INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

IBM PROJECT REPORT

Submitted by

AHAMED JAASIR	**110119205010**
MOHAMED HUSSAIN KANI	**110119205022**
MOHIEDDIN ABDUL QADHAR	**110119205032**
SHEIK AMANULLAH	**110119205036**

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING

ANNA UNIVERSITY, CHENNAI 600 025

DECEMBER 2022

ANNA UNIVERSITY, CHENNAI – 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“INVENTORY MANAGEMENT SYSTEM FOR RETAILERS”** is the Bonafede work of **“AHAMED JAASIR (110119205010), MOHAMED HUSSAIN KANI (110119205022), MOHIEDDIN ABDUL QADHAR (110119205032), SHEIK AMANULLAH (110119205036)”** who carried out project work under my supervision

SIGNATURE	**SIGNATURE**
Dr. ARIF ABDUL RAHMAN	**Ms. RAEESHATHUL HAFE ELA K R**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of Information Technology	Department of Information Technology
Aalim Muhammed Salegh	Aalim Muhammed Salegh
College of Engineering	College of Engineering,
Muthapudupet, Avadi IAF,	Muthapudupet,Avadi IAF,
Chennai 600 055.	Chennai 600 055.

CERFICATE OF EVALUATION

COLLEGE NAME	:	AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING
BRANCH	:	INFORMATION TECHNOLOGY
PROJECT TITLE	:	INVENTORY MANAGEMENT SYSTEM FOR RETAILERS
TEAM ID	:	PNT2022TMID24036

NAME	ROLL NO	CONTRIBUTION
AHAMED JAASIR	110119205010	CREATE DATABASE IN IBM DB2
MOHAMED HUSSAIN KANI	110119205022	CREATE DOCKER FILE AND PUSH TO RESPECTED REPOSITORIES. POST IMAGES TO CONTAINER REGISTRY
MOHIEDDIN ABDUL QADHAR	110119205032	CREATE KUBERNETE SERVICE AND DEPLOY IT
SHEIK AMANULLAH	110119205036	CREATE SOURCE CODE FOR THE FRONT AND BACKEND

The report of this project is submitted by the above students in partial fulfillment for the award of Bachelor of Technology in **INFORMATION TECHNOLOGY** of Anna University are evaluated and confirmed to report of the work done by the above students during the academic year of 2022-2023

This report work is submitted for the Anna University project viva voce work
Held on …………….

…………………………………..　　　　　　　　…………………………………

INTERNAL EXAMINER　　　　　　　　**EXTERNAL EXAMINER**

ABSTRACT

- Inventory management system which is helpful for the business operators, where shopkeeper keep the records of purchase and sales. Mismanaged inventory means disappointed customers, too much cash tied up in slower sale and warehouses.

- This inventory is eliminated paperwork, human faults, manual delay and speed up process. This inventory management system will have the ability to track sales and available inventory, tells a shopkeeper when it's time to reorder and how much to purchase.

- Inventory management system is windows application developed for windows operating systems which focused in the area of inventory control and generate.

- 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 to apply an agent system for automation of inventory management processes. Inventory management system (IMS) use for a departmental store.

- This system can be used to store the details of the inventory based on the sale details, generate sale and inventory report periodically etc. this is one integrated system that contains both the user component (used by sales persons , sales managers inventory managers) and the admin component (used by the administrators for performing admin level function such as adding new item to the inventory) etc

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

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

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

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

1.2PURPOSE

- The Inventory Management System is a real-time inventory database capable of connecting multiple stores. This can be used to track the inventory of a single store or to manage the delivery of stock between several branches of a larger franchise. However, the system merely records sales and restocking data and provides warning of low stock at any location through email at a specified interval.

- The goal is to reduce the stress of tracking rather than to holder all store maintenance. Further features may consist of the ability to create reports of sales, but again the explanation is left to the management. In addition, since theft does occasionally occur, the system provides solutions for confirming the store inventory and for correcting stock quantities.

- Production units use an inventory management system to reduce their transport costs. The system is used to track products and parts as they are transported from a seller to a storeroom, between storerooms, and finally to a retail location or directly to a customer.

The inventory management system is used for various purposes, including:

- ✓ Maintaining and recording the information between too much and too little inventory in the company.
- ✓ Keep track of inventories as it is transported between different locations.
- ✓ Recording product information in a warehouse or other location.
- ✓ Having a record of Picking, packing, and selling products from a warehouse.
- ✓ Reduction of product obsolescence and decay.
- ✓ Avoiding out-of-stock situations.

- Inventory management keeps many retailers up at night, and for good reason: staying on top of your store's stock levels is a balancing act that can make or break your sales and customer satisfaction.

- Too much stock on hand ties up your capital and can end up killing your margins if you decide to mark down unsold products. But not having enough merchandise is just as bad and can lead to lost sales as well as lower customer satisfaction and loyalty.

- So how can you can get stock levels *just right* in your store? Well, each retailer has different inventory needs, so there aren't any silver bullet for this. There are, however, steps and best practices that you can implement to figure out the right product mix for your shop. And that's exactly what this guide is for.

- Below you'll find tips, how-tos, and examples to help you win at inventory management. From picking the right solution and entering your products, to tracking stock levels and automating parts of the process, this guide has you covered.

- Whether you're choosing your inventory software for the first time or you already have a system but are looking to improve and further optimize it, you're bound to pick up something useful from this handy guide

CHAPTER 2

LITERATURE SURVEY

2.1EXISTING SYSTEM

- There is a few Inventories Management System available in the market. After doing my research, I have come to know that most of them are limited to few products. Some others are lacking in good UI. Marketing points are not much focused on increasing sales.

- Customer management system and Inventory Management system can’t be linked due to different organization which leads to compromising the client satisfaction level. Most of them are not using the cloud computing concept but we are trying to develop such a system that is for everyone rather than for only big companies or for a small organization.

- Most of them are expensive to use and their maintenance is generally not cheap. Our system is Pay-as-per-Use.

2.2REFERENCE

- Inventory management is considered as major concerns of every organization. In inventory holding, many steps are taken by managers that result a cost involved in this row. This cost may not be constant in nature during time horizon in which perishable stock is held. To investigate on such a case, Taygi (2014) proposes an optimization of inventory model where items deteriorate in stock conditions.

- In this paper, based on a real-world case study for a municipal district in Tehran, a multi- objective mathematical model is developed for the location-distribution problem. The proposed model considers the role of demand in an urban area, which might be affected by neighbor wards. Integrating decision making process for a disaster helps to improve a better relief operation during response phase of disaster management cycle. In the proposed approach, Esmaeili (2014) says, a proactive damage estimation method is used to estimate demands for the district based on worst-case scenario of earthquake in Tehran.

- This paper deals with the application of six most potential preference ranking methods for selecting the best FMS for a given manufacturing organization. Chatterjeea and

Chakraborty (2014) say, it is observed that although the performances of these six methods are almost similar, ORESTE method slightly outperforms thioethers.

- It is particularly applicable to those situations where the decision maker is unable to provide crisp evaluation data and attribute weight. Ulrich and Pearson (1998) introduce approaches for the integration of the Quality Function Deployment method as well as feedback with system components for computer aided product development. The integration is based on information models representing product, process, and factory information.

- Pastore and Martin (2012) study was to examine students' perceptions of designing and developing mobile based instructions by interviewing and surveying of graduate students. Results of the survey and qualitative data analysis indicated that usability was a key issue on the mobile device. Users enjoyed quick access, good organization, user control, single column layouts, and large links/buttons. These findings contribute to the literature base on the design and development of mobile based instruction.

- Norman E (2012) discusses, while existing factors identified in the literature were found to be present in the context of today's design program, the critical perspective of this study recontextualized these factors, along with the identification of new or underrepresented factors. Taking on the perspective of a student's experience of pedagogy foregrounds issues of uncertainty and ambiguity, highlighting the social interactions between fellow students, and the role of communication and individual effort in learning to think in a more designedly way.

- A design literature discusses the role of the studio and its related pedagogy in the development of design thinking. Scholars in a variety of design disciplines pose a number of factors that potentially affect this development process, but a full understanding of these factors as experienced from a critical pedagogy or student perspective is lacking. In this study, Gray (2013) explains the experiences of six first-year design students were examined as they evolved in their conceptions of design.

- Didonet and Díaz, (2012) explains, the supply chain management studies have verified that integration and collaboration in the supply chain can provide important benefits to the companies involved. Among these benefits are added value, the creation of efficiencies and client, which are represented by the reduction in inventories, improvements in service delivery and quality and shorter product development cycles.

- Zabala (2012) investigates whether decisions considered as common in new product development literature are also valid in a region characterized by traditional industries.

The author aims to link the theoretical and empirical fields in the context of new product development and product innovation management.

- Lebar (2014) reports the results of a survey on the use of innovation management techniques with the potential to improve effectiveness of new product development, and customer satisfaction. Failure mode and effects analysis was found as the most applied IMT in Slovene firms with the highest perceived utility potential to reduce development costs and improve customer satisfaction.

- Nezhad (2013) employed the decision on belief (DOB) approach for fault detection in univariate process control. The concept of DOB and its application in decision making problems were introduced, and then methodology of modeling fault detection in statistical process control by DOB approach was discussed.

2.3PROBLEM STATEMENT DEFINITION

- Once the planning and analysis of the project are Done, the design phase begins. The goal of system design is to transform the information collected about the project into the blueprint structure which will serve as a base while constructing the system. It is an unwieldy process as most errors are introduced in this phase.

- However, if the error gets unnoticed in a later process it may become difficult to track them down.

PROPERTIES:

ADMIN:

Primarily, the user who will interact with the system will be the administrator of the institution assigned to take care of all data transactions and insertion or update. It will have to go through an authorization process of login and logout. It will have the ability to create storage records, add inventory details, item details, Orders, Shipment details and take care of the development and maintenance of the application.

- **Name:** Name of Admin.

- **Admin ID:** It will be a unique value that will act as the primary key and will be the same as the employee id in the company.

