"]) == 0 :
    Pdetails["productname"] = pname
    Pdetails["suppliername"] = sname
    pricedetails["productprice"] = price
    pricedetails["sellingprice"] = sprice
    pricedetails["date"] = date
    pricedetails["qty"] = qty
    Pdetails["pricedetails"].append(pricedetails)
    x["ProductDetails"].append(Pdetails)

print(x["HubManager"] )
```

```
    print(data)
    f = open(name + "hub", "w")
    f.write(json.dumps(data))
    f.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    print(data)
    os.remove(os.path.abspath(name + "hub"))

    return  render_template("HubEntry.html",name=name)
@app.route("/addsupplier/<name>",methods=["POST"])
def addsupplier(name):
    hub = request.form["hub"]
    sname = request.form["sname"]
    sloc = request.form["sloc"]
    soty = request.form["qty"]
    Sdetails["suppliername"]=sname
    Sdetails["supplierlocation"]=sloc
    Sdetails["suppliedproducts"]=soty
    by=ObjectStorage.get_item(name,name+"hub")
    by=by.decode("UTF-8")
    data=json.loads(by)
    print(data)
    for x in data["listofhubs"]:
        if x["HubName"] == hub:
          x["SupplierDetails"].append(Sdetails)
          break
    f = open(name + "hub", "w")
    f.write(json.dumps(data))
    f.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    print(data)
    os.remove(os.path.abspath(name + "hub"))
    return  render_template("HubEntry.html",name=name)
@app.route("/productdetails/<name>",methods=["POST"])
def productdetails(name):
    hubname=request.form["fname"]
    print(hubname)
    by = ObjectStorage.get_item(name, name + "hub")
    print(by)
    b = by.decode("UTF-8")
    data=json.loads(b)
    senddata = ""
    for x in data["listofhubs"]:
        if x["HubName"] == hubname:
            senddata = json.dumps(x["ProductDetails"])
            print(senddata)
    return senddata
@app.route("/supplierdetails/<name>",methods=["POST"])
def supplierdetails(name):
    hubname=request.form["fname"]
    print(hubname)
    by = ObjectStorage.get_item(name, name + "hub")
    print(by)
    b = by.decode("UTF-8")
    data=json.loads(b)
    senddata=""
    for x in data["listofhubs"]:
        if x["HubName"]==hubname:
            senddata=json.dumps(x["SupplierDetails"])
    return senddata
@app.route("/changepassword/<name>")
def changepassword(name):
    print(name)
    return render_template("passwordchange.html",name=name)
@app.route("/passwordvalidate",methods=["POST"])
def validatepassword():
    password=request.form["pass"]
    username=request.form["name"]
    print(password)
    print(username)
    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7YkTR3uKK3a",
            "", "")
        try:
            qry = f"SELECT  * FROM FZN32689.REGISTRATION"
            stmt = ibm_db.exec_immediate(con, qry)
            result = ibm_db.fetch_both(stmt)
            while result!=False:
                if(result["USERNAME"]==username and result["PASSWORD"]==password):
                    otp[result["USERNAME"]]=TwoStepAuthenticator.generateOTP()
                    TwoStepAuthenticator.send_otp(result["MAILID"],otp[result["USERNAME"]])
                    return ""
                else:
                    result = ibm_db.fetch_both(stmt)
                return "incorrect password"

            except:
                return "incorrect password"
        except:
            return "something went wrong"
@app.route("/verifyotp",methods=["POST"])
def verifyotp():
    print("hii fro verifier of otp")
    mail=request.form["mail"]
    ot=request.form["otp"]
    print(mail)
    print(ot)
    if otp[mail]== ot:
        print("matched")
        return "otp matched"
    print("mismatch")
    return "otp mismatch"
@app.route("/psck",methods=["POST"])
def passwordchecker():
    passwd = request.form["fname"]
    reg = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!#%*?&]{6,20}$"

    # compiling regex
    pat = re.compile(reg)

    # searching regex
    mat = re.search(pat, passwd)
    # validating conditions
    if mat:
        return ""
    else:
        return "password should contain 6 to 20 characters,one special symbol,at least one
uppercase and one lowercase character, at least one number"
@app.route("/passwordchange/<name>",methods=["POST"])
def changed(name):
    newpas=request.form["pass"]
    print(newpas)

    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7Yk
TR3uKK3a",
            '', '')
        qry1 = f"SELECT  * FROM FZN32689.REGISTRATION"
        stmt = ibm_db.exec_immediate(con, qry1)
        qry = ""
        result = ibm_db.fetch_both(stmt)
        username=""
        mailid=""
        phno=""
        subuser=""
        while result != False:
            if result["USERNAME"]==name:
                username=result['USERNAME']
                mailid=result['MAILID']
                phno=result['PHNO']
                subuser=result['SUBUSER' ]
                qry=f"INSERT INTO FZN32689.REGISTRATION (USERNAME,MAILID,PHNO,PASSWORD,SUBUSER) VALUES('{username}','{mailid}','{phno}','{newpas}',0);"
                db='"FZN32689"."REGISTRATION"  '
                un='"USERNAME"'
                nm='"bhawin"'
                qry2=f"DELETE FROM {db} WHERE {un} = {nm};"

                stmt = ibm_db.exec_immediate(con, qry2)
                stmt = ibm_db.exec_immediate(con, qry)
                break
            else:
                result=ibm_db.fetch_both(stmt)

    except Exception as e:
        print(e)
        return "Something went wrong"
    return redirect("http://127.0.0.1:5002/profile/"+name)
@app.route("/removeproductdetails",methods=["POST"])
def removepd():
    hubname=request.form["hubname"]
    prname=request.form["prdname"]
    username=request.form["urname"]
    data=ObjectStorage.get_item(username,username+"hub")
    data=data.decode("UTF -8")
    print(data)
    data=json.loads(data)
    for x in data["listofhubs"]:
        if x["HubName"]==hubname:
          for y in x["ProductDetails"]:
              if y["productname"]==prname:
                  x["ProductDetails"].remove(y)
                  break
    print(data)
    file=open(username+"hub","w")
    file.write(json.dumps(data))
    file.close()

ObjectStorage.multi_part_upload(username,username+"hub",os.path.abspath(username+"hub"))
    os.remove(os.path.abspath(username+"hub"))


    return "Removed success fully refresh your page to see the changes"
@app.route("/removehub/<name>",methods=["POST"])
def removeHub(name):
    hubname=request.form["hub"]
    data = ObjectStorage.get_item(name,name+"hub")
    data=data.decode("UTF -8")
    data=json.loads(data)
    for x in data["listofhubs"]:
        if x["HubName"]==hubname:
            data["listofhubs"].remove(x)
            break
    file = open(name+"hub", "w")
    file.write(json.dumps(data))
    file.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    os.remove(os.path.abspath(name + "hub"))
    return "Removed "+hubname+"successfully reload your page to see the changes"


@app.route("/removesupplier/<name>",methods=["POST"])
def removeSupplier(name):
    supname=request.form["sname"]
    hubname=request.form["hname"]
    data = ObjectStorage.get_item(name,name+"hub")
    data=data.decode("UTF -8")
    data=json.loads(data)
    for x in data["listofhubs"]:
        if x["HubName"]==hubname:
            for t in x["SupplierDetails"]:
                if t["suppliername"] == supname:
                    x["SupplierDetails"].remove(t)
                    break


            break
    file = open(name+"hub", "w")
    file.write(json.dumps(data))
    file.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    os.remove(os.path.abspath(name + "hub"))
    return "Removed "+supname+"successfully reload your page to see the changes"
@app.route("/changepname/<name>",methods=["POST"])
def changePname(name):
    pname=request.form["prdname"]
    olpname=request.form["olpname"]
    hname=request.form["hubname"]
    print(pname)
    print(olpname)
    print(hname)
    data=ObjectStorage.get_item(name,name+"hub")
    data=data.decode("UTF -8")
    data=json.loads(data)
    for x in data["listofhubs"]:
        if x["HubName"]==hname:
            for y in x["ProductDetails"]:
                if  y["productname"] == olpname:
                    y["productname"]=pname
                    file=open(name+"hub","w")
                    file.write(json.dumps(data))
                    file.close()

ObjectStorage.multi_part_upload(name,name+"hub",os.path.abspath(name+"hub"))
                    os.remove(os.path.abspath(name+"hub"))
                    break
            break
    return "changed successfully"
@app.route("/changesname/<name>",methods=["POST"])
def changeSname(name):
    sname=request.form["prdname"]
    olpname=request.form["olpname"]
    hname=request.form["hubname"]
    print(sname)
    print(olpname)
    print(hname)
    data=ObjectStorage.get_item(name,name+"hub")
    data=data.decode("UTF -8")
    data=json.loads(data)
    for x in data["listofhubs"]:
        if x["HubName"]==hname:
            for y in x["ProductDetails"]:
                if  y["productname"] == olpname:
                    y["suppliername"]=sname
                    file=open(name+"hub","w")
                    file.write(json.dumps(data))
                    file.close()

ObjectStorage.multi_part_upload(name,name+"hub",os.path.abspath(name+"hub"))
                    os.remove(os.path.abspath(name+"hub"))
                    break
            break
    return "changed successfully"
@app.route("/addpricedetails/<name>",methods=["POST"])
def addpricedetails(name):
    sname=request.form["prdname"]
    olpname=request.form["olpname"]
    hname=request.form["hubname"]
    price=request.form["price"]
    sprice=request.form["sprice"]
    date=request.form["date"]
    qty=request.form["qty"]
    print(sname)
    print(olpname)
    print(hname)
    data=ObjectStorage.get_item(name,name+"hub")
    data=data.decode("UTF -8")
    data=json.loads(data)
    pricedetails["productprice"]=price
    pricedetails["sellingprice"]=sprice
    pricedetails["date"]=date
    pricedetails["qty"]=qty
    for x in data["listofhubs"]:
        if x["HubName"]==hname:
            for y in x["ProductDetails"]:
                if  y["productname"] == olpname:
                    y["pricedetails"].append(pricedetails)
                    print(pricedetails)
                    file=open(name+"hub","w")
                    file.write(json.dumps(data))
                    file.close()

ObjectStorage.multi_part_upload(name,name+"hub",os.path.abspath(name+"hub"))
                    os.remove(os.path.abspath(name+"hub"))
                    break
            break
    return "Added successfully"
@app.route("/addsalesdetails/<name>",methods=["POST"])
def addsalesdetails(name):
    sname=request.form["prdname"]
    olpname=request.form["olpname"]
    hname=request.form["hubname"]
    date=request.form["date"]
    qty=request.form["qty"]

    data=ObjectStorage.get_item(name,name+"hub")
```

```
data=data.decode("UTF -8")
data=json.loads(data)
sales["saledqty"]=qty
sales["saleddate"]=date
for x in data["listofhubs"]:
    if x["HubName"]==hname:
        for y in x["ProductDetails"]:
            if  y["productname"] == olpname and y["suppliername"]==sname:
                dy = 0;
                mnt = 0;
                yr = 0;
                qt=""
                zin=""
                for z in y["pricedetails"]:
                    dt=z["date"]
                    dt=dt.split("-")
                    print(dt)
                    print(type(int(dt[0])))
                    if  int(dt[0]) > yr:
                        yr=int(dt[0])
                        qt=z["qty"]
                        zin=z
                    if mnt<int(dt[1]):
                        mnt=int(dt[1])
                        qt = z["qty"]
                        zin = z
                    if dy < int(dt[2]):
                        dy = int(dt[2])
                        qt = z["qty"]
                        zin = z
                text = ""
                numbers = ""
                text1 = ""
                numbers1 = ""
                for i in zin["qty"]:
                    if (i.isdigit()):
                        numbers += i
                    else:
                        text += i
                for i in qty:
                    if (i.isdigit()):
                        numbers1 += i
                    else:
                        text1 += i
                if int(numbers) < int(numbers1):
                    return "Invalid input"
                zin["qty"] = str(abs(int(numbers1) - int(numbers))) + text1
                newqty=abs(int(numbers1)-int(numbers))
                by=ObjectStorage.get_item(name,name+"settings")
                by=by.decode("UTF -8")
                by=json.loads(by)
                num=""
                if text !="":
                 for i in by["productalertkl"]:
                    if (i.isdigit()):
                        num += i
                else:
                    num=by["productalertcnt"]
```