- **Email ID:** Employee address is also an attribute that helps to get more about the employee.

MANAGER:

Second, the user who will interact with the system will be the Manager of the institution assigned to take care of management services. It will have to go through an authorization process of and logout. It will have the ability to create storage records, add inventory details, item details, Orders, Shipment details and take care of notification, and can see reports and other business-related data.

- **Name:** Name of Admin.
- **Emp ID:** It will be a unique value that will act as the primary key.
- **Admin ID:** It will be a unique value that will act as the primary key and will be the same as the employee id in the company.
- **Email ID:** Employee address is also an attribute that helps to get more about the employee.

STORAGE:

Storage is used to store raw material and product that has been produced but not being order.

- **S-No:** Serial number is assigned to every product or raw material to keep their records. It is the Primary key.
- **Bar-Code:** To make the record update process faster. We have added a bar code system that would help to update the status of the product just after a scan using a bar
- code reader.
- **Name:** Name of product or raw materials.

INVENTORY:

Inventory is basically having records of items and their quality.

IT HAS FOLLOWING ATTRIBUTES:

- **Inventory ID:** Inventory Id is the primary key to identify each record.

- **Item ID:** We have already an Items table in our Database. Here Item ID is a foreign key to that table.
- **Quantity:** Quantity describes the number of units available or the amount of product or material available

ITEMS:

Item is actual product we produce in our company.

IT HAS FOLLOWING ATTRIBUTES:

- **Item No**: Item number is numeric data assign to every product. This is unique for every product. That means this Primary key.
- **Bar Code:** Item No is converted into bar code and updated in the barcode field. This would increase the process of tracking and getting actual information.
- **Item-description**: This attribute basically keeps the record of every information about the product.

ORDERS:

Whenever an order is received from the customer. It fetches the item from the item table and tags Order No. to it.

IT HAS THE FOLLOWING ATTRIBUTES:

- **Order No**: This is the primary key to the Order table. It uniquely identifies every record of this table.
- **Barcode**: Every order No is converting to bar code and tag to product and barcode is generated and pasted over product. This will help to track the product.
- **Date Required:** This is attribute store the information of deadline of the product.
- **Date Completed:** When a product is delivered to the client. The date should be updated and payment clearance should note.

SHIPMENTS:

When product is successfully ordered. Its time ship the product.

IT CONTAINS FOLLOWING ATTRIBUTES:

- **Shipment No**: This is the primary key for the Shipment table. It uniquely defines every shipment.
- **Address:** Address is a mandatory field without this field data would not be saved in the database.
- **Shipment Date**: When data is successfully shipped date of that day would update to our database.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

- Traditional empathy maps are split into 4 quadrants (*Says*, *Thinks*, *Does*, and *Feels*), with the user or persona in the middle. Empathy maps provide a glance into who a user is as a whole and are **not** chronological or sequential.
- The Says quadrant contains what the user says out loud in an interview or some other usability study. Ideally, it contains verbatim and direct quotes from research.
- However, pay special attention to what users think, but may not be willing to vocalize. Try to understand why they are reluctant to share — are they unsure, self-conscious, polite, or afraid to tell others something?

  ✓ "This is really annoying."
  ✓ "Am I dumb for not understanding this?"

- The **Does** quadrant encloses the actions the user takes. From the research, what does the user physically do? How does the user go about doing it?

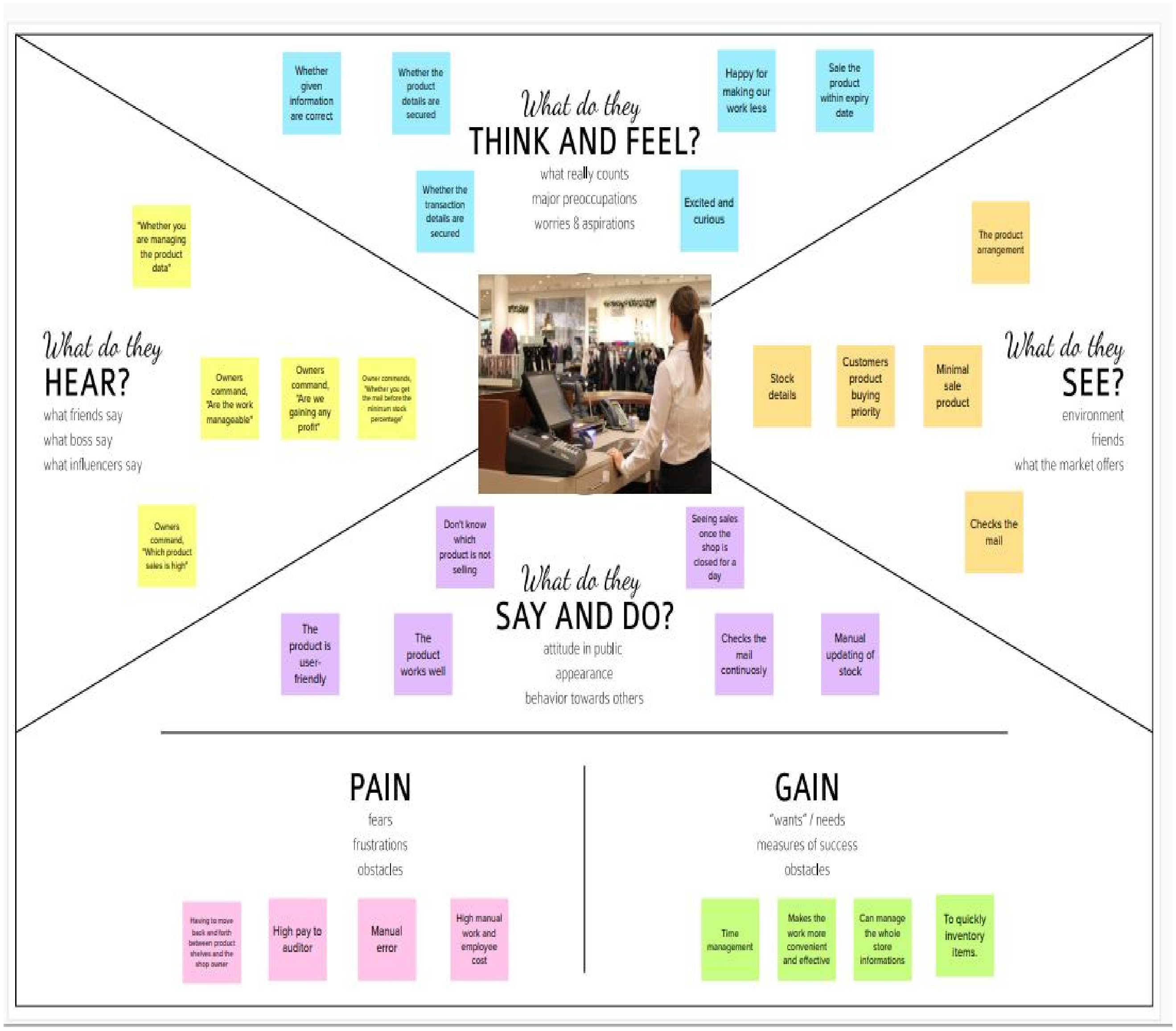

Whether given information are correct
Whether the product details are secured
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Happy for making our work less
Sale the product within expiry date
Whether the transaction details are secured
Excited and curious
"Whether you are managing the product data"
The product arrangement
What do they HEAR?
what friends say
what boss say
what influencers say
Owners command, "Are the work manageable"
Owners command, "Are we gaining any profit"
Owner commands, "Whether you get the mail before the minimum stock percentage"
Stock details
Customers product buying priority
Minimal sale product
What do they SEE?
environment
friends
what the market offers
Owners command, "Which product sales is high"
Don't know which product is not selling
Seeing sales once the shop is closed for a day
Checks the mail
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
The product is user-friendly
The product works well
Checks the mail continuosly
Manual updating of stock
PAIN
fears
frustrations
obstacles
GAIN
"wants" / needs
measures of success
obstacles
Having to move back and forth between product shelves and the shop owner
High pay to auditor
Manual error
High manual work and employee cost
Time management
Makes the work more convenient and effective
Can manage the whole store informations
To quickly inventory items.

Fig 3.1

- The **Does** quadrant encloses the actions the user takes. From the research, what does the user physically do? How does the user go about doing it?

  ✓ Refreshes page several times.
  ✓ Shops around to compare prices.

- The **Feels** quadrant is the user's emotional state, often represented as an adjective plus a short sentence for context. Ask yourself: what worries the user? What does the user get excited about? How does the user feel about the experience?

  ✓ Impatient: pages load too slowly
  ✓ Confused: too many contradictory prices

- ✓ Worried: they are doing something wrong

- Our users are complex humans. It is natural (and extremely beneficial) to see juxtaposition between quadrants. You will also encounter inconsistencies for example, seemingly positive actions but negative quotes or emotions coming from the same user.

- This is when empathy maps become treasure maps that can uncover nuggets of understanding about our user. It is our job as UX professionals to investigate the cause of the conflict and resolve it.

- Some of these quadrants may seem ambiguous or overlapping — for example, it may be difficult to distinguish between *Thinks* and *Feels*. Do not focus too much on being precise: if an item may fit into multiple quadrants, just pick one.

- The 4 quadrants exist only to push our knowledge about users and to ensure we don't leave out any important dimension. (If you don't have anything to put into a certain quadrant, it's a strong signal that you need more user research before proceeding in the design process.)

3.2 IDEATION & BRAINSTROMING

IDEATION

- The first pioneers of inventory management were shop keepers and merchants. Retail inventory management system has come a long way since then. The scales of operations are no more minuscule and today every retailer should have an effective retail inventory management system irrespective of the size of your business. Barcoding and wireless mobility requirements are already in the market for years now.

- It should be noted that all the good systems still leave some room for human decision-making. They do their tasks brilliantly but still leave a final call option to the managers.
- As we can see all around us, most retailers have bought into the multiple advantages offered by such retail inventory management systems. Advantages like high efficiency, lesser space, effective rolling of cash without being tied up, more effective information sharing between retailers, vendors and customers etc are given advantages by implementing these systems.

- There is also a multi-channel move that is happening in the retail sector. In-store channels and mechanisms like mobile-shopping and virtual dressing rooms are now being utilized to offer shoppers a choice to experience shopping on multiple devices.

- This practice is growing tremendously in retail. Retailers are competing hard to provide a seamless shopping experience at any place or device. For these reasons, real-time inventory management across all channels is becoming more important.

- In short, retail inventory management has become a very important aspect of the retail industry itself. Though such automation measures are making things easier (for both online and offline activities), it is recommended to use some crafty measures and techniques that will help retailers manage their inventory much better.

-

STEP-1: TEAM GATHERING, COLLABORATION AND SELECT THE PROBLEM STATEMENT

ACCURACY IN TRACKING

- Retail inventory management systems need accurate calculation. Simply considering how much you can stuff in your facility is not the right way to go. There are factors to be carefully evaluated like how much you have sold in the last month and the month before that. For example, if you have been selling 100 stock or merchandise every month for the last couple of months, it is better to project a number around 100. Of course, there are other factors to consider too (like season).

- Inventory tracking is an area where mistakes are bound to happen (during order-fulfillment for example). This is where inventory management systems like barcode scanner & electronic data interchange (EDI) should be actively used.

- When calculations and tracking are erroneous, the purpose & some advantages of inventory management systems are often defeated.

FAST MOVING & SLOW-MOVING ITEMS

- This is a simple process to understand but a tricky one to satisfy. Items that sell quickly need to be stocked accordingly and its procurement & availability automatically becomes a top priority. All retailers are aware of this process.

- But, there are periods where the slow moving product suddenly goes up in demand and you might be caught unaware with no supplies to back it up (just because your inventory management system suggested keeping that stock low priority).

- Large retailers install costly and complex inventory systems for demand forecasting while small retailers predominantly communicate with their suppliers. This is one challenge that can pop up any time in the retail sector.
- Therefore, discriminating between products and calculating accordingly is an area that retailers always have to keep in mind while managing their inventory.
- There is a common rule that most retailers are aware – that 80% of consumer demand is generated by 20% of all their merchandise.We recommend you to continuously review the stock position of those items and others because both are your priorities (but in order of importance).
- Again, using automated inventory management systems highly reduces your difficulties for these tasks.

STOCK OPTIMIZATION

- Continuing from the above point, stock optimization is an important area that provides efficiency in inventory control, and which gives you some leverage against the uncertainties of fast moving & slow-moving items.
- Some proven techniques that strengthen your stock optimization efforts include updated stocking policies, automatic retail inventory management systems, inventory turnover ratio, inventory audit and budget etc.

TEAM GATHERING

- As the supply chain leader of a large multisite organization, how do you organize and drive an effective, sustainable process to attack the biggest inventory management problems daily across complex teams with multiple ERP systems?
- Have you ever been in a meeting where inventory optimization was brought up as a top priority by leadership without clear direction on how exactly to attack the issue?
- We work with organizations like yours every day and hear how teams are innovating new approaches to these challenges.
- When it comes to driving coordinated action toward inventory optimization goals, one of the best tactics we've heard recently is the deployment of an Inventory Attack Team using advanced analytics, your buyers, and the leaders of your site procurement teams to empower the right people to focus on the most impactful act.

- Let's take a closer look at how you can implement this strategy in your supply chain organization.

STEP-2: BRAINSTORM, IDEA LISTING AND GROUPING

BRAINSTROM

Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge.

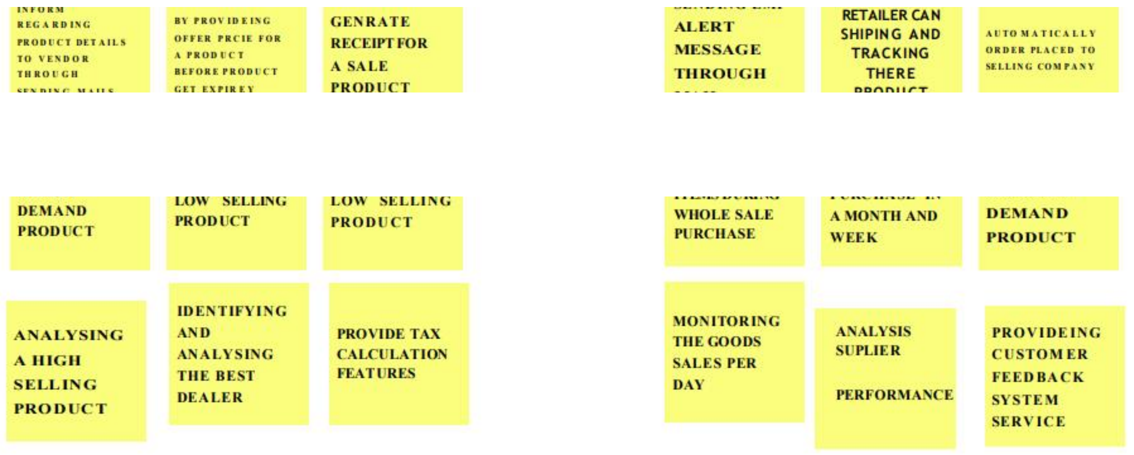

INFORM REGARDING PRODUCT DETAILS TO VENDOR THROUGH
BY PROVIDEING OFFER PRCIE FOR A PRODUCT BEFORE PRODUCT GET EXPIREY
GENRATE RECEIPT FOR A SALE PRODUCT
ALERT MESSAGE THROUGH
RETAILER CAN SHIPING AND TRACKING THERE
AUTOMATICALLY ORDER PLACED TO SELLING COMPANY
DEMAND PRODUCT
LOW SELLING PRODUCT
LOW SELLING PRODUCT
WHOLE SALE PURCHASE
A MONTH AND WEEK
DEMAND PRODUCT
ANALYSING A HIGH SELLING PRODUCT
IDENTIFYING AND ANALYSING THE BEST DEALER
PROVIDE TAX CALCULATION FEATURES
MONITORING THE GOODS SALES PER DAY
ANALYSIS SUPLIER PERFORMANCE
PROVIDEING CUSTOMER FEEDBACK SYSTEM SERVICE

Fig 3.1

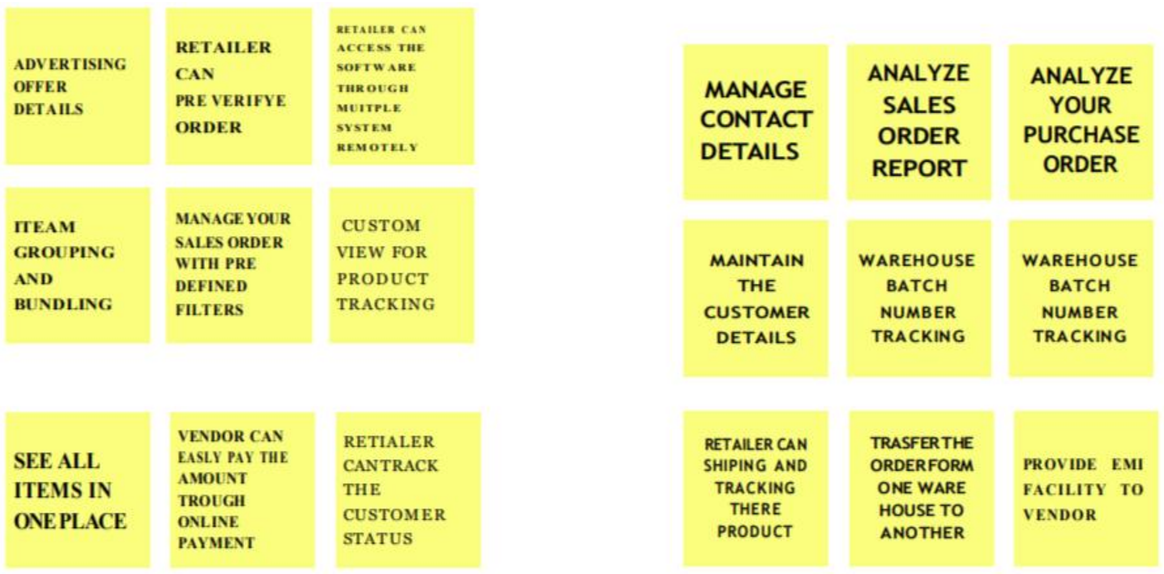

ADVERTISING OFFER DETAILS
RETAILER CAN PRE VERIFYE ORDER
RETAILER CAN ACCESS THE SOFTWARE THROUGH MUITPLE SYSTEM REMOTELY
MANAGE CONTACT DETAILS
ANALYZE SALES ORDER REPORT
ANALYZE YOUR PURCHASE ORDER
ITEAM GROUPING AND BUNDLING
MANAGE YOUR SALES ORDER WITH PRE DEFINED FILTERS
CUSTOM VIEW FOR PRODUCT TRACKING
MAINTAIN THE CUSTOMER DETAILS
WAREHOUSE BATCH NUMBER TRACKING
WAREHOUSE BATCH NUMBER TRACKING
SEE ALL ITEMS IN ONE PLACE
VENDOR CAN EASLY PAY THE AMOUNT TROUGH ONLINE PAYMENT
RETIALER CANTRACK THE CUSTOMER STATUS
RETAILER CAN SHIPING AND TRACKING THERE PRODUCT
TRASFER THE ORDERFORM ONE WARE HOUSE TO ANOTHER
PROVIDE EMI FACILITY TO VENDOR