```
                    print(num)
                    if newqty<=int(num):
                        by = ObjectStorage.get_item(name, name + "profile")
                        by = by.decode("UTF -8")
                        by = json.loads(by)
                        print(by)
                        TwoStepAuthenticator.message(by["mailId"],"Low stock ","Losw stock  make
an order to meet out the demand product name= "+olpname+"quantity
left="+str(newqty)+text1)
                        by = ObjectStorage.get_item(x["HubManager"], x["HubManager"] + "profile")
                        by = by.decode("UTF -8")
                        by = json.loads(by)
                        print(by)
                        TwoStepAuthenticator.message(by["mailId"], "Low stock ",
                                                     "Losw stock  make an order to meet out the demand product
name= " + olpname + "quantity left=" + str(
                                                         newqty) + text1)


                    y["salesdetails"].append(sales)


                    file=open(name+"hub","w")
                    file.write(json.dumps(data))
                    file.close()

ObjectStorage.multi_part_upload(name,name+"hub",os.path.abspath(name+"hub"))
                    os.remove(os.path.abspath(name+"hub"))
                    break
            break
    return "Added successfully"
@app.route("/changesdname/<name>",methods=["POST"])
def changeSdname(name):
    pname=request.form["prdname"]
    olpname=request.form["olpname"]
    hname=request.form["hubname"]
    print(pname)
    print(olpname)
    print(hname)
    data=ObjectStorage.get_item(name,name+"hub")
    data=data.decode("UTF -8")
    data=json.loads(data)
    for x in data["listofhubs"]:
        if x["HubName"]==hname:
            for y in x["SupplierDetails"]:
                if  y["suppliername"] == olpname:
                    y["suppliername"]=pname
                    file=open(name+"hub","w")
                    file.write(json.dumps(data))
                    file.close()

ObjectStorage.multi_part_upload(name,name+"hub",os.path.abspath(name+"hub"))
                    os.remove(os.path.abspath(name+"hub"))
                    break
            break
    return "changed successfully"
@app.route("/changesdlocation/<name>",methods=["POST"])
def changeSdlocation(name):
    pname=request.form["prdname"]
    olpname=request.form["olpname"]
    hname=request.form["hubname"]
    print(pname)
    print(olpname)
    print(hname)
    data=ObjectStorage.get_item(name,name+"hub")
    data=data.decode("UTF -8")
    data=json.loads(data)
    for x in data["listofhubs"]:
        if x["HubName"]==hname:
            for y in x["SupplierDetails"]:
                if  y["suppliername"] == olpname:
                    y["supplierlocation"]=pname
                    file=open(name+"hub","w")
                    file.write(json.dumps(data))
                    file.close()

ObjectStorage.multi_part_upload(name,name+"hub",os.path.abspath(name+"hub"))
                    os.remove(os.path.abspath(name+"hub"))
                    break
            break
    return "changed successfully"
@app.route("/changesdqty/<name>",methods=["POST"])
def changeSdqty(name):
    pname=request.form["prdname"]
    olpname=request.form["olpname"]
    hname=request.form["hubname"]
    print(pname)
    print(olpname)
    print(hname)
    data=ObjectStorage.get_item(name,name+"hub")
    data=data.decode("UTF -8")
    data=json.loads(data)
    for x in data["listofhubs"]:
        if x["HubName"]==hname:
            for y in x["SupplierDetails"]:
                if  y["suppliername"] == olpname:
                    y["suppliedproducts"]=pname
                    file=open(name+"hub","w")
                    file.write(json.dumps(data))
                    file.close()

ObjectStorage.multi_part_upload(name,name+"hub",os.path.abspath(name+"hub"))
                    os.remove(os.path.abspath(name+"hub"))
                    break
            break
    return "changed successfully"


@app.route("/settings/<name>",methods=["POST"])
def settings(name):
    hr=request.form["hr"]
    pr=request.form["pr"]
    kl = request.form["kl"]
    cnt = request.form["cnt"]
    Settings["productalertkl"]=kl
    Settings["productalertcnt"]=str(cnt)
    Settings["productranking"]=pr
    Settings["hubranking"]=hr
    file=open(name+"settings","w")
    file.write(json.dumps(Settings))
    file.close()
```

```
ObjectStorage.multi_part_upload(name,name+"settings",os.path.abspath(name+"settings"))
    os.remove(os.path.abspath(name+"settings"))



    return "Updated"
if __name__=="__main__":
    app.run(port=5002,debug=True)
```

**ObjectStorage.py**

```
import ibm_boto3
from ibm_botocore.client import Config, ClientError
import PIL.Image as Image
import io



COS_ENDPOINT="https://s3.tok.ap.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID="dRpfBDLhp5Y2FqwqaZHEq6cWeinyufVjZLRz0VNl7Hnj"
COS_INSTANCE_CRN="crn:v1:bluemix:public:cloud-object-storage:global:a/702af44240f54d66ba7adebefb61dd74:21d01580-e4e2-41a3-8589-ef29aaacb70d::"
COS_BUCKET_LOCATION="jp-tok-smart"

cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)
def get_buckets():
    print("Retrieving list of buckets")
    try:
        buckets = cos.buckets.all()
        print(buckets)
        for bucket in buckets:
            print("Bucket Name: {0}".format(bucket.name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve list buckets: {0}".format(e))



def create_bucket(bucket_name):
    print("Creating new bucket: {0}".format(bucket_name))
    try:
        cos.Bucket(bucket_name).create()
        print("Bucket: {0} created!".format(bucket_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to create bucket: {0}".format(e))

def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB

        with open(file_path, "rb") as file_data:
             cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))


def get_item(bucket_name, item_name):
    print("Retrieving item from bucket: {0}, key: {1}".format(bucket_name, item_name))
    try:
        file = cos.Object(bucket_name, item_name).get()
        by=file["Body"].read()
        return by



    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve file contents: {0}".format(e))
```

**TwoStepAuthenticator.py**

```
import sendgrid
from python_http_client.exceptions import HTTPError
import math,random
import apikey
def send_otp(mailId,otp):
    API_KEY = apikey.api_key

    sg = sendgrid.SendGridAPIClient(API_KEY)
    data = {
        "personalizations": [
            {
                "to": [
                    {
                        "email": mailId
                    }
                ],
                "subject": "OTP FROM INVENTORY MANAGEMENT SYSTEM IBM PROJECT"
            }
        ],
        "from": {
            "email": "bhawinjasperbj@gmail.com"
        },
        "content": [
            {
                "type": "text/plain",
                "value": "your otp don't share with any one " + otp
            }
        ]
    }
    try:
        response = sg.client.mail.send.post(request_body=data)
        print(response.status_code)
        print(response.body)
        print(response.headers)
        return ""
    except HTTPError as e:
        print(e.to_dict)
        return "INVALID MAILID"


def generateOTP():
    # Declare a digits variable
    # which stores all digits
    digits = "0123456789"
    OTP = ""

    # length of password can be changed
    # by changing value in range
    for i in range(4):
        OTP += digits[math.floor(random.random() * 10)]
    print(OTP)
    return OTP

def message(mailId,subject,message):
    API_KEY = apikey.api_key

    sg = sendgrid.SendGridAPIClient(API_KEY)
    data = {
        "personalizations": [
            {
                "to": [
                    {
                        "email": mailId
                    }
                ],
                "subject": subject
            }
        ],
        "from": {
            "email": "bhawinjasperbj@gmail.com"
        },
        "content": [
            {
                "type": "text/plain",
                "value": message
            }
        ]
    }
    try:
        response = sg.client.mail.send.post(request_body=data)
        print(response.status_code)
        print(response.body)
        print(response.headers)
        return ""
    except HTTPError as e:
        print(e.to_dict)
        return "INVALID MAILID"
```

**<u>HUB USER DASH BOARD</u>**

**TEMPLATES**

**MainDashBoard.HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Dashboard</title>
    <!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>


<!-- you need to include the shieldui css and js assets in order for the charts to work -->
```

```
<link rel="stylesheet" type="text/css"
href="https://www.shieldui.com/shared/components/latest/css/light-bootstrap/all.min.css" />
<script type="text/javascript"
src="https://www.shieldui.com/shared/components/latest/js/shieldui-all.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.js"></script>
<script>
   var points = JSON.parse('{{ data|safe }}');
    function my(){
      alert(points["{{'%s'%name}}"]);
    }

</script>
<style>
  #Profit-Growth{

  width :50%;
  float :left;
}
#Sales-Growth{

  width:50%;
  float :right;
}
</style>




</head>
<body  >
<div id="menubar">
```

```
  <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;" >

       <div class="container-fluid">

  <a class="navbar-brand" href="{{'/profile/%s'%name}}">

    <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">

    {{name}}

  </a>

 </div>

      <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>

      <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>

      <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>

      <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>

      <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>


  </nav>


</div>

<div id ="content">

 <div id="Profit-Growth">


<canvas id="myChart" style="width:100%;max-width:600px"></canvas>


<script>

var xValues = [50,60,70,80,90,100,110,120,130,140,150];

var yValues = [7,8,8,9,9,9,10,11,14,14,15];


new Chart("myChart", {

 type: "line",

 data: {
  labels: xValues,
  datasets: [{
    fill: false,
    lineTension: 0,
    backgroundColor: "rgba(0,0,255,1.0)",
    borderColor: "rgba(0,0,255,0.1)",
    data: yValues
  }]
 },
 options: {
  legend: {display: false},
  scales: {
    yAxes: [{ticks: {min: 6, max:16}}],
  },
  title: {
    display: true,
    text: "Profit growth"
  }
 }
});
</script><br><br>
</div>
<div id="Sales-Growth">

<canvas id="myChart1" style="width:100%;max-width:600px"></canvas>

<script>
var xValues = ["Italy", "France", "Spain", "USA", "Argentina"];
var yValues = [55, 49, 44, 24, 15];
var barColors = ["red", "green","blue","orange","brown"];

new Chart("myChart1", {
 type: "bar",
 data: {
  labels: xValues,
  datasets: [{
    backgroundColor: barColors,
    data: yValues
  }]
 },
 options: {
  legend: {display: false},
  title: {
    display: true,
    text: "Sales growth"
  }
 }
});
</script>
</div><br><br><br>
<div id="Customer Base">


 <canvas id="myChart2" style="width:100%;max-width:600px"></canvas>

<script>
var xValues = ["Italy", "France", "Spain", "USA", "Argentina"];
var yValues = [55, 49, 44, 24, 15];
var barColors = [
  "#b91d47",
  "#00aba9",
  "#2b5797",
  "#e8c3b9",
  "#1e7145"
];

new Chart("myChart2", {
  type: "pie",
  data: {
    labels: xValues,
    datasets: [{
      backgroundColor: barColors,
      data: yValues
    }]
  },
  options: {
    title: {
      display: true,
      text: "Customer Base "
    }
  }
});
</script>
</div>

</div>
```

```
</body>
</html>
```

**Profile.HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>{{name}}</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
<style>
#dp{
 border-radius:8 px;
}
#profile{
  border-style: solid;
  border-color: rgb(68, 183, 215);
  border-width: 1px;
  width :50%;
  float :left;
}
#settings{
  border-style: solid;
  border-width: 1px;
  border-color: rgb(68, 183, 215);
  width:50%;
  float :right;
}

/* Button used to open the contact form - fixed at the bottom of the page */
.open-button {
  background-color: #555;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  opacity: 0.8;
  position: fixed;
  bottom: 23px;
  right: 28px;
  width: 280px;
}

/* The popup form - hidden by default */
.form-popup {
  display: none;
  position: fixed;
  bottom: 0;
  right: 15px;
  border: 3px solid #f1f1f1;
  z-index: 9;
}
```

```
/* Add styles to the form container */
.form-container {
  max-width: 300px;
  padding: 10px;
  background-color: white;
}

/* Full-width input fields */
.form-container input[type=text], .form-container input[type=password] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  border: none;
  background: #f1f1f1;
}

/* When the inputs get focus, do something */
.form-container input[type=text]:focus, .form-container input[type=password]:focus {
  background-color: #ddd;
  outline: none;
}

/* Set a style for the submit/login button */
.form-container .btn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  width: 100%;
  margin-bottom:10px;
  opacity: 0.8;
}

/* Add a red background color to the cancel button */
.form-container .cancel {
  background-color: red;
}

/* Add some hover effects to buttons */
.form-container .btn:hover, .open-button:hover {
  opacity: 1;
}
</style>
<script>
  function openForm() {
  document.getElementById("myForm").style.display = "block";
}

function closeForm() {
  document.getElementById("myForm").style.display = "none";
}
function openForm1() {
  document.getElementById("myForm1").style.display = "block";
}

function closeForm1() {
  document.getElementById("myForm1").style.display = "none";
```

```
}
function settings(){
     var ck1=document.getElementById("Hday");
     var ck2=document.getElementById("Hweek");
     var ck3=document.getElementById("Hmonth");
     var ck4=document.getElementById("Pday");
     var ck5=document.getElementById("Pweek");
     var ck6=document.getElementById("Pmonth");
     var hr="";
     var pr="";
     var kl=document.getElementById("kilo").value;
     var cnt=document.getElementById("count").value;
     if(ck1.checked == true){
        hr=ck1.value;
     }
     else if(ck2.checked == true){
      hr=ck2.value;
     }else if(ck3.checked == true){
      hr=ck3.value;
     }
     if(ck4.checked == true){
        pr=ck4.value;
     }
     else if(ck5.checked == true){
      pr=ck5.value;
     }else if(ck6.checked == true){
      pr=ck6.value;
     }
     alert("going to make changes");
     const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
         alert(this.responseText);



       }
  xhttp.open("POST", "{{'/settings/%s'%name}}");
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhttp.send("hr="+hr+"&pr="+pr+"&kl="+kl+"&cnt="+cnt);
  }
</script>
</head>
<body>
<div id="menubar">
   <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color:
#e3f2fd;" >
        <div class="container-fluid">
   <div class="navbar-brand" onclick="history.back()">
     <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30"
height="24" class="d-inline-block align-text-top">
   </div>
  </div>
      <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
      <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>
      <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>
      <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>
      <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>


   </nav>
```

```
</div>

<div>

<div class="container" id="profile">

  <img id="dp" src="MAINDASHBOARD\anojprofilepic.png" alt="{{'%s'%name}}" style="width:200px"><hr><br>

  <form action = {{'/changeprofile/%s'%name}} method = "post" enctype="multipart/form-data">

    <input type="file" name="file" />

    <input type = "submit" value="Upload" class="btn btn-primary">

  </form>  <hr><br>

  <span > {{data["mailId"]}} &emsp; <button class="btn btn-primary" onclick="openForm()">change</button></span><hr><br>

  <span > {{data["phNo"]}} &emsp; <button class="btn btn-primary" onclick="openForm1()">change</button></span>

   <hr><br>  <a  href=" {{'/changepassword/%s'%name}}">CHANGE PASSWORD</a>

   <div class="form-popup" id="myForm">

    <form action="/action_page.php" class="form-container">




     <label for="email"><b>Email Id</b></label><br><br>

     <input name="email"  type="email" placeholder="Enter new mail id" class="form-control" ><br>




     <label for="psw"><b>VALUE</b></label>

     <input type="text" placeholder="Enter the otp" name="psw" required>


     <button type="submit" class="btn">Submit</button>

     <button type="button" class="btn cancel" onclick="closeForm()">Close</button>

    </form>

   </div>
```

```
    <div class="form-popup" id="myForm1">
      <form action="/action_page.php" class="form-container">
        <label for="psw"><b>PH NO :</b></label><br>
        <input type="tel" placeholder="Enter the number" name="psw" class="form-control" required><br><br>


        <button type="submit" class="btn">Submit</button>
        <button type="button" class="btn cancel" onclick="closeForm1()">Close</button>
      </form>
    </div><br><br>
    <button class="btn btn-danger" ><a href="http://127.0.0.1:5000">Log Out</a></button>


</div>
<div  class ="container" id="settings">
  <h3>Hub Ranking</h3>
  <input type="checkbox" value="day"   class="form-check-input" id="Hday"> Day</input><br><br>
  <input type="checkbox" value="week"   class="form-check-input" id="Hweek"> Week</input><br><br>
  <input type="checkbox" value="month"   class="form-check-input" id="Hmonth"> Month</input><br><hr>
  <h3>Product Ranking</h3>
  <input type="checkbox" value="day"   class="form-check-input" id="Pday"> Day</input><br><br>
  <input type="checkbox" value="week"   class="form-check-input" id="Pweek"> Week</input><br><br>
  <input type="checkbox" value="month"   class="form-check-input"  id="Pmonth"> Month</input><br><hr>


  <h3>Product Alert</h3>
  <input type="text"   class="form-control" placeholder="For Kilogram " id="kilo"></input><br><br>
```

```
 <input type="number" class="form-control"placeholder="for countable products" id="count"></input><br><br>


 <button class="btn btn-primary" class="form-control" onclick="settings()">Save Changes</button>


</div>
</div>


</body>
</html>
```

**Analysis.HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Analysis</title>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>
<script>
  function getHub1(){
     getHub();
     const xhttp = new XMLHttpRequest();
       xhttp.onload = function() {
           var data=this.responseText;
```

```
        alert(data);
        var obj=document.getElementById("hublist");
        var pr=JSON.parse(data);

        for(let x in pr){
          var opt=document.createElement("option");
          opt.setAttribute("value",pr[x]["HubName"]);
          const optionText = document.createTextNode(pr[x]["HubName"]);
          opt.appendChild(optionText);
          obj.appendChild(opt);
        }

     }
xhttp.open("POST", "{{'/gethublist/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send();
}
function getHub(){

  const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        var data=this.responseText;
        alert(data);
        var obj=document.getElementById("mytable");
        var pr=JSON.parse(data);
        alert(pr[0]["HubName"]);
        for(let x in pr){
         var row = obj.insertRow(-1);
         var cell1 = row.insertCell(0);
         var cell2 = row.insertCell(1);
         var cell3 = row.insertCell(2);
         var cell4 = row.insertCell(3);
         var cell5 = row.insertCell(4);
         var cell6 = row.insertCell(5);
         var salesgrowth=0;
         var profit=0;
         var customergrowth=0;
          const optionText1 = document.createTextNode(pr[x]["HubName"]);
          const optionText2 = document.createTextNode(pr[x]["HubLocation"]);
          const optionText3 = document.createTextNode(x);
          const optionText4 = document.createTextNode(salesgrowth);
          const optionText5 = document.createTextNode(profit);
          const optionText6 = document.createTextNode(customergrowth);
           cell1.appendChild(optionText3);
           cell2.appendChild(optionText1);
           cell3.appendChild(optionText2);
           cell4.appendChild(optionText4);
           cell5.appendChild(optionText5);
           cell6.appendChild(optionText6);


        }


     }
xhttp.open("POST", "{{'/gethublist/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send();
}
</script>
```

```
</head>
<body onload="getHub1()">
<div id="menubar">
   <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;" >
      <div class="navbar-brand" onclick="history.back()">
     <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30" height="24" class="d-inline-block align-text-top">
   </div>
       <div class="container-fluid">
   <a class="navbar-brand" href="{{'/profile/%s'%name}}">
     <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">
     {{name}}
   </a>
  </div>
      <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
      <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>
      <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>
      <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>


   </nav>
   <h3>Select Hub For Product Analysis</h3><br><br>
<select  id="hublist" class="form-select"  name="hub" onchange="productdetails(this.value)" >
         <option><p>Select the hub to see the productdetails</p></option>
</select><br><br>
<table class="table" id="mytable">
  <thead>
   <tr>
     <th scope="col">S.no</th>
```

```
      <th scope="col">HUB NAME</th>
      <th scope="col">HUB LOCATION</th>
      <th scope="col">SALES GROWTH</th>
      <th scope="col">PROFIT</th>
      <th scope="col">CUSTOMER GROWTH</th>

    </tr>
  </thead>
  <tbody>

  </tbody>
</table>

</div>
</body>
</html>
```

**Ranking.HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Ranking</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
```

```
</head>
<script>
  function getHub1(){
    getHub();
    const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
          var data=this.responseText;
          alert(data);
          var obj=document.getElementById("hublist");
          var pr=JSON.parse(data);

          for(let x in pr){
            var opt=document.createElement("option");
            opt.setAttribute("value",pr[x]["HubName"]);
            const optionText = document.createTextNode(pr[x]["HubName"]);
            opt.appendChild(optionText);
            obj.appendChild(opt);
          }

        }
   xhttp.open("POST", "{{'/gethublist/%s'%name}}");
   xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
   xhttp.send();
   }
   function getHub(){

    const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
          var data=this.responseText;
          alert(data);
          var obj=document.getElementById("mytable");
          var pr=JSON.parse(data);
          alert(pr[0]["HubName"]);
          for(let x in pr){
           var row = obj.insertRow(-1);
           var cell1 = row.insertCell(0);
           var cell2 = row.insertCell(1);
           var cell3 = row.insertCell(2);
           var cell4 = row.insertCell(3);

           var rank=1;
            const optionText1 = document.createTextNode(pr[x]["HubName"]);
            const optionText2 = document.createTextNode(pr[x]["HubLocation"]);
            const optionText3 = document.createTextNode(x);
            const optionText4 = document.createTextNode(rank);
             cell1.appendChild(optionText3);
             cell2.appendChild(optionText1);
             cell3.appendChild(optionText2);
             cell4.appendChild(optionText4);

          }

        }
   xhttp.open("POST", "{{'/gethublist/%s'%name}}");
   xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
   xhttp.send();
   }
```

```
</script>

<body onload="getHub1()">

<div id="menubar">

   <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;" >

       <div class="navbar-brand" onclick="history.back()">

     <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30" height="24" class="d-inline-block align-text-top">

   </div>

        <div class="container-fluid">

   <a class="navbar-brand" href="{{'/profile/%s'%name}}">

     <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">

     {{name}}

   </a>

  </div>

       <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>

       <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>

       <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>

       <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>


   </nav>

   <h3>Select Hub For Product Ranking</h3><br><br>

<select  id="hublist" class="form-select"  name="hub" onchange="productdetails(this.value)" >