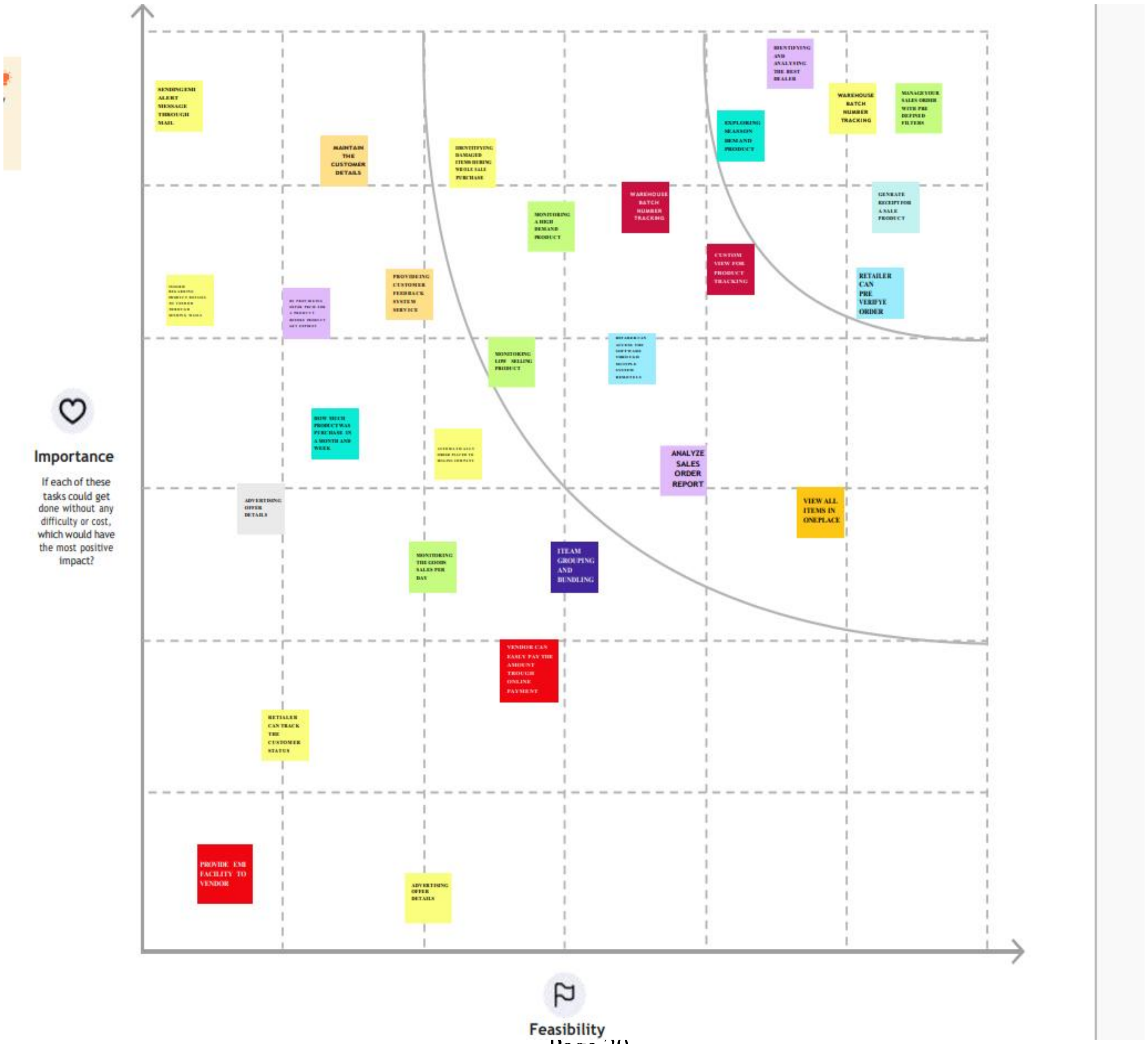
Fig 3.2

STEP-3: IDEA PRIORITIZATION

PRIORITIZE YOUR INVENTORY.

- Categorizing your inventory into priority groups can help you understand which items you need to order more of and more frequently, and which are important to your business but may cost more and move more slowly. Experts typically suggest segregating your inventory into A, B and C groups.

- Items in the A group are higher-ticket items that you need fewer of. Items in the C category are lower-cost items that turn over quickly. The B group is what's in between: items that are moderately priced and move out the door more slowly than C items but more quickly than A items.



SENDING EMI ALERT MESSAGE THROUGH MAIL
MAINTAIN THE CUSTOMER DETAILS
IDENTIFYING DAMAGED ITEMS DURING WHOLE SALE PURCHASE
IDENTIFYING AND ANALYSING THE BEST DEALER
EXPLORING SEASSON DEMAND PRODUCT
WAREHOUSE BATCH NUMBER TRACKING
MANAGE YOUR SALES ORDER WITH PRE DEFINED FILTERS
MONITORING A HIGH DEMAND PRODUCT
WAREHOUSE BATCH NUMBER TRACKING
GENRATE RECEIPT FOR A SALE PRODUCT
CUSTOM VIEW FOR PRODUCT TRACKING
RETAILER CAN PRE VERIFYE ORDER
PROVIDEING CUSTOMER FEEDBACK SYSTEM SERVICE
MONITORING LOW SELLING PRODUCT
HOW MUCH PRODUCT WAS PURCHASE IN A MONTH AND WEEK
ANALYZE SALES ORDER REPORT
ADVERTISING OFFER DETAILS
VIEW ALL ITEMS IN ONEPLACE
MONITORING THE GOODS SALES PER DAY
ITEAM GROUPING AND BUNDLING
VENDOR CAN EASILY PAY THE AMOUNT TROUGH ONLINE PAYMENT
RETAILER CAN TRACK THE CUSTOMER STATUS
PROVIDE EMI FACILITY TO VENDOR
ADVERTISING OFFER DETAILS
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility

Fig 3.3

TRACK ALL PRODUCT INFORMATION.

- Make sure to keep records of the product information for items in your inventory. This information should include SKUs, barcode data, suppliers, countries of origin and lot numbers.

- You might also consider tracking the cost of each item over time so you're aware of factors that may change the cost, like scarcity and seasonality.

AUDIT YOUR INVENTORY.

- Some businesses do a comprehensive count once a year. Others do monthly, weekly, or even daily spot checks of their hottest items. Many do all the above.

- Regardless of how often you do it, make it a point to physically count your inventory regularly to ensure it matches up with what you think you have.

ANALYZE SUPPLIER PERFORMANCE.

- An unreliable supplier can cause problems for your inventory. If you have a supplier that is habitually late with deliveries or frequently shorts an order, it's time to act.

- Discuss the issues with your supplier and find out what the problem is. Be prepared to switch partners, or deal with uncertain stock levels and the possibility of running out of inventory as a result.

PRACTICE THE 80/20 INVENTORY RULE.

- As a rule, 80% of your profits come from 20% of your stock. Prioritize inventory management of this 20% of items. You should understand the complete sales lifecycle of these items, including how many you sell in a week or a month, and closely monitor them.

- These are the items that make you the most money; don't fall short in managing them.

BE CONSISTENT IN HOW YOU RECEIVE STOCK.

- It may seem like common sense to make sure incoming inventory is processed, but do you have a standard process that everyone follows, or does each employee receiving and processing incoming stock do it differently? Small discrepancies in how new stock is taken in could leave you scratching your head at the end of the month or year, wondering why your numbers don't align with your purchase orders.

- Make sure all staff that receives stock does it the same way, and that all boxes are verified, received, and unpacked together, accurately counted, and checked for accuracy.

TRACK SALES.

- Again, this seems like a no-brainer, but it goes beyond simply adding up sales at the end of the day. You should understand, daily, what items you sold and how many, and update your inventory totals. But beyond that, you'll need to analyze this data.

- Do you know when certain items sell faster or drop off? Is it seasonal? Is there a specific day of the week when you sell certain items? Do some items almost always sell together? Understanding not just your sales totals but the broader picture of how items sell is important to keeping your inventory under control.

ORDER RESTOCKS YOURSELF.

- Some vendors offer to do inventory reorders for you. On the surface, this seems like a good thing – you save on staff and time by letting someone else manage the process for at least a few of your items.

- But remember that your vendors don't have the same priorities you do. They are looking to move their items, while you're looking to stock the items that are most profitable for your business. Take the time to check inventory and order restocks of all your items yourself.

INVEST IN INVENTORY MANAGEMENT TECHNOLOGY.

- If you're a small enough business, managing the first eight things on this list manually, with spreadsheets and notebooks, is doable. But as your business grows, you'll spend more time on inventory than you do on your business, or risk your stock getting out of control.

- Good inventory management software makes all these tasks easier. Before you choose a software solution, make sure you understand what you need, that it provides the analytics important to your business and that it's easy to use.

USE TECHNOLOGY THAT INTEGRATES WELL.

- Inventory management software isn't the only technology that can help you manage stock. Things like mobile scanners and POS systems can help you stay on track. When investing in technology, prioritize systems that work together.

- Having a system that can't communicate with your inventory management software isn't the end of the world, but it might cost you extra time to transfer the data from one system to another, making it easy to end up with inaccurate inventory counts.