          <option><p>Select the hub to see the productdetails</p></option>

</select><br><br>

<table class="table" id="mytable">

  <thead>

   <tr>

     <th scope="col">S.no</th>
```

```
      <th scope="col">HUB NAME</th>

      <th scope="col">HUB LOCATION</th>

      <th scope="col">RANK</th>



    </tr>

  </thead>

  <tbody>


  </tbody>

</table>

</div>

</body>

</html>
```

**HubEntry.HTML**

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>HubEntry</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">

<!-- JavaScript Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>

<style>

.sidenav  {
  font-family: "Lato", sans-serif;
}

.sidenav {
  position: relative;
  height: 100%;
  width: 15%;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  padding-top: 20px;
  float: left;
}

.sidenav a {
  padding: 6px 6px 6px 32px;
  text-decoration: none;
  font-size: 20px;
  color: #b7afaf;
  display: block;
}

.sidenav a:hover {
  color: #f1f1f1;
}

#main{
  float: right;
  width: 80%;
}

@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}
</style>
<script>
  function changenav(value){
    const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
          document.getElementById("main").innerHTML=this.responseText;
          if(value=="AddProductDetails" || value=="AddSupplierDetails"){
            getHub();
          }
       }
  xhttp.open("POST", "{{'/navforhubentry/%s'%name}}");
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhttp.send("fname="+value);
  }
   function getHub(){

    const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {

          var data=this.responseText;
```

```
            var obj=document.getElementById("hublist");
            var pr=JSON.parse(data);

            for(let x in pr){
              var opt=document.createElement("option");
              opt.setAttribute("value",pr[x]["HubName"]);
              const optionText = document.createTextNode(pr[x]["HubName"]);
              opt.appendChild(optionText);
              obj.appendChild(opt);
            }

         }
    xhttp.open("POST", "{{'/gethublist/%s'%name}}");
    xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhttp.send();
    }
</script>
</head>
<body>
<div id="menubar">
    <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #e3f2fd;" >
        <div class="navbar-brand" onclick="history.back()">
      <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30" height="24" class="d-inline-block align-text-top">
    </div>
         <div class="container-fluid">
    <a class="navbar-brand" href="{{'/profile/%s'%name}}">
      <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">
      {{name}}
    </a>
  </div>
      <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
      <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>
      <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>
      <a class="navbar-brand" href="{{'/HubDashBoard/%s'%name}}">Hub DashBoard</a>

    </nav>

</div>
<div class="sidenav">
  <a href="#" onclick="changenav(this.textContent)">AddProductDetails</a>
  <a href="#" onclick="changenav(this.textContent)">AddSupplierDetails</a>
</div>
<div id="main">

</div>
</body>
</html>
```

**AddProduct.HTML**

```
<div>
    <h4>Product Details</h4><br><br>



    <form action={{'/addproduct/%s'%name}} method="post">
        <input name="pname" type="text" placeholder="Product Name" class="form-control"><br><br>
        <input name="sname" type="text" placeholder="Supplier Name" class="form-control"><br><br>
        <input name="price" type="text" placeholder="Product price" class="form-control"><br><br>
        <input name="sprice" type="text" placeholder="Selling  price" class="form-control"><br><br>
        <input name="qty" type="text" placeholder="supplied quantity" class="form-control"><br><br>
        <input name="date" type="date" placeholder="Date of supplied" class="form-control"><br><br>
        <select  id="hublist" class="form-select"  name="hub" >
            <option><p>NONE</p></option>
        </select><br><br>
        <input type="submit" class= "btn btn-primary">
    </form>
</div>
```

**AddSupplier.HTML**

```
<div>
    <h4>Supplier Details</h4><br><br>
    <form action={{'/addsupplier/%s'%name}} method="post">
        <input name="sname" type="text" placeholder="Supplier Name" class="form-control"><br><br>
        <input name="sloc" type="text" placeholder="Supplied From" class="form-control"><br><br>
        <input name="qty" type="number" placeholder="Supplied No Of Products" class="form-control"><br><br>
        <select name="hub" id="hublist" class="form-select">
            <option><p>NONE</p></option>
        </select><br><br>
        <input type="submit" class= "btn btn-primary">
    </form>
</div>
```

**HubDashBoard.HTML**

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>HubDashBoard</title>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
<style>
.sidenav  {
  font-family: "Lato", sans-serif;
}

.sidenav {
  position: relative;
  height: 100%;
  width: 15%;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  padding-top: 20px;
  float: left;
}
```

```
.sidenav a {
  padding: 6px 6px 6px 32px;
  text-decoration: none;
  font-size: 20px;
  color: #b7afaf;
  display: block;
}

.sidenav a:hover {
  color: #f1f1f1;
}

#main{
  float: right;
  width: 80%;
}

@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}

/* Button used to open the contact form - fixed at the bottom of the page */
.open-button {
  background-color: #555;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  opacity: 0.8;
  position: fixed;
  bottom: 23px;
  right: 28px;
  width: 280px;
}

/* The popup form - hidden by default */
.form-popup {
  display: none;
  position: fixed;
  bottom: 0;
  right: 15px;
  border: 3px solid #f1f1f1;
  z-index: 9;
}

/* Add styles to the form container */
.form-container {
  max-width: 300px;
  padding: 10px;
  background-color: white;
}

/* Full-width input fields */
.form-container input[type=text], .form-container input[type=password] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  border: none;
  background: #f1f1f1;
}

/* When the inputs get focus, do something */
.form-container input[type=text]:focus, .form-container input[type=password]:focus {
  background-color: #ddd;
  outline: none;
}

/* Set a style for the submit/login button */
.form-container .btn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  width: 100%;
  margin-bottom:10px;
  opacity: 0.8;
}

/* Add a red background color to the cancel button */
.form-container .cancel {
  background-color: red;
}
```

```
/* Add some hover effects to buttons */
.form-container .btn:hover, .open-button:hover {
  opacity: 1;
}

</style>
<script>
  var index;
  function openForm(value) {
    index=Number(value);
    document.getElementById("myForm").style.display = "block";
}

function closeForm() {
  document.getElementById("myForm").style.display = "none";
}
function openForm1(value) {
  index=Number(value);
  document.getElementById("myForm1").style.display = "block";
}

function closeForm1() {
  document.getElementById("myForm1").style.display = "none";
}
function changeProductDetails(x){
  alert(x);
  if(x =="PRICE DETAILS"){
    var y=document.getElementById("pdform");
    var cnt=y.childElementCount;
```

```
   while(cnt>0){


    y.removeChild(y.firstElementChild);
    cnt--;
   }
  var inp=document.createElement("INPUT");
  var inp1=document.createElement("INPUT");
  var inp2=document.createElement("INPUT");
  var inp3=document.createElement("INPUT");
  inp.setAttribute("placeholder","Enter the Product Price");
  inp.setAttribute("type","text");
  inp.setAttribute("id","ppriceet");
  inp2.setAttribute("type","date");
  inp2.setAttribute("id","dateet");
  inp1.setAttribute("placeholder","Enter the Selling Price");
  inp1.setAttribute("type","text");
  inp1.setAttribute("id","spriceet");
  inp3.setAttribute("placeholder","Enter the quantity");
  inp3.setAttribute("type","text");
  inp3.setAttribute("id","qtyet");

  document.getElementById("pdform").appendChild(inp);
  document.getElementById("pdform").appendChild(inp1);
  document.getElementById("pdform").appendChild(inp2);
  document.getElementById("pdform").appendChild(inp3);
}else if(x == "SALES DETAILS"){

  var y=document.getElementById("pdform");
  var cnt=y.childElementCount;
```

```
   while(cnt>0){


    y.removeChild(y.firstElementChild);
    cnt--;
   }
  var inp=document.createElement("INPUT");
  var inp2=document.createElement("INPUT");
  inp.setAttribute("placeholder","Enter the qty");
  inp.setAttribute("type","text");
  inp.setAttribute("id",x+"qty");
  inp2.setAttribute("type","date");
  inp2.setAttribute("id",x+"date");
  y.appendChild(inp);
  y.appendChild(inp2);
} else{
  var y=document.getElementById("pdform");
  var y=document.getElementById("pdform");
  var cnt=y.childElementCount;
   while(cnt>0){


    y.removeChild(y.firstElementChild);
    cnt--;
   }
  var inp=document.createElement("INPUT");
  inp.setAttribute("placeholder","Enter the "+x);
  inp.setAttribute("type","text");
  inp.setAttribute("id",x);
  y.appendChild(inp);
}
```

```
}
  function changenav(value){
  const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("main").innerHTML=this.responseText;
        if(value=="Hub's"){
          getHub();
        }else{
          getHub1();
        }
      }
 xhttp.open("POST", "{{'/navforhubentry/%s'%name}}");
 xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
 xhttp.send("fname="+value);
 }
 function removeHub(hubname){
   alert("Going to Remove "+hubname);
   const xhttp = new XMLHttpRequest();
     xhttp.onload = function() {
         alert(this.responseText);
         location.reload();
       }
 xhttp.open("POST", "{{'/removehub/%s'%name}}");
 xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
 xhttp.send("hub="+hubname);
 }
 function getHub(){
     const xhttp = new XMLHttpRequest();
       xhttp.onload = function() {
           var data=this.responseText;
           alert(data);
           var obj=document.getElementById("mytable");
           var pr=JSON.parse(data);
           alert(pr[0]["HubName"]);
           for(let x in pr){
            var row = obj.insertRow(-1);
             var cell1 = row.insertCell(0);
            var cell2 = row.insertCell(1);
            var cell3 = row.insertCell(2);
            var cell4 = row.insertCell(3);
            var cell5 = row.insertCell(4);
            var cell6 = row.insertCell(5);
             var a=document.createElement("a");

a.setAttribute("href","{{'http://127.0.0.1:5001/subregister/%s/'%(name)}}"+pr[x]["HubName"])
             if(pr[x]["HubManager"]==""){
              const optionText = document.createTextNode("ADD COWORKER");
              a.appendChild(optionText);
             }else{
              const optionText = document.createTextNode("CHANGE COWORKER");
              a.appendChild(optionText);
             }
             var bt=document.createElement("BUTTON");
             bt.setAttribute("class","btn btn-primary");
             bt.setAttribute("value",pr[x]["HubName"])
             bt.addEventListener("click",function(){
```

```
          removeHub(this.value);
        });
        var t = document.createTextNode("Remove");

         bt.appendChild(t);




         const optionText1 = document.createTextNode(pr[x]["HubName"]);
         const optionText2 = document.createTextNode(pr[x]["HubLocation"]);
         const optionText3 = document.createTextNode(pr[x]["HubManager"]);
         const optionText4 = document.createTextNode(x);
         cell1.appendChild(optionText4);
         cell2.appendChild(optionText1);
         cell3.appendChild(optionText2);
         cell4.appendChild(optionText3);
         cell5.appendChild(a);
         cell6.appendChild(bt);

       }

    }
xhttp.open("POST", "{{'/gethublist/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send();
}
function getHub1(){

  const xhttp = new XMLHttpRequest();
```

```
    xhttp.onload = function() {
        var data=this.responseText;
        alert(data);
        var obj=document.getElementById("hublist");
        var pr=JSON.parse(data);

        for(let x in pr){
          var opt=document.createElement("option");
          opt.setAttribute("value",pr[x]["HubName"]);
          const optionText = document.createTextNode(pr[x]["HubName"]);
          opt.appendChild(optionText);
          obj.appendChild(opt);
        }