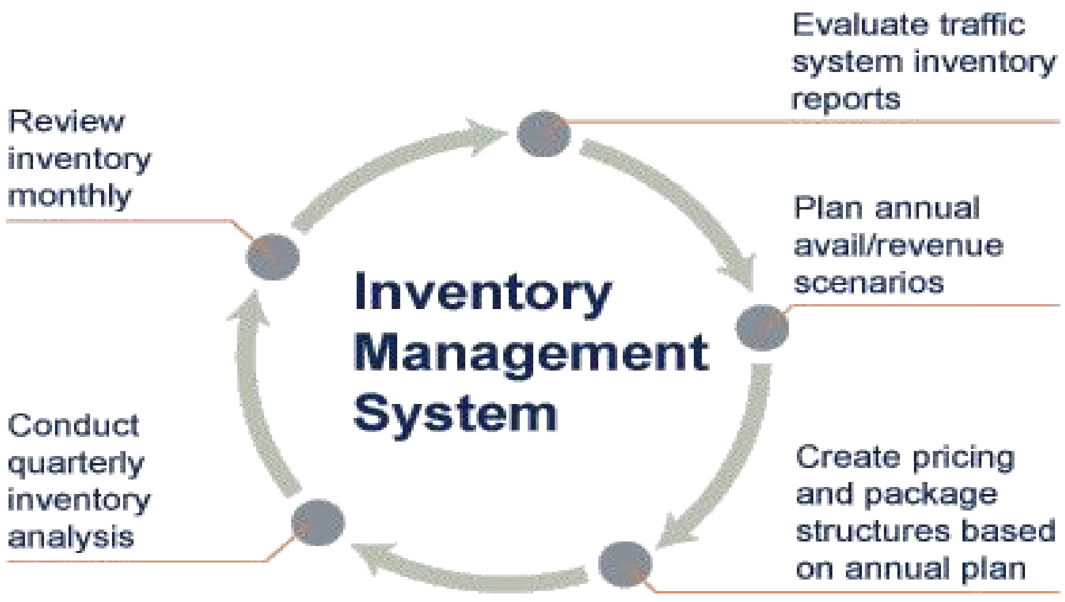

Inventory Management System
Evaluate traffic system inventory reports
Plan annual avail/revenue scenarios
Create pricing and package structures based on annual plan
Conduct quarterly inventory analysis
Review inventory monthly

Fig 3.4

GRAPHICAL REPRESENTATION OF INVENTORY SYSTEM

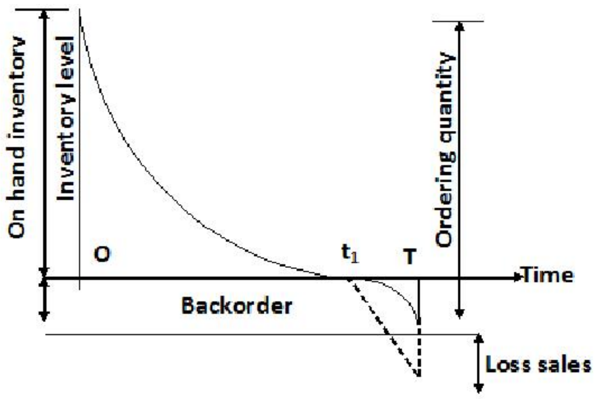


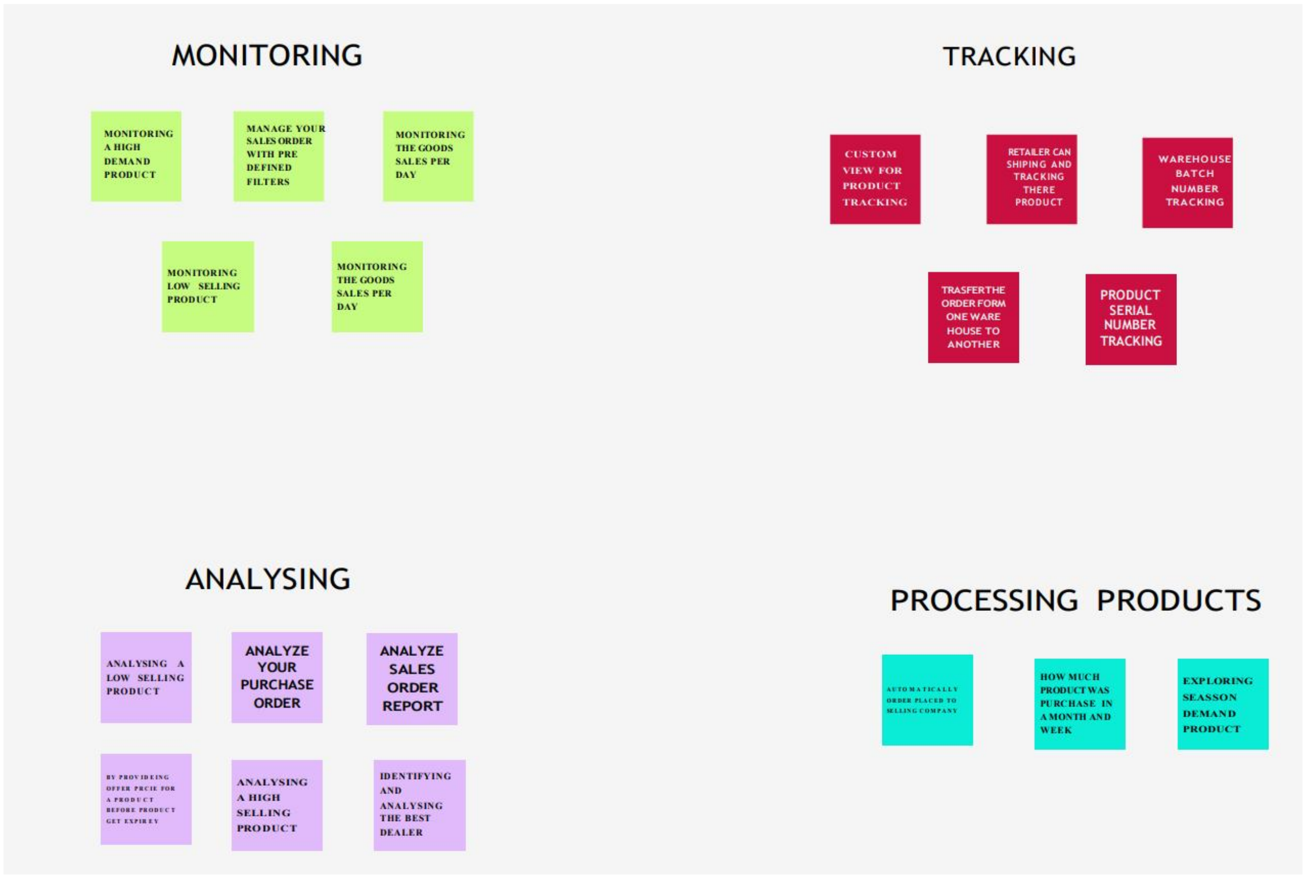
Fig 3.5

STEP-4: GROUP PRIORITIZATION

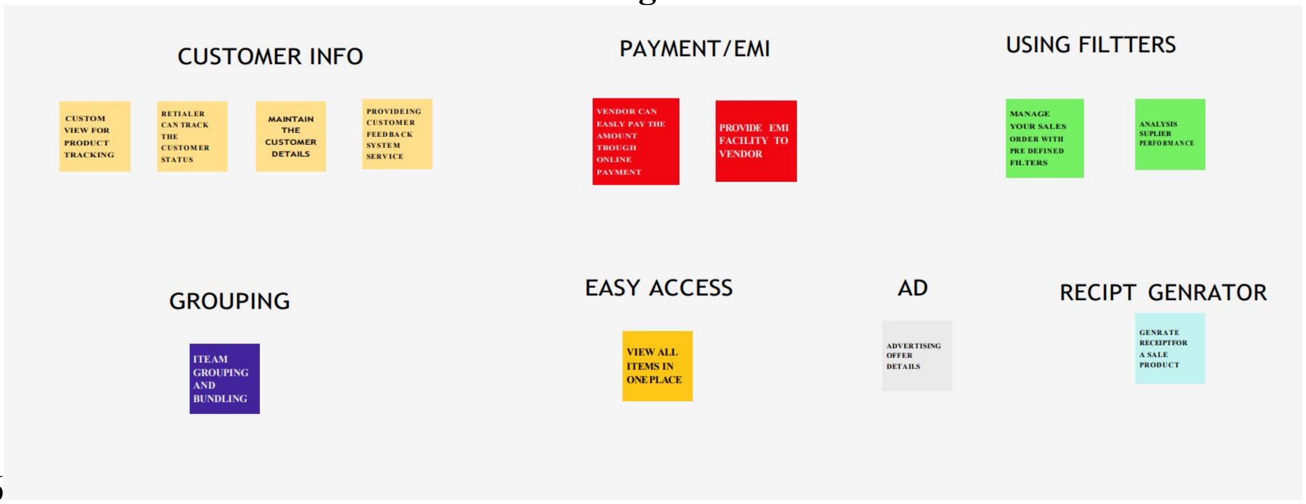
GROUP IDEA

- Before building an inventory management plan, you'll need to have a solid understanding of each step in the inventory management process. This is crucial to minimizing error and choosing the most effective inventory management software for your business.

  ✓ Choose an appropriate fulfillment option.
  ✓ Take forecasting seriously.
  ✓ Set reorder points for each product.
  ✓ Use EOQ for optimal order quantities.
  ✓ Give each variant a dedicated warehouse bin.
  ✓ Sell older inventory first.
  ✓ Prioritize with ABC analysis.
  ✓ Always track your metrics.
  ✓ Verify accuracy with regular counts.
  ✓ Automate as much as possible.


MONITORING
MONITORING A HIGH DEMAND PRODUCT
MANAGE YOUR SALES ORDER WITH PRE DEFINED FILTERS
MONITORING THE GOODS SALES PER DAY
MONITORING LOW SELLING PRODUCT
MONITORING THE GOODS SALES PER DAY
TRACKING
CUSTOM VIEW FOR PRODUCT TRACKING
RETAILER CAN SHIPING AND TRACKING THERE PRODUCT
WAREHOUSE BATCH NUMBER TRACKING
TRASFER THE ORDER FORM ONE WARE HOUSE TO ANOTHER
PRODUCT SERIAL NUMBER TRACKING
ANALYSING
ANALYSING A LOW SELLING PRODUCT
ANALYZE YOUR PURCHASE ORDER
ANALYZE SALES ORDER REPORT
BY PROVIDEING OFFER PRCIE FOR A PRODUCT BEFORE PRODUCT GET EXPIREY
ANALYSING A HIGH SELLING PRODUCT
IDENTIFYING AND ANALYSING THE BEST DEALER
PROCESSING PRODUCTS
AUTOMATICALLY ORDER PLACED TO SELLING COMPANY
HOW MUCH PRODUCT WAS PURCHASE IN A MONTH AND WEEK
EXPLORING SEASSON DEMAND PRODUCT