    }
xhttp.open("POST", "{{'/gethublist/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send();
}
function removeSupplier(suppliername){
  alert("removing "+suppliername);
  var hname=document.getElementById("hublist").value;

  const xhttp = new XMLHttpRequest();
   xhttp.onload = function() {
       alert(this.responseText);
       location.reload();
    }
xhttp.open("POST", "{{'/removesupplier/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("sname="+suppliername+"&hname="+hname);
}
function supplierdetails(value){
 const xhttp = new XMLHttpRequest();
   xhttp.onload = function() {
      var data=this.responseText;
        var obj=document.getElementById("mytable");
        var pr=JSON.parse(data);
        for(let x in pr){
         var row = obj.insertRow(-1);
          var cell1 = row.insertCell(0);
         var cell2 = row.insertCell(1);
         var cell3 = row.insertCell(2);
         var cell4 = row.insertCell(3);
         var cell5 = row.insertCell(4);
         var cell6 = row.insertCell(5)
          var bt=document.createElement("BUTTON");
          bt.setAttribute("class","btn btn-primary");
          bt.setAttribute("value",pr[x]["suppliername"]);

          var t = document.createTextNode("Remove");
         bt.addEventListener("click",function(){
           removeSupplier(this.value)}
           );
          bt.appendChild(t);
          var bt1=document.createElement("BUTTON");
          bt1.setAttribute("class","btn btn-primary");
          bt1.setAttribute("value",x);
```

```
            var t1 = document.createTextNode("Edit");

            bt1.appendChild(t1);

            bt1.addEventListener("click", function(){
             openForm(this.value);
            });
            const optionText1 = document.createTextNode(pr[x]["suppliername"]);
            const optionText2 = document.createTextNode(pr[x]["supplierlocation"]);
            const optionText3 = document.createTextNode(pr[x]["suppliedproducts"]);
            const optionText4 = document.createTextNode(x);
            cell1.appendChild(optionText4);
            cell2.appendChild(optionText1);
            cell3.appendChild(optionText2);
            cell4.appendChild(optionText3);
            cell5.appendChild(bt);
            cell6.appendChild(bt1)

          }
      }
xhttp.open("POST", "{{'/supplierdetails/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("fname="+value);
 }
 function removeproductdetais(prname){
  window.confirm("Are you sure to delete");
   var prdname=prname;
   var urname="{{'%s'%name}}";
   var hubname=document.getElementById("hublist").value;
   const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
          alert(this.responseText);
          location.reload();

       }
 xhttp.open("POST", "/removeproductdetails");
 xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
 xhttp.send("prdname="+prname+"&urname="+urname+"&hubname="+hubname);

 }
 function productdetails(value){
  const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        var data=this.responseText;
        alert(data);

        var obj=document.getElementById("mytable");
          var pr=JSON.parse(data);
          for(let x in pr){
           var row = obj.insertRow(-1);
            var cell1 = row.insertCell(0);
           var cell2 = row.insertCell(1);
           var cell3 = row.insertCell(2);
           var cell4 = row.insertCell(3);
           var cell5 = row.insertCell(4);
           var cell6 = row.insertCell(5);
           var cell7 = row.insertCell(6);
 var cell8 = row.insertCell(7);
 var cell9 = row.insertCell(8);
 var cell10 = row.insertCell(9);
 var cell11 = row.insertCell(10);
  var bt=document.createElement("BUTTON");
  bt.setAttribute("class","btn btn-primary");
  bt.setAttribute("value",pr[x]["productname"])
  bt.addEventListener("click",function(){
      removeproductdetais(this.value);
  });

 var t = document.createTextNode("Remove");

  bt.appendChild(t);
  var bt1=document.createElement("BUTTON");
  bt1.setAttribute("class","btn btn-primary");

  var t1 = document.createTextNode("Edit");
  bt1.setAttribute("value",x)
  bt1.appendChild(t1);
  bt1.addEventListener("click", function(){
   openForm1(this.value);
  });
  const optionText1 = document.createTextNode(x);
  const optionText2 = document.createTextNode(pr[x]["productname"]);
  const optionText3 = document.createTextNode(pr[x]["suppliername"]);
  cell1.appendChild(optionText1);
  cell2.appendChild(optionText2);
  cell3.appendChild(optionText3);
```

```
cell10.appendChild(bt);
var prprice="";
var selprice="";
var datec="";
var qty="";
var mnt=0;
var yr=0;
var date=0;
var saledqty="";
var saleddate="";
for(let y in pr[x]["pricedetails"]){

     var myar=pr[x]["pricedetails"][y]["date"].split("-");

     if(yr<=myar[0]){
        yr=myar[0];
       if (mnt<=myar[1]){
         mnt=myar[1];
         if (date<myar[2]){
           date=myar[2];
           prprice=pr[x]["pricedetails"][y]["productprice"];
           selprice=pr[x]["pricedetails"][y]["sellingprice"];
           datec=pr[x]["pricedetails"][y]["date"]
           qty=pr[x]["pricedetails"][y]["qty"];
         }
       }
     }
}
const optionText4 = document.createTextNode(prprice);
```

```
           const optionText5 = document.createTextNode(selprice);

           const optionText6 = document.createTextNode(datec);

           const optionText7 = document.createTextNode(qty);

           const optionText10 = document.createTextNode(saledqty);

           const optionText11 = document.createTextNode(saleddate);

           cell4.appendChild(optionText4);

           cell5.appendChild(optionText5);

           cell6.appendChild(optionText6);

           cell7.appendChild(optionText7);

           cell11.appendChild(bt1);

           cell8.appendChild(optionText10);

           cell9.appendChild(optionText11);



         }

     }

xhttp.open("POST", "{{'/productdetails/%s'%name}}");

xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhttp.send("fname="+value);

 }

 function changeSd(field){

  alert(field);

    if(field=="SUPPLIER NAME"){


      var pname=document.getElementById("vl").value;

      var hubname=document.getElementById("hublist").value;

      var table=document.getElementById("mytable");

      var obj= table.rows.item(index+1).cells;

      var currentproduct=obj.item(1).innerHTML;
     alert(pname+currentproduct+hubname);
     const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();

     }
xhttp.open("POST", "{{'/changesdname/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);

   }
   else if(field=="SUPPLIER LOCATION"){
     var pname=document.getElementById("vl").value;
     var hubname=document.getElementById("hublist").value;
     var table=document.getElementById("mytable");
     var obj= table.rows.item(index+1).cells;
     var currentproduct=obj.item(1).innerHTML;
     alert(pname+currentproduct+hubname);
     const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();

     }
xhttp.open("POST", "{{'/changesdlocation/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);
```

```
   }
   else if(field=="NO OF PRODUCTS"){
     var pname=document.getElementById("vl").value;
     var hubname=document.getElementById("hublist").value;
     var table=document.getElementById("mytable");
     var obj= table.rows.item(index+1).cells;
     var currentproduct=obj.item(1).innerHTML;
     alert(pname+currentproduct+hubname);
     const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();

     }
xhttp.open("POST", "{{'/changesdqty/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);

   }



   }
   function changePd(field){
 alert(field);
   if(field=="PRODUCT NAME"){

     var pname=document.getElementById(field).value;
     var hubname=document.getElementById("hublist").value;
     var table=document.getElementById("mytable");
     var obj= table.rows.item(index+1).cells;
     var currentproduct=obj.item(1).innerHTML;
     const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();

     }
xhttp.open("POST", "{{'/changepname/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);

   }
   else if(field=="SUPPLIER NAME"){
     var pname=document.getElementById(field).value;
     var hubname=document.getElementById("hublist").value;
     var table=document.getElementById("mytable");
     var obj= table.rows.item(index+1).cells;
     var currentproduct=obj.item(1).innerHTML;
     alert(pname+currentproduct+hubname)
     const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        alert(this.responseText);
        location.reload();

     }
xhttp.open("POST", "{{'/changesname/%s'%name}}");
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname);
```

```
      }
      else if(field=="PRICE DETAILS"){
        var price=document.getElementById("ppriceet").value;
        var sprice=document.getElementById("spriceet").value;
        var date=document.getElementById("dateet").value;
        var qty=document.getElementById("qtyet").value;
        var hubname=document.getElementById("hublist").value;
        var table=document.getElementById("mytable");
        var obj= table.rows.item(index+1).cells;
        var currentproduct=obj.item(1).innerHTML;
        var pname=obj.item(1).innerHTML;
        const xhttp = new XMLHttpRequest();
       xhttp.onload = function() {
          alert(this.responseText);
          location.reload();

        }
   xhttp.open("POST", "{{'/addpricedetails/%s'%name}}");
   xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname+"&
price="+price+"&sprice="+sprice+"&date="+date+"&qty="+qty);

      }
    else {
        var qty=document.getElementById(field+"qty").value;
        var date=document.getElementById(field+"date").value;
        var hubname=document.getElementById("hublist").value;
        var table=document.getElementById("mytable");
```

```
         var obj= table.rows.item(index+1).cells;
         var currentproduct=obj.item(1).innerHTML;
         var pname=obj.item(2).innerHTML;
         const xhttp = new XMLHttpRequest();
        xhttp.onload = function() {
           alert(this.responseText);
           location.reload();

         }
    xhttp.open("POST", "{{'/addsalesdetails/%s'%name}}");
    xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhttp.send("prdname="+pname+"&olpname="+currentproduct+"&hubname="+hubname+"&
qty="+qty+"&date="+date);

       }
     }



</script>
</head>
<body>
<div id="menubar">
    <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color:
#e3f2fd;" >
         <div class="navbar-brand" onclick="history.back()">
       <img src="https://static.thenounproject.com/png/251451-200.png" alt="" width="30"
height="24" class="d-inline-block align-text-top">
    </div>
          <div class="container-fluid">
    <a class="navbar-brand" href="{{'/profile/%s'%name}}">
```

```
      <img src="" alt="" width="30" height="24" class="d-inline-block align-text-top">
      {{name}}
    </a>
  </div>
        <a class="navbar-brand" href="{{'/dashboard/%s'%name}}">Home</a>
        <a class="navbar-brand" href="{{'/Analysis/%s'%name}}">Analysis</a>
        <a class="navbar-brand" href="{{'/Ranking/%s'%name}}">Ranking</a>
        <a class="navbar-brand" href="{{'/HubEntry/%s'%name}}">Hub Entry</a>



    </nav>

</div>
<div class="sidenav">
  <a href="#" onclick="changenav(this.textContent)">Hub's</a>
  <a href="#" onclick="changenav(this.textContent)">ProductDetails</a>
  <a href="#" onclick="changenav(this.textContent)">SupplierDetails</a>
</div>
<div id="main">

</div>
<div class="form-popup" id="myForm">
  <form  class="form-container" onsubmit="prevent(this)">



    <label for="email"><b>Select Coloumn</b></label><br><br>
    <select name="email" class="form-select" id="opt1" required>
      <option value="" >None</option>
       <option>SUPPLIER NAME</option>
```

```
        <option>SUPPLIER LOCATION</option>
        <option>NO OF PRODUCTS</option>
    </select><br><br>

    <label for="psw"><b>VALUE</b></label>
    <input type="text" placeholder="Enter the value" name="psw" id="vl" required>

    <button type="submit" class="btn"
onclick="changeSd(document.getElementById('opt1').value)">Submit</button>
    <button type="button" class="btn cancel" onclick="closeForm()">Close</button>
  </form>
</div>
<script>
  function prevent(e){
     e.preventDefault();
  }
</script>
<div class="form-popup" id="myForm1">
  <form  class="form-container" onsubmit="prevent(this)" >



    <label for="email"><b>Select Coloumn</b></label><br><br>
    <select name="email" id="opt" class="form-select"
onchange="changeProductDetails(this.value)" required>
      <option value="" >None</option>
        <option value="PRODUCT NAME" >PRODUCT NAME</option>
        <option value="SUPPLIER NAME">SUPPLIER NAME</option>
        <option value="PRICE DETAILS">PRICE DETAILS</option>
        <option value="SALES DETAILS">SALES DETAILS</option>
    </select><br><br>
  <div id="pdform">
  <input id="first" type="password" placeholder="Enter the value" name="psw" required>
  </div><br><br>

  <button type="submit" class="btn"
onclick="changePd(document.getElementById('opt').value)">Submit</button>
  <button type="button" class="btn cancel" onclick="closeForm1()">Close</button>
 </form>
</div>
</body>
</html>
```

**Hubs.HTML**

```
<h3>Hub List</h3><br><br>
<table class="table" id="mytable">
  <thead>
   <tr>
    <th scope="col">S.no</th>
    <th scope="col">HUB NAME</th>
    <th scope="col">HUB LOCATION</th>
    <th scope="col">HUB MANAGER</th>
    <th scope="col">ACTION</th>
    <th scope="col"></th>
   </tr>
  </thead>
  <tbody>


  </tbody>
 </table>
```

**ProductDetails.HTML**

```
<h3>Product Detail's</h3><br><br>
<select  id="hublist" class="form-select"  name="hub" onchange="productdetails(this.value)" >
        <option><p>Select the hub to see the productdetails</p></option>
</select><br><br>
<table class="table" id="mytable">
    <thead>
      <tr>
        <th scope="col">S.no</th>
        <th scope="col">PRODUCT NAME</th>
        <th scope="col">SUPPLIER NAME</th>
        <th scope="col">PRODUCT PRICE</th>
        <th scope="col">SELLING PRICE</th>
        <th scope="col">DATE</th>
        <th scope="col">QUANTITY</th>
        <th scope="col"> SALED QUANTITY</th>
        <th scope="col"> SALED DATE</th>
        <th scope="col"></th>
        <th scope="col"></th>
      </tr>
    </thead>
    <tbody>

    </tbody>
  </table>
```

**SupplierDetails.HTML**

```
<h3>Supplier Detail's</h3><br><br>
<select id="hublist" class="form-select" name="hub" onchange="supplierdetails(this.value)">
```

```
            <option><p>Select the hub to see the suppliersdetails</p></option>
</select><br><br>
<table class="table" id="mytable">
    <thead>
      <tr>
        <th scope="col">S.no</th>
        <th scope="col">SUPPLIER NAME</th>
        <th scope="col">SUPPLIER LOCATION</th>
        <th scope="col">NO OF PRODUCTS</th>
        <th scope="col"></th>
        <th scope="col"></th>
      </tr>
    </thead>
    <tbody>

    </tbody>
  </table>
```

## BUSINESS LOGIC:

**This part provide business logic for Hub user dash board. It includes FLASK API, IBM DB API, SENDGRID API, IBM COS SDK API FOR OBJECT STORAGE.**

### MAIN.PY:

```
from flask import Flask, render_template, request, redirect
import json
import PIL.Image as Image
import io
import os
import ObjectStorage
import ibm_db
import TwoStepAuthenticator
import re

app = Flask(__name__)
Hntry = {
    "HubName": "",
    "HubLocation": "",
    "HubManager": "",
    "ProductDetails": [],
    "SupplierDetails": []
}
Pdetails = {
    "productname": "",
    "suppliername": "",
    "pricedetails": [],
    "salesdetails": []
}
pricedetails = {
    "productprice": "",
    "sellingprice": "",
    "date": "",
    "qty": ""
}
sales = {
    "saledqty": "",
    "saleddate": "",


}
```

```
Sdetails = {
    "suppliername": "",
    "supplierlocation": "",
    "suppliedproducts": ""

}
Settings={
    "productranking":"",
    "hubranking":"",
    "productalertkl":"",
     "productalertcnt":""
}
otp = {}



@app.route("/dashboard/<name>")
def dashboard(name):
    data = {name: "you can da " + name}
    return render_template("MainBoard.html", name=name, data=json.dumps(data))



@app.route("/profile/<name>")
def profile(name):
    b = ObjectStorage.get_item(name, name + "profile")
    print(b)
    b = b.decode("UTF-8")
    data = json.loads(b)
    return render_template("profile.html", name=name, data=data)
```

```
@app.route("/Analysis/<name>")
def Analysis(name):
    return render_template("Analysis.html", name=name)



@app.route("/Ranking/<name>")
def Ranking(name):
    return render_template("Ranking.html", name=name)



@app.route("/HubEntry/<name>")
def HubEntry(name):
    return render_template("HubEntry.html", name=name)



@app.route("/HubDashBoard/<name>")
def HubDashBoard(name):
    return render_template("HubDashBoard.html", name=name)



@app.route("/changeprofile/<name>", methods=["POST"])
def changeprofile(name):
    by = request.files['file']
    by.save(by.filename)
    ObjectStorage.multi_part_upload(name, name + "profilepic", os.path.abspath(by.filename))
    os.remove(by.filename)
    b = ObjectStorage.get_item(name, name + "profile")
    print(b)
    b = b.decode("UTF-8")
    data = json.loads(b)
    print(type(data))
    data1 = data
    data1["profileImage"] = name + "profilepic"
    print(data1["profileImage"])

    print(data1)
    fl = open(name + "profile", "w")
    fl.write(json.dumps(data1))
    fl.close()
    ObjectStorage.multi_part_upload(name, name + "profile", os.path.abspath(name + "profile"))
    os.remove(os.path.abspath(name + "profile"))
    return render_template("profile.html", name=name, data=data)



@app.route("/navforhubentry/<name>", methods=["POST"])
def navforhubentry(name):
    navnm = request.form["fname"]
    if (navnm == "Add Hub"):
        return render_template("addHub.html", name=name)
    if (navnm == "AddProductDetails"):
        return render_template("addProduct.html", name=name)
    if (navnm == "AddSupplierDetails"):
        return render_template("addSupplier.html", name=name)
    if (navnm == "Hub's"):
        return render_template("Hub's.html", name=name)
    if (navnm == "ProductDetails"):
        return render_template("Productdetails.html", name=name)
    if (navnm == "SupplierDetails"):
        return render_template("Supplierdetails.html", name=name)



@app.route("/hubentry/<name>", methods=["POST"])
def hubentry(name):
    hname = request.form["hname"]
    hloc = request.form["hloc"]
    print(hname)
    print(hloc)
    by = ObjectStorage.get_item(name, name + "hub")
    print(by)
    b = by.decode("UTF-8")
    data = json.loads(b)
    print(type(data))
    Hntry["HubName"] = str(hname)
    Hntry["HubLocation"] = str(hloc)
    data["listofhubs"].append(Hntry)
    print(data)
    f = open(name + "hub", "w")
    f.write(json.dumps(data))
    f.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    os.remove(os.path.abspath(name + "hub"))
    return render_template("HubEntry.html", name=name)
```

```
@app.route("/gethublist/<name>", methods=["POST"])
def hublist(name):
    b = ObjectStorage.get_item(name, name + "hub")
    print(b)
    b = b.decode("UTF-8")
    b=json.loads(b)
    rtdata=[]
    for x in b["listofhubs"]:
        by=ObjectStorage.get_item(x["ownername"],x["ownername"]+"hub")
        print(by)
        by=by.decode("UTF -8")
        by=json.loads(by)
        for y in by["listofhubs"]:
            print(x)
            print(y)
            if y["HubName"] == x["hubname"]:
                rtdata.append(y)
    return json.dumps(rtdata)


@app.route("/addproduct/<name>", methods=["POST"])
def addproduct(name):
    pname = request.form["pname"]
    sname = request.form["sname"]
    price = request.form["price"]
    sprice = request.form["sprice"]
    qty = request.form["qty"]
    date = request.form["date"]
    hub = request.form["hub"]
```

```
by = ObjectStorage.get_item(name, name + "hub")
by = by.decode("UTF-8")
data = json.loads(by)
print(data)

for x in data["listofhubs"]:
    if x["hubname"] == hub:
        print(x)
        nw=x
        by=ObjectStorage.get_item(x['ownername'],x['ownername']+"hub")
        by=by.decode("UTF -8")
        by=json.loads(by)
        for t in by["listofhubs"]:
            if t["HubName"] == hub:
                x=t
        for y in x["ProductDetails"]:
            print(y)
            if y["productname"] == pname and y["suppliername"] == sname:
                pricedetails["productprice"] = price
                pricedetails["sellingprice"] = sprice
                pricedetails["date"] = date
                pricedetails["qty"] = qty
                x["HubName"]["ProductDetails"]["pricedetails"].append(pricedetails)
                break
            else:
                Pdetails["productname"] = pname
                Pdetails["suppliername"] = sname
                pricedetails["productprice"] = price
                pricedetails["sellingprice"] = sprice
```

```
                    pricedetails["date"] = date
                    pricedetails["qty"] = qty
                    Pdetails["pricedetails"].append(pricedetails)
                    x["ProductDetails"].append(Pdetails)
                    break
            if len(x["ProductDetails"]) == 0:
                Pdetails["productname"] = pname
                Pdetails["suppliername"] = sname
                pricedetails["productprice"] = price
                pricedetails["sellingprice"] = sprice
                pricedetails["date"] = date
                pricedetails["qty"] = qty
                Pdetails["pricedetails"].append(pricedetails)
                x["ProductDetails"].append(Pdetails)
            print(by)
            file=open(nw['ownername']+"hub","w")
            file.write(json.dumps(by))
            file.close()
            ObjectStorage.multi_part_upload(nw['ownername'],nw['ownername']+"hub",
os.path.abspath(nw['ownername']+"hub"))
            print(data)
            os.remove(os.path.abspath(nw['ownername']+"hub"))

            break

    print(data)
    f = open(name + "hub", "w")
    f.write(json.dumps(data))
    f.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    print(data)
    os.remove(os.path.abspath(name + "hub"))

    return render_template("HubEntry.html", name=name)



@app.route("/addsupplier/<name>", methods=["POST"])
def addsupplier(name):
    hub = request.form["hub"]
    sname = request.form["sname"]
    sloc = request.form["sloc"]
    soty = request.form["qty"]
    Sdetails["suppliername"] = sname
    Sdetails["supplierlocation"] = sloc
    Sdetails["suppliedproducts"] = soty
    by = ObjectStorage.get_item(name, name + "hub")
    by = by.decode("UTF-8")
    data = json.loads(by)
    print(data)
    for x in data["listofhubs"]:
        if x["hubname"] == hub:
            print(x)
            nw = x
            by = ObjectStorage.get_item(x['ownername'], x['ownername'] + "hub")
            by = by.decode("UTF -8")
            by = json.loads(by)
            for t in by["listofhubs"]:
                if t["HubName"] == hub:
```

```
                x = t
            x["SupplierDetails"].append(Sdetails)
            print(by)
            file = open(nw['ownername'] + "hub", "w")
            file.write(json.dumps(by))
            file.close()
            ObjectStorage.multi_part_upload(nw['ownername'], nw['ownername'] + "hub",
                                            os.path.abspath(nw['ownername'] + "hub"))
            print(data)
            os.remove(os.path.abspath(nw['ownername'] + "hub"))
            break
    f = open(name + "hub", "w")
    f.write(json.dumps(data))
    f.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    print(data)
    os.remove(os.path.abspath(name + "hub"))
    return render_template("HubEntry.html", name=name)