Fig


CUSTOMER INFO
CUSTOM VIEW FOR PRODUCT TRACKING
RETIALER CAN TRACK THE CUSTOMER STATUS
MAINTAIN THE CUSTOMER DETAILS
PROVIDEING CUSTOMER FEEDBACK SYSTEM SERVICE
PAYMENT/EMI
VENDOR CAN EASLY PAY THE AMOUNT TROUGH ONLINE PAYMENT
PROVIDE EMI FACILITY TO VENDOR
USING FILTTERS
MANAGE YOUR SALES ORDER WITH PRE DEFINED FILTERS
ANALYSIS SUPLIER PERFORMANCE
GROUPING
ITEAM GROUPING AND BUNDLING
EASY ACCESS
VIEW ALL ITEMS IN ONEPLACE
AD
ADVERTISING OFFER DETAILS
RECIPT GENRATOR
GENRATE RECEIPTFOR A SALE PRODUCT

3.6

Fig 3.7

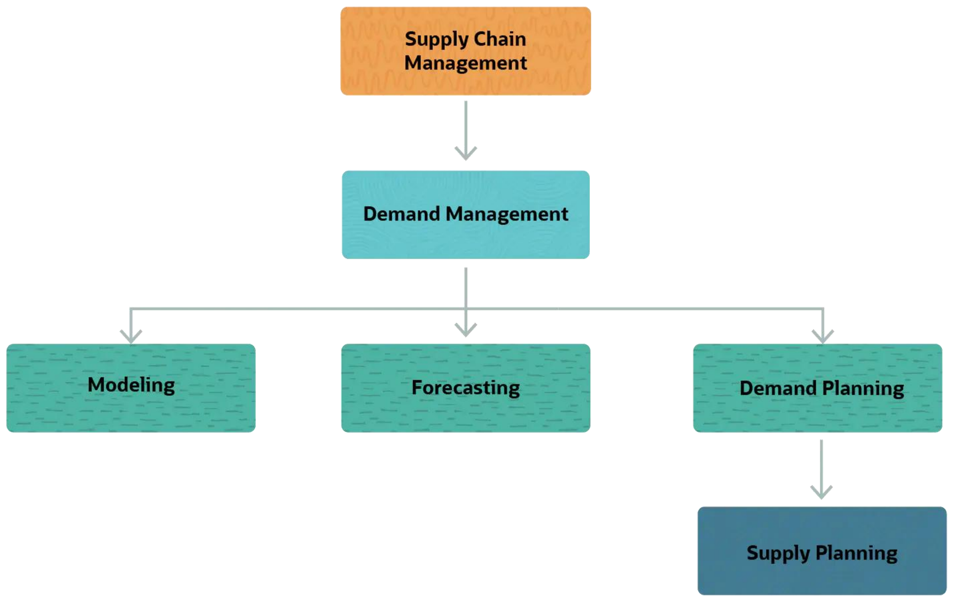
OBJECTIVES OF DEMAND MANAGEMENT

- Successful demand management teams today are customer-centric it's all about the ability to predict and fulfill demand with the right products and services. Specific areas of focus include improved customer service, more accurate forecasting, and lower costs.

- Specific objectives of customer-centric demand management include:

  - ✓ **Improved customer service:** Understanding client needs and behaviors increases customer satisfaction and improves service.
  - ✓ **Forecasting with greater accuracy:** Predictive analytics efforts optimize decisions by business leaders and improve supply chain management.
  - ✓ **Reduced costs:** Improved forecasting optimizes inventory investments and can minimize safety stock levels.
  - ✓ **Enhance existing products and excel at new product introductions:** Create a line of customer-appropriate new products and refine them based on feedback.
  - ✓ **More efficient planning:** Strike the right balance of demand to supply and minimize surpluses with reliable data.

The Demand Management Process and Components



COST EFFICIENCY

- Cost-effective inventory management is an important source for creating competitive advantages. It enables companies to deliver products in the right amounts, at the right time with the right quality.
- Too much stock results in unnecessary costs. Equally, too little in stock can lead to lost sales. In this article, we explore the 5 fundamentals to attain the optimal level of inventory at the lowest possible cost.

THE FUNDAMENTALS OF COST-EFFECTIVE INVENTORY MANAGEMENT

The basis of 'good' inventory management is all about ensuring that the right goods are available in the right quantity, with the right inventory turnover. However, with a conscious focus on the following 5 areas, you can build more cost-effective inventory management processes:

1) Demand Forecasting
2) Master Data
3) Software Support
4) Human Competencies
5) Relationship Management.

DEMAND FORECASTING

- Reliable forecasts are important to create a trustworthy basis for all decisions around assortments, purchasing volume, and safety stocks. Historical data is often used together with knowledge about inventory turnovers, current order levels, and expectations for future sales.

- Here it is important to ask whether the existing forecasting processes are up to date. More importantly, is the quality of the forecast satisfactory. Often with modest efforts, opportunities to improve can quickly be identified.

MASTER DATA

- A companies' business processes are only as good as the data they are built on. Poor data will lead to poor decision making. And poor decisions often result in costly excess, crippling stockouts & angry customers.

- The question is: How often is your inventory master data re-evaluated? Are you confident that you have the right lead times in the system? Are your suppliers' minimum order quantities up to date? Are your service levels rules & safety stock levels appropriate?

SOFTWARE SUPPORT

- Cost-effective inventory management typically requires a lot of data & many algorithms. However, many businesses struggle with 'spreadsheet fascists' who strive to deliver decision-relevant information through very manual processes. As a consequence, many businesses simply lack the transparency to manage inventory effectively.

HUMAN COMPETENCIES

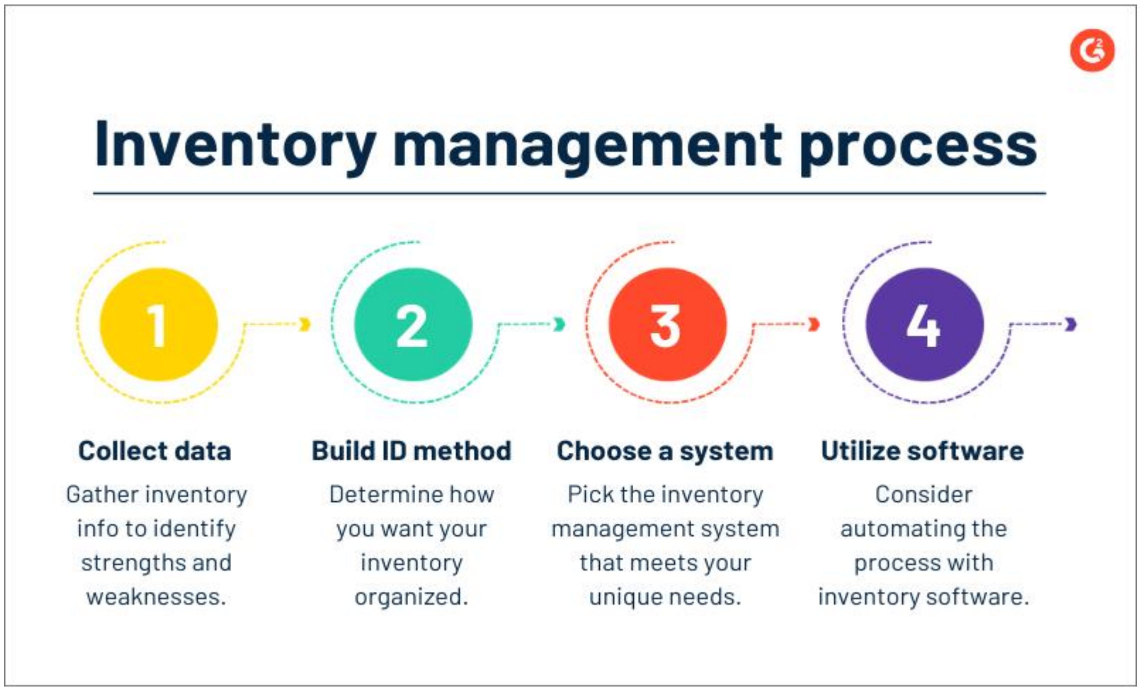
- Humans play a pivotal role in cost-effective inventory management. It is important to create a common overview and understanding of the inventory workflows. Therefore, everyone involved must receive sufficient training.

- More importantly, it's essential that supply chain roles and responsibilities are clearly defined. For example: Who 'owns' the Master data? Who is responsible for ensuring the forecasts are accurate? Who ultimately determines the service level?
- Without the right knowledge and accountability, the entire inventory management process could be hindered by errors & inefficiencies.

RELATIONSHIP MANAGEMENT

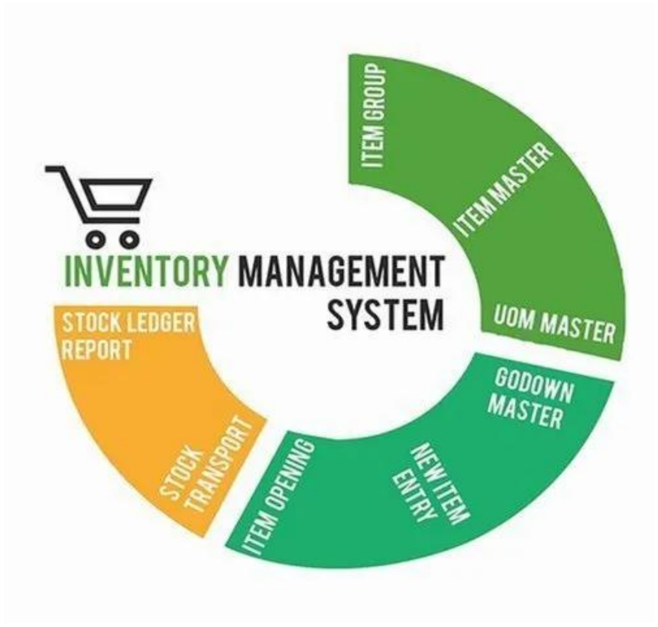
The last area that contributes to cost-effective inventory management comes down to how well supplier relations are managed. No chain is stronger than the weakest link. Therefore, communication and information sharing is crucial to make effective decisions. As well as to take corrective actions where supply fails to align with demand.

MANAGEMENT


Inventory management process
1
2
3
4
Collect data
Gather inventory info to identify strengths and weaknesses.
Build ID method
Determine how you want your inventory organized.
Choose a system
Pick the inventory management system that meets your unique needs.
Utilize software
Consider automating the process with inventory software.