@app.route("/productdetails/<name>", methods=["POST"])
def productdetails(name):
    hubname = request.form["fname"]
    print(hubname)
    by = ObjectStorage.get_item(name, name + "hub")
    print(by)
    b = by.decode("UTF-8")
    data = json.loads(b)
    senddata=""
    for x in data["listofhubs"]:
        if x["hubname"] == hubname:
            by=ObjectStorage.get_item(x["ownername"],x["ownername"]+"hub")
            by=by.decode("UTF -8")
            by=json.loads(by)
            for y in by["listofhubs"]:
                if y["HubName"] == hubname:
                    senddata = json.dumps(y["ProductDetails"])
                    break
            break

    return senddata



@app.route("/supplierdetails/<name>", methods=["POST"])
def supplierdetails(name):
    hubname = request.form["fname"]
    print(hubname)
    by = ObjectStorage.get_item(name, name + "hub")
    print(by)
    b = by.decode("UTF-8")
    data = json.loads(b)
    senddata = ""
    for x in data["listofhubs"]:
        if x["hubname"] == hubname:
            by = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
            by = by.decode("UTF -8")
            by = json.loads(by)
            for y in by["listofhubs"]:
                if y["HubName"] == hubname:
                    senddata = json.dumps(y["SupplierDetails"])
                    break
            break
    return senddata



@app.route("/changepassword/<name>")
def changepassword(name):
    print(name)
    return render_template("passwordchange.html", name=name)



@app.route("/passwordvalidate", methods=["POST"])
def validatepassword():
    password = request.form["pass"]
    username = request.form["name"]
    print(password)
    print(username)
    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7Yk
TR3uKK3a",
            "", "")
        try:
            qry = f"SELECT  * FROM FZN32689.REGISTRATION"
            stmt = ibm_db.exec_immediate(con, qry)
            result = ibm_db.fetch_both(stmt)
            while result != False:
                if (result["USERNAME"] == username and result["PASSWORD"] == password):
                    otp[result["USERNAME"]] = TwoStepAuthenticator.generateOTP()
                    TwoStepAuthenticator.send_otp(result["MAILID"], otp[result["USERNAME"]])
                    return ""
                else:
                    result = ibm_db.fetch_both(stmt)
            return "incorrect password"

        except:
            return "incorrect password"
    except:
        return "something went wrong"



@app.route("/verifyotp", methods=["POST"])
def verifyotp():
    print("hii fro verifier of otp")
    mail = request.form["mail"]
    ot = request.form["otp"]
    print(mail)
    print(ot)
    if otp[mail] == ot:
        print("matched")
        return "otp matched"
    print("mismatch")
    return "otp mismatch"
```

```
@app.route("/psck", methods=["POST"])
def passwordchecker():
    passwd = request.form["fname"]
    reg = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!#%*?&]{6,20}$"

    # compiling regex
    pat = re.compile(reg)

    # searching regex
    mat = re.search(pat, passwd)

    # validating conditions
    if mat:
        return ""
    else:
        return "password should contain 6 to 20 characters,one special symbol,at least one uppercase and one lowercase character, at least one number"



@app.route("/passwordchange/<name>", methods=["POST"])
def changed(name):
    newpas = request.form["pass"]
    print(newpas)

    try:
        con = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fzn32689;PWD=bPKXp7YkTR3uKK3a",
            "", "")
        qry1 = f"SELECT  * FROM FZN32689.REGISTRATION"
        stmt = ibm_db.exec_immediate(con, qry1)
        qry = ""
        result = ibm_db.fetch_both(stmt)
        username = ""
        mailid = ""
        phno = ""
        subuser = ""
        while result != False:
            if result["USERNAME"] == name:
                username = result['USERNAME']
                mailid = result['MAILID']
                phno = result['PHNO']
                subuser = result['SUBUSER']
                qry = f"INSERT INTO FZN32689.REGISTRATION (USERNAME,MAILID,PHNO,PASSWORD,SUBUSER) VALUES('{username}','{mailid}','{phno}','{newpas}',0);"
                db = '"FZN32689"."REGISTRATION"  '
                un = '"USERNAME"'
                nm = '"bhawin"'
                qry2 = f"DELETE FROM        {db}        WHERE {un} = {nm};"

                stmt = ibm_db.exec_immediate(con, qry2)
                stmt = ibm_db.exec_immediate(con, qry)
                break
            else:
                result = ibm_db.fetch_both(stmt)

    except Exception as e:
        print(e)
        return "Something went wrong"
    return redirect("http://127.0.0.1:5002/profile/" + name)



@app.route("/removeproductdetails", methods=["POST"])
def removepd():
    hubname = request.form["hubname"]
    prname = request.form["prdname"]
    username = request.form["urname"]
    data = ObjectStorage.get_item(username, username + "hub")
    data = data.decode("UTF -8")
    print(data)
    data = json.loads(data)
    for x in data["listofhubs"]:
        if x["hubname"] == hubname:
            b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
            print(b)
            b = b.decode("UTF -8")
            b = json.loads(b)
            for y in b["listofhubs"]:
                if y["HubName"] == hubname:
                    for z in y["ProductDetails"]:
                        if z["productname"] == prname:
                            y["ProductDetails"].remove(z)
                            break;
                    break
            file = open(x["ownername"] + "hub", "w")
            file.write(json.dumps(b))
            file.close()
            ObjectStorage.multi_part_upload(x["ownername"], x["ownername"] + "hub",
                                            os.path.abspath(x["ownername"] + "hub"))
            os.remove(os.path.abspath(x["ownername"] + "hub"))
            break

    print(data)
    file = open(username + "hub", "w")
    file.write(json.dumps(data))
    file.close()
    ObjectStorage.multi_part_upload(username, username + "hub", os.path.abspath(username +
"hub"))
    os.remove(os.path.abspath(username + "hub"))

    return "Removed success fully refresh your page to see the changes"



@app.route("/removehub/<name>", methods=["POST"])
def removeHub(name):
    hubname = request.form["hub"]
    data = ObjectStorage.get_item(name, name + "hub")
    data = data.decode("UTF -8")
    print(data)
    data = json.loads(data)
    for x in data["listofhubs"]:
        if x["hubname"] == hubname:
            b=ObjectStorage.get_item(x["ownername"],x["ownername"]+"hub")
            print(b)
            b=b.decode("UTF -8")
            b=json.loads(b)
            for y in b["listofhubs"]:
                if y["HubName"] == hubname:
                    b["listofhubs"].remove(y)
                    break
            file = open(x["ownername"] + "hub", "w")
            file.write(json.dumps(b))
            file.close()
            ObjectStorage.multi_part_upload(x["ownername"],x["ownername"] + "hub", os.path.abspath(x["ownername"] + "hub"))
            os.remove(os.path.abspath(x["ownername"]+ "hub"))
            break
    file = open(name + "hub", "w")
    file.write(json.dumps(data))
    file.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    os.remove(os.path.abspath(name + "hub"))
    return "Removed " + hubname + "successfully reload your page to see the changes"


@app.route("/removesupplier/<name>", methods=["POST"])
def removeSupplier(name):
    supname = request.form["sname"]
    hubname = request.form["hname"]
    data = ObjectStorage.get_item(name, name + "hub")
    data = data.decode("UTF -8")
    data = json.loads(data)
    for x in data["listofhubs"]:
            if x["hubname"] == hubname:
                b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
                print(b)
                b = b.decode("UTF -8")
                b = json.loads(b)
                for y in b["listofhubs"]:
                    if y["HubName"] == hubname:
                        for z in y["SupplierDetails"]:
                            if z["suppliername"] == supname:
                                y["SupplierDetails"].remove(z)
                                break;
                        break
                file = open(x["ownername"] + "hub", "w")
                file.write(json.dumps(b))
                file.close()
                ObjectStorage.multi_part_upload(x["ownername"], x["ownername"] + "hub",
                                                os.path.abspath(x["ownername"] + "hub"))
                os.remove(os.path.abspath(x["ownername"] + "hub"))
                break

            break
    file = open(name + "hub", "w")
    file.write(json.dumps(data))
    file.close()
    ObjectStorage.multi_part_upload(name, name + "hub", os.path.abspath(name + "hub"))
    os.remove(os.path.abspath(name + "hub"))
    return "Removed " + supname + "successfully reload your page to see the changes"



@app.route("/changepname/<name>", methods=["POST"])
def changePname(name):
```

```
pname = request.form["prdname"]
olpname = request.form["olpname"]
hname = request.form["hubname"]
print(pname)
print(olpname)
print(hname)
data = ObjectStorage.get_item(name, name + "hub")
data = data.decode("UTF -8")
data = json.loads(data)
for x in data["listofhubs"]:
    if x["hubname"] == hname:
        b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
        print(b)
        b = b.decode("UTF -8")
        b = json.loads(b)
        for y in b["listofhubs"]:
            if y["HubName"] == hname:
                for z in y["ProductDetails"]:
                    if z["productname"] == olpname:
                        z["productname"] = pname
                        break;
                break
        file = open(x["ownername"] + "hub", "w")
        file.write(json.dumps(b))
        file.close()
        ObjectStorage.multi_part_upload(x["ownername"], x["ownername"] + "hub",
                                        os.path.abspath(x["ownername"] + "hub"))
        os.remove(os.path.abspath(x["ownername"] + "hub"))
        break
    return "changed successfully"



@app.route("/changesname/<name>", methods=["POST"])
def changeSname(name):
    sname = request.form["prdname"]
    olpname = request.form["olpname"]
    hname = request.form["hubname"]
    print(sname)
    print(olpname)
    print(hname)
    data = ObjectStorage.get_item(name, name + "hub")
    data = data.decode("UTF -8")
    data = json.loads(data)
    for x in data["listofhubs"]:
        if x["hubname"] == hname:
            b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
            print(b)
            b = b.decode("UTF -8")
            b = json.loads(b)
            for y in b["listofhubs"]:
                if y["HubName"] == hname:
                    for z in y["ProductDetails"]:
                        if z["productname"] == olpname:
                            z["suppliername"] = sname
                            break;
                    break
            file = open(x["ownername"] + "hub", "w")
            file.write(json.dumps(b))
            file.close()
            ObjectStorage.multi_part_upload(x["ownername"], x["ownername"] + "hub",
                                            os.path.abspath(x["ownername"] + "hub"))
            os.remove(os.path.abspath(x["ownername"] + "hub"))

            break
    return "changed successfully"



@app.route("/addpricedetails/<name>", methods=["POST"])
def addpricedetails(name):
    sname = request.form["prdname"]
    olpname = request.form["olpname"]
    hname = request.form["hubname"]
    price = request.form["price"]
    sprice = request.form["sprice"]
    date = request.form["date"]
    qty = request.form["qty"]
    print(sname)
    print(olpname)
    print(hname)
    data = ObjectStorage.get_item(name, name + "hub")
    data = data.decode("UTF -8")
    data = json.loads(data)
    pricedetails["productprice"] = price
    pricedetails["sellingprice"] = sprice
    pricedetails["date"] = date
    pricedetails["qty"] = qty
    for x in data["listofhubs"]:
        if x["hubname"] == hname:
            b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
            print(b)
            b = b.decode("UTF -8")
            b = json.loads(b)
            ne=x
            for x in b["listofhubs"]:
                if x["HubName"] == hname:
                    for y in x["ProductDetails"]:
                        if y["productname"] == olpname:
                            y["pricedetails"].append(pricedetails)
                            print(pricedetails)
                            file = open(ne["ownername"] + "hub", "w")
                            file.write(json.dumps(b))
                            file.close()
                            ObjectStorage.multi_part_upload(ne["ownername"],ne["ownername"]        +
"hub", os.path.abspath(ne["ownername"] + "hub"))
                            os.remove(os.path.abspath(ne["ownername"]+ "hub"))
                            break
            break
    return "Added successfully"



@app.route("/addsalesdetails/<name>", methods=["POST"])
def addsalesdetails(name):
    sname = request.form["prdname"]
    olpname = request.form["olpname"]
    hname = request.form["hubname"]
```

```
date = request.form["date"]
qty = request.form["qty"]

data = ObjectStorage.get_item(name, name + "hub")
data = data.decode("UTF -8")
data = json.loads(data)
sales["saledqty"] = qty
sales["saleddate"] = date
for x in data["listofhubs"]:
    if x["hubname"] == hname:
        b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
        print(b)
        b = b.decode("UTF -8")
        b = json.loads(b)
        ne = x
        for x in b["listofhubs"]:
            if x["HubName"] == hname:
             for y in x["ProductDetails"]:
              if y["productname"] == olpname:
                if y["productname"] == olpname and y["suppliername"] == sname:
                    dy = 0;
                    mnt = 0;
                    yr = 0;
                    qt = ""
                    zin = ""
                    for z in y["pricedetails"]:
                        dt = z["date"]
                        dt = dt.split("-")
                        print(dt)
                        print(type(int(dt[0])))
                        if int(dt[0]) > yr:
                            yr = int(dt[0])
                            qt = z["qty"]
                            zin = z
                        if mnt < int(dt[1]):
                            mnt = int(dt[1])
                            qt = z["qty"]
                            zin = z
                        if dy < int(dt[2]):
                            dy = int(dt[2])
                            qt = z["qty"]
                            zin = z
                    text = ""
                    numbers = ""
                    text1 = ""
                    numbers1 = ""
                    for i in zin["qty"]:
                        if (i.isdigit()):
                            numbers += i
                        else:
                            text += i
                    for i in qty:
                        if (i.isdigit()):
                            numbers1 += i
                        else:
                            text1 += i

                    if int(numbers)<int(numbers1):
```

```
                            return "Invalid input"
                        zin["qty"] = str(abs(int(numbers1) - int(numbers))) + text1
                        newqty = abs(int(numbers1) - int(numbers))
                        by = ObjectStorage.get_item(name, name + "settings")
                        by = by.decode("UTF -8")
                        by = json.loads(by)
                        num = "0"
                        if text != "":
                            for i in by["productalertkl"]:
                                if (i.isdigit()):
                                    num += i
                        else:
                            num = by["productalertcnt"]
                        print(num)
                        if newqty <= int(num):
                            by = ObjectStorage.get_item(name, name + "profile")
                            by = by.decode("UTF -8")
                            by = json.loads(by)
                            print(by)
                            TwoStepAuthenticator.message(by["mailId"], "Low stock ",
                                                        "Losw stock   make an order to meet out the demand
product name= " + olpname + "quantity left=" + str(
                                                            newqty) + text1)
                            by = ObjectStorage.get_item(x["ownername"], x["ownername"] + "profile")
                            by = by.decode("UTF -8")
                            by = json.loads(by)
                            print(by)
                            TwoStepAuthenticator.message(by["mailId"], "Low stock ",
                                                        "Losw stock   make an order to meet out the demand
product name= " + olpname + "quantity left=" + str(
                                                              newqty) + text1)

                    y["salesdetails"].append(sales)
                    print(pricedetails)
                    file = open(ne["ownername"] + "hub", "w")
                    file.write(json.dumps(b))
                    file.close()
                    ObjectStorage.multi_part_upload(ne["ownername"], ne["ownername"] + "hub",
                                                     os.path.abspath(ne["ownername"] + "hub"))
                    os.remove(os.path.abspath(ne["ownername"] + "hub"))
                    break
            break
    return "Added successfully"



@app.route("/changesdname/<name>", methods=["POST"])
def changeSdname(name):
    pname = request.form["prdname"]
    olpname = request.form["olpname"]
    hname = request.form["hubname"]
    print(pname)
    print(olpname)
    print(hname)
    data = ObjectStorage.get_item(name, name + "hub")
    data = data.decode("UTF -8")
    data = json.loads(data)
    for x in data["listofhubs"]:
     if x["hubname"] == hname:
      b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
      print(b)
      b = b.decode("UTF -8")
      b = json.loads(b)
      for z in b["listofhubs"]:
        if z["HubName"] == hname:
            for y in z["SupplierDetails"]:
                if y["suppliername"] == olpname:
                    y["suppliername"] = pname
                    file = open(x["ownername"] + "hub", "w")
                    file.write(json.dumps(b))
                    file.close()
                    ObjectStorage.multi_part_upload(x["ownername"], x["ownername"] + "hub",
                                                     os.path.abspath(x["ownername"] + "hub"))
                    os.remove(os.path.abspath(x["ownername"] + "hub"))
                    break
            break
    return "changed successfully"


@app.route("/changesdlocation/<name>", methods=["POST"])
def changeSdlocation(name):
    pname = request.form["prdname"]
    olpname = request.form["olpname"]
    hname = request.form["hubname"]
    print(pname)
    print(olpname)
    print(hname)
    data = ObjectStorage.get_item(name, name + "hub")
    data = data.decode("UTF -8")
    data = json.loads(data)
    for x in data["listofhubs"]:
     if x["hubname"] == hname:
        b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
        print(b)
        b = b.decode("UTF -8")
        b = json.loads(b)
        for z in b["listofhubs"]:
         if z["HubName"] == hname:
            for y in z["SupplierDetails"]:
                if y["suppliername"] == olpname:
                    y["supplierlocation"] = pname
                    file = open(x["ownername"] + "hub", "w")
                    file.write(json.dumps(b))
                    file.close()
                    ObjectStorage.multi_part_upload(x["ownername"], x["ownername"] + "hub",
                                                     os.path.abspath(x["ownername"] + "hub"))
                    os.remove(os.path.abspath(x["ownername"] + "hub"))
                    break
            break
    return "changed successfully"


@app.route("/changesdqty/<name>", methods=["POST"])
def changeSdqty(name):
    pname = request.form["prdname"]
    olpname = request.form["olpname"]
    hname = request.form["hubname"]
    print(pname)
    print(olpname)
    print(hname)
    data = ObjectStorage.get_item(name, name + "hub")
    data = data.decode("UTF -8")
    data = json.loads(data)
    for x in data["listofhubs"]:
     for x in data["listofhubs"]:
        if x["hubname"] == hname:
            b = ObjectStorage.get_item(x["ownername"], x["ownername"] + "hub")
            print(b)
            b = b.decode("UTF -8")
            b = json.loads(b)
            for z in b["listofhubs"]:
             if z["HubName"] == hname:
              for y in z["SupplierDetails"]:
                if y["suppliername"] == olpname:
                    y["suppliedproducts"] = pname
                    file = open(x["ownername"] + "hub", "w")
                    file.write(json.dumps(b))
                    file.close()
                    ObjectStorage.multi_part_upload(x["ownername"], x["ownername"] + "hub",
                                                    os.path.abspath(x["ownername"] + "hub"))
                    os.remove(os.path.abspath(x["ownername"] + "hub"))
                    break
            break
    return "changed successfully"
@app.route("/settings/<name>",methods=["POST"])
def settings(name):
    hr=request.form["hr"]
    pr=request.form["pr"]
    kl = request.form["kl"]
    cnt = request.form["cnt"]
    Settings["productalertkl"]=kl
    Settings["productalertcnt"]=str(cnt)
    Settings["productranking"]=pr
    Settings["hubranking"]=hr
    file=open(name+"settings","w")
    file.write(json.dumps(Settings))
    file.close()
ObjectStorage.multi_part_upload(name,name+"settings",os.path.abspath(name+"settings"))
    os.remove(os.path.abspath(name+"settings"))
    return "Updated"

if __name__ == "__main__":
    app.run(port=5012, debug=True)
```

**ObjectStorage.py**

```
import ibm_boto3
from ibm_botocore.client import Config, ClientError
import PIL.Image as Image
import io


COS_ENDPOINT="https://s3.tok.ap.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID="dRpfBDLhp5Y2FqwqaZHEq6cWeinyufVjZLRz0VNl7Hnj"
```

```
COS_INSTANCE_CRN="crn:v1:bluemix:public:cloud-object-storage:global:a/702af44240f54d66ba7adebefb61dd74:21d01580-e4e2-41a3-8589-ef29aaacb70d::"

COS_BUCKET_LOCATION="jp-tok-smart"


cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)
def get_buckets():
    print("Retrieving list of buckets")
    try:
        buckets = cos.buckets.all()
        print(buckets)
        for bucket in buckets:
            print("Bucket Name: {0}".format(bucket.name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve list buckets: {0}".format(e))



def create_bucket(bucket_name):
    print("Creating new bucket: {0}".format(bucket_name))
    try:
        cos.Bucket(bucket_name).create()
        print("Bucket: {0} created!".format(bucket_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to create bucket: {0}".format(e))

def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB

        with open(file_path, "rb") as file_data:
             cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))



def get_item(bucket_name, item_name):
    print("Retrieving item from bucket: {0}, key: {1}".format(bucket_name, item_name))
    try:
        file = cos.Object(bucket_name, item_name).get()

        by=file["Body"].read()
        return by



    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve file contents: {0}".format(e))
```

**TwoStepAuthenticator.py**

```
import sendgrid
from python_http_client.exceptions import HTTPError
import math,random
import apikey

def send_otp(mailId,otp):
    API_KEY = apikey.api_key
    sg = sendgrid.SendGridAPIClient(API_KEY)
    data = {
        "personalizations": [
            {
                "to": [
                    {
                        "email": mailId
                    }
                ],
                "subject": "OTP FROM INVENTORY MANAGEMENT SYSTEM IBM PROJECT"
            }
        ],
        "from": {
            "email": "bhawinjasperbj@gmail.com"
        },
        "content": [
            {
                "type": "text/plain",
                "value": "your otp don't share with any one " + otp
            }
        ]
    }
    try:
        response = sg.client.mail.send.post(request_body=data)
        print(response.status_code)
        print(response.body)
        print(response.headers)
        return ""
    except HTTPError as e:
        print(e.to_dict)
        return "INVALID MAILID"



def generateOTP():
    # Declare a digits variable
    # which stores all digits
    digits = "0123456789"
    OTP = ""

    # length of password can be changed
    # by changing value in range
    for i in range(4):
        OTP += digits[math.floor(random.random() * 10)]
    print(OTP)
    return OTP



def message(mailId,subject,message):
    API_KEY = apikey.api_key

    sg = sendgrid.SendGridAPIClient(API_KEY)
    data = {
        "personalizations": [
            {
                "to": [
                    {
                        "email": mailId
                    }
                ],
                "subject": subject
            }
        ],
        "from": {
            "email": "bhawinjasperbj@gmail.com"
        },
        "content": [
            {
                "type": "text/plain",
                "value": message
            }
        ]
    }
    try:
        response = sg.client.mail.send.post(request_body=data)
        print(response.status_code)
        print(response.body)
        print(response.headers)
        return ""
    except HTTPError as e:
        print(e.to_dict)
        return "INVALID MAILID"
```