3.3 PROPOSED SOLUTOIN

- The technical solution to be provided by the Implementation agency in response to the requirements and the objectives of the Project. Sample 1Sample 2. Proposed Solution means the Proposed System with modifications that meet the Agency's requirements as set forth in this RFP.



- Inventory management helps companies identify which and how much stock to order at what time. It tracks inventory from purchase to the sale of goods. The practice identifies and responds to trends to ensure there's always enough stock to fulfill customer orders and proper warning of a shortage.

S.NO	PARAMETERS	DESCRIPTION
1.	Problem Statement (Problem to be solved)	The problem to be solved is to make an application for retailers to track their inventory stocks to manage the purchases, sales, stocks, etc

2.	Idea / Solution description	The idea to solve this is by developing an application to track and manage stocks related to their own products. The retailers create their accounts by proving their details and entering the stocks/inventory of their products. Once done, they can login to the application and view their stocks, sales, update their stocks when restocking, etc. They can see which stocks are fast moving and when in case of running out, they get notification, and they can restock their fastmoving stocks
3.	Novelty / Uniqueness	As we have data of the sales per stocks, we can include a prediction of stocks to guess which will be the most purchased stocks so that the retailers can restock up on that prior. The data can be obtained by regression and previous sales data within our application. We can also make maintenance and development easier by containerizing via Docker application.
4.	Social Impact / Customer Satisfaction	By using our application, we can see which stocks are being sold and which are not much as expected, so by using that data we can purchase and restock only the required stocks and thus reducing excess stocks in the inventory which might be a wastage of products. Since we will know which products are needed in bulk, we can request vendors and suppliers the required number of stocks and negotiate better deals with them beforehand.
5.	Business Model (Revenue Model)	Retailers can order the fast-moving products and the right number of stocks from suppliers and vendors by analyzing the predicted products which has higher chance of being purchased in large amounts, and thus eliminating unnecessary redundant products which might be excess when not ordered in the right amount.
6.	Scalability of the Solution	Scalable cloud architecture is made possible through virtualization... Unlike physical machines whose resources and performances are set by their physical hardware, processors and memory. Virtual Machines that we use in IBM Cloud are highly flexible and scalable. Kubernetes allows the users to scale the containers based on the application requirements which may vary over time. It’s easy to change the

		number via command lines.

3.4 PROBLEM SOLUTION FIT

1.CUSTOMER SEGMENTS

- Customer segmentation is an important marketing tool. Effective customer segmentation helps the enterprises increase profits and improve customer service level. On the other hand, due to possible detrimental consequences, supply disruptions have been receiving more and more attention.

- This paper aims to investigate the effect of customer segmentation on a single-product inventory system in the presence of supply disruptions. The concerned inventory system involves an unreliable supplier, a retailer, and customers.

- The retailer adopts a continuous-review (s, S) inventory policy. Partial backordering is considered when stockouts occur. This inventory system is simulated. Based on different customer backorder proportions, the effect of customer segmentation on the inventory system is studied under different scenarios about supply disruption severity.

-
- The experimental results show that supply disruption duration is an important factor in influencing the effect of customer segmentation on the inventory system. Some managerial insights are also derived from the results.

2.LACK OF INVENTORY VISIBILITY

- If you're unable to locate or identify stocks in your inventory, shipping products on time becomes very difficult, and this can dent your business reputation.

- Inventory that is incomplete, difficult to find, or erroneous, is sure to hamper your bottom line.

- In fact, the most common reason for delayed, wrong, or partial shipments is the difficulty of locating or identifying inventory in the warehouse.

- Receiving in finding the correct stock is critical for ensuring warehouse efficiency, as well as good experiences for the customer.

- Solution: Real-time inventory management system

- When you implement a real-time inventory management system like ERP, you will have all the accurate details regarding location data, and stock availability.

- This will help in the easy location of stocks, which ultimately translates to better order fulfilment and customer satisfaction.

3.INEFFICIENT INVENTORY MANAGEMENT PROCESS OR SOFTWARE

- This is probably the most common, and the biggest **inventory challenges**.

- Many businesses still try to manage their inventory with manual procedures, or outdated legacy software – which may stunt your business growth.

- It may not seem much of a bother to use manual, labor-intensive, or low-tech systems when you're a small, one-warehouse business – but that will change when you expand.

- When sales volumes balloon, you need to expand inventory and add warehouses.

- Old and inefficient inventory management practices will be tough to scale, and prove to be a handicap, and not give you the results you need.

- Manual inventory tracking procedures involve either paperwork or tracking procedures across multiple spreadsheets and software, which can lead to data redundancy, incomplete data, and a lot of time spent on it; it also provides less security.

4.TRACKING OBSOLETE MATERIAL

- In almost every business, you are likely to face this problem at some time or the other.
- There will be some products or materials that remain unsold or unused, and they may become obsolete, or past their expiry date.

- These materials or products tend to accumulate over time as they are mostly ignored by inventory managers.

- When that product or material is needed sometime in the future, unfortunately, the unsold stock stays forgotten, and new stock is purchased; the older one may remain in the warehouse for so long that it gets damaged completely.

- This increases expenses, and material wastage.

- Solution: Efficient stock control system

5.IDENTIFYING INCORRECTLY LOCATED MATERIALS

- Employees, When there is no proper system to track products, materials, or equipment in the store, it can be cumbersome and time-consuming to find them when you have sales orders.
- After all, a warehouse may typically store thousands of products.
- This can delay sales and make customers unhappy.
- Solution: Product finder
- All products that you have should be tagged with RFID, barcodes, or QR codes etched with a laser.
- This will help your employees identify the products that are needed.
- They only need to be equipped with a scanner. Once the scanner finds the required product, a lamp glows, indicating a match.
- This helps your pickers to quickly find the product and send it to the sales agents, saving them time, speeding up the sales cycle, and making customers happy.

6.KEEPING UP WITH OVERSTOCKS

- When you purchase new materials with a few unsold products lying in your warehouse, it can affect your profitability.
- This situation mostly arises due to the inefficiencies of manual processes, which causes poor control of stock.
- Storing too much stock is as bad as storing too little, as overstocking hampers your cash flow and creates problems related to inventory, like storage, or loss.
- Solution: Stock audit process
- When you implement a stock audit process, inventory managers will be able to audit stocks regularly so that unused stocks are quickly identified.
- This boosts the efficiency of inventory greatly, enabling your company to cut costs,

eliminate delays, and enhance profitability.

7.LACK OF CENTRALIZED INVENTORY HUB

- Stocktaking becomes very challenging when you have inventories in multiple locations.
- Discrete stock data from various locations makes shipping complex, resulting in delays.

- It’s one of the biggest and continual challenges faced by most businesses today.

- Solution: Central inventory system

- You can significantly reduce expenses and save a great deal of time by simply creating a centralized inventory hub for your inventory-related data, including stock-taking.

- This gives you comprehensive visibility and control of inventory and data in one single location, making stock management simple.

- It also becomes much easier to track the inventory that enters and leaves your business premises.

8.CHANGING DEMAND

- Consumer demand is in a constant state of flux; this makes storing inventory complicated.
- How much to store? Too much, and you could end up with dead stock; too little, and you won’t be able to fulfill customer demands.

- Solution: Technology to Plan Inventory

- What you need is robust inventory forecasting technology that takes into consideration all these factors, and helps you plan your inventory more efficiently.

- Tranquil ERP’s inventory management module has the forecast feature that helps create and implement the optimal inventory plan.

- This can help you to keep up with fluctuating customer demand.

9.SUPPLY CHAIN COMPLEXITY

- International supply chains are dynamic and can create roadblocks in the management and planning of your inventory.

- Manufacturers and distributors are impacted by unforeseen economic booms and slumps which impact raw material prices and availability.

- They also decide when, how, and where to ship the inventory – and this means you have lead times that you cannot predict, necessitating you to be much more flexible.

- Solution: Robust Inventory Management application

- With the right inventory management application implemented in your business, you can predict lead times as close to accuracy as possible, and be better prepared to handle supply chain complexities.

10.MANAGING WAREHOUSE SPACE AND EFFICIENCY

- One of the most challenging tasks for any business is the efficient management of space.
-
- Warehouses need to be planned and designed with the help of inventory management platforms so that you can control when new stock is delivered, and help you make the best use of available space.

- If you deal in fragile or perishable products, you need to arrange specialized care and storage – for example, cold storage.

- You have to implement specific strategies for expensive inventory, to prevent theft and damage.

- Warehouse inventory control is labor-intensive, and necessitates multiple steps like receiving the stocks, putting them away, picking inventory, packing, and finally, shipping.
- It is critical that all of these tasks are executed as efficiently as possible.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

- **Functional requirements** are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions. For example:

- The system sends an approval request after the user enters personal information.

- A search feature allows a user to hunt among various invoices if they want to credit an issued invoice.

- The system sends a confirmation email when a new user account is created.

TYPES OF FUNCTIONAL REQUIREMENTS AND THEIR SPECIFICATIONS

Functional requirements can be classified according to different criteria. For example, we can group them on the basis of the *functions* a given feature must perform in the end product. Of course, they would differ depending on the product being developed, but for the sake of an example, the types of functional requirements might be

AUTHENTICATION

❖ Authentication is the process of determining whether someone or something is, in fact, who or what it says it is.

❖ Authentication technology provides access control for systems by checking to see if a user's credentials match the credentials in a database of authorized users or in a data authentication server.

AUTHORIZATION LEVELS

❖ Authorization is the rights and permissions granted to a user or application that enables access to a network or computing resource.

- Once a user has been properly identified and authenticated, authorization levels determine the extent of system rights that the user has access to.

COMPLIANCE TO LAWS OR REGULATIONS

- Regulatory compliance is an organization's adherence to laws, regulations, guidelines, and specifications relevant to its business processes.
- Violations of regulatory compliance often result in legal punishment, including federal fines.

EXTERNAL INTERFACES

- The external interface is the interface that connects to the Internet or a Wide Area Network (WAN). The external interface must have an IP address to operate correctly.

- You can assign a static or dynamic IP address to the external interface.

TRANSACTIONS PROCESSING

- Transaction processing is a style of computing, typically performed by large server computers, that supports interactive applications
-
- . In transaction processing, work is divided into individual, indivisible operations, called transactions.

REPORTING

- Reporting is **the presenting of news in newspapers, on radio, and on television**. ... honest and impartial political reporting. Synonyms: journalism, writing, presenting, newscasting More Synonyms of reporting.

BUSINESS RULES

- Business rules are directives that define an organization's business activities. They are important because they clarify an organization's objectives and detail how processes will be performed.
- Business rules can be informal, written, or automated.

- The System calculates and predicts the amount of usage for specific set days that are pre-set by the user(admin) , it also alerts the user of an impending action to order ingredients before the specific day set by the user.
- Therefore, the user never has to worry about manually calculating the estimated usage of the ingredients as the System does it for the user.

- The simple interface of the System has functions like adding a recipe, removing or updating the recipe. It also extends to functions such as adding a vendor for an ingredient, removing the vendor, checking threshold levels, processing orders, altering processed orders etc.

4.2 NON-FUNCTIONAL REQUIREMENT

- The non-functional requirement says about "what a system should be" rather than "what a system should do" (functional requirement). They are mostly derived from functional requirements based on input from the customer and other stakeholders. Non-functional requirement implementation details are documented in the System Architecture document.

- Non-functional requirements explain the quality aspects of the system to be constructed viz. performance, portability, usability, etc. Non-functional requirements, unlike functional requirements, are implemented incrementally in any system.

- URPS (Usability, Reliability, Performance, and Supportability) from FURPS (Functionality, Usability, Reliability, Performance, and Supportability) quality attributes that are widely used in the IT industry to measure the quality of a software developer, are all covered in non-functional requirements. Besides, there are other quality attributes also (details in the next section).

USABILITY

- The system must be easy to use by both managers and chefs such that they do not need to read an extensive number of manuals.

- The system must be intuitive and simple in the way it displays all relevant data and relationships.

- The menus of the system must be easily navigable by the users with buttons that are easy to understand.

RELIABILITY

- The System must give accurate inventory status to the user continuously. Any inaccuracies are taken care by the regular confirming of the actual levels with the levels displayed in the system.

- The system should provide the user updates on completion of requested processes and if the requested processes fail, it should provide the user the reason for the failure.

- The system should not update the data in any database for any failed processes.

PERFORMANCE

- The system must not lag, because the workers using it don't have down-time to wait for it to complete an action.
- The calculations performed by the system must comply according to the norms set by the user and should not vary unless explicitly changed by the user.

SUPPORTABILITY

- The software is designed such that it works even on systems having the minimum configuration.
- The system is adaptable even if additional plugins or modules are added at a later point.
- The packaging must come with a manual that details the use of the system, and also the instructions on how to use the program. This manual may be included either in a booklet that comes with the software, or on the disc that the software itself is on.

IMPLEMENTATION

- The System User Interface is built on Microsoft Visual Studio 2022.The Programing is done in Microsoft Visual Studio 2022.
- The Database is implemented on the Microsoft Access 2022.
- The connection between the Database and the System is achieved using ODBC connection available at hand in Visual Studio 2010.

INTERFACING

- The system must offer an easy and simple way of viewing the current inventory.
- The system must be able to display the relationships between vendors, ingredients, and recipes in an intuitive manner.

LEGAL

- The software must be licensed on an individual basis for smaller companies, as well as through a multi-license deal for larger corporations.
- The client should agree to EULA before using our software.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

0 LEVEL DFD FOR INVENTORY MANAGEMENT SYSTEM

- The parameters of the inventory management system are specified in a context diagram (level 0 data-flow diagram). It illustrates how information moves between the system and its external entities.

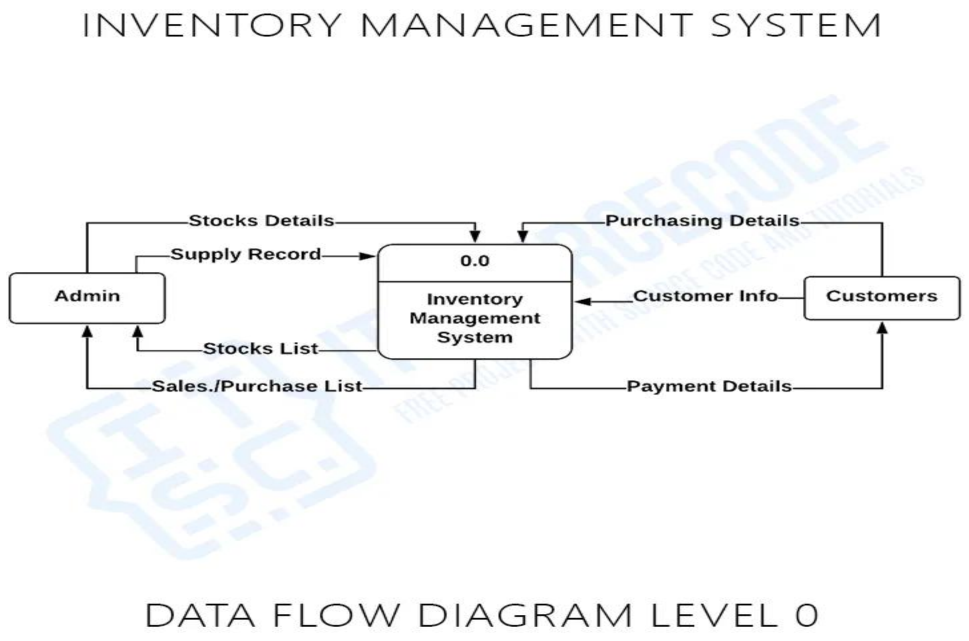


Fig 5.1

- The diagram’s arrows show the direction in which the data input flowed. The major function is labeled as the “inventory management system,” and its name sparks the discussion under it.

- The external entities that cause the system to perform a certain function are as follows:
  - ✓ Customers
  - ✓ System Admin

- As a result, the following illustrations begin at the system’s DFD level 0.

LEVEL 1 DFD FOR INVENTORY MANAGEMENT SYSTEM

- The **first level DFD of the inventory management system** describes each of the system’s primary sub-processes. This level represents the context diagram’s “**extended viewpoint**.”

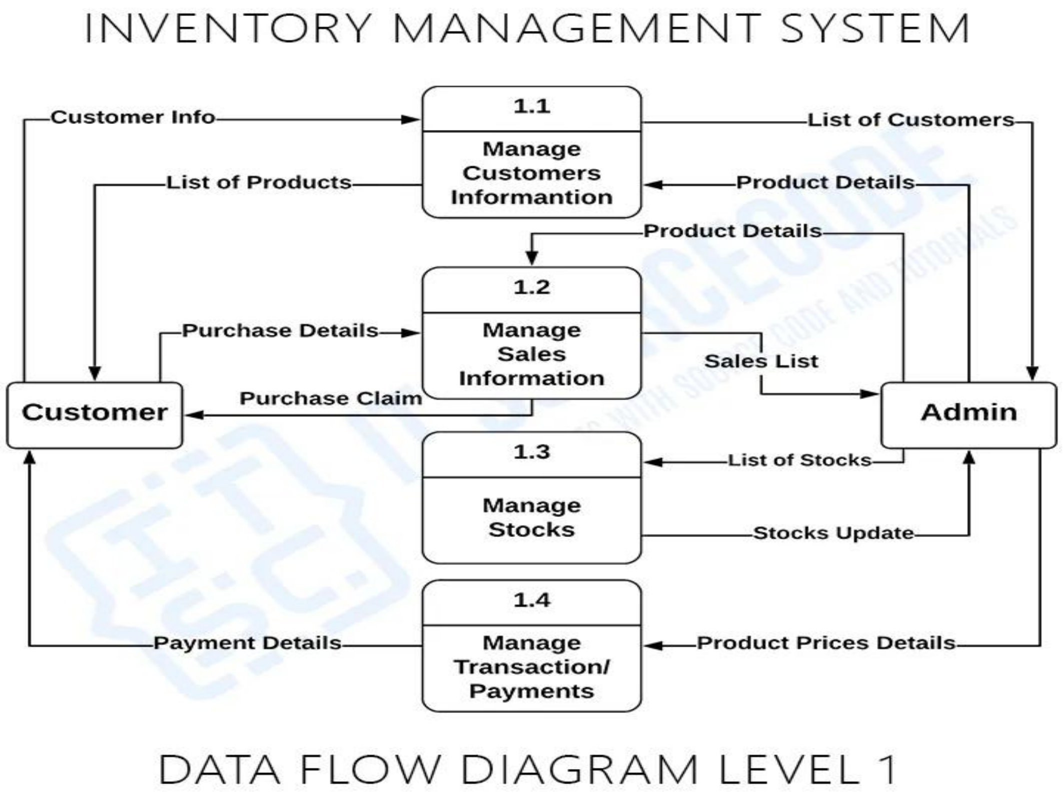


Fig 5.2

- The given example reveals the sub-processes under inventory management. Each of them complements in showing how exactly the inventory management system works. The sub-processes ar
- The process of managing customer information includes the gathering of the customer’s basic data. This data is used for transactions, purchasing, and inventory

purposes.

- The system also **manages the transactions and payments** of each customer to complete the overall function. Then the transactions were saved in the database for inventory purposes.

- Be aware that the basis for these concepts comes from the basic activities associated with managing practical inventories. The suggested idea can be used as is or altered to serve your intended purpose.

- You can stop working at this point because you already understand how the project will function overall. The next level, though, might pique your interest due to what it is and how it functions.

DFD LEVEL 2 FOR INVENTORY MANAGEMENT SYSTEM

- Among the preceding levels, **DFD level 2** has the highest concept abstraction. The reason for this is that this level describes the processes (if any) that fall under the level 1 sub-process.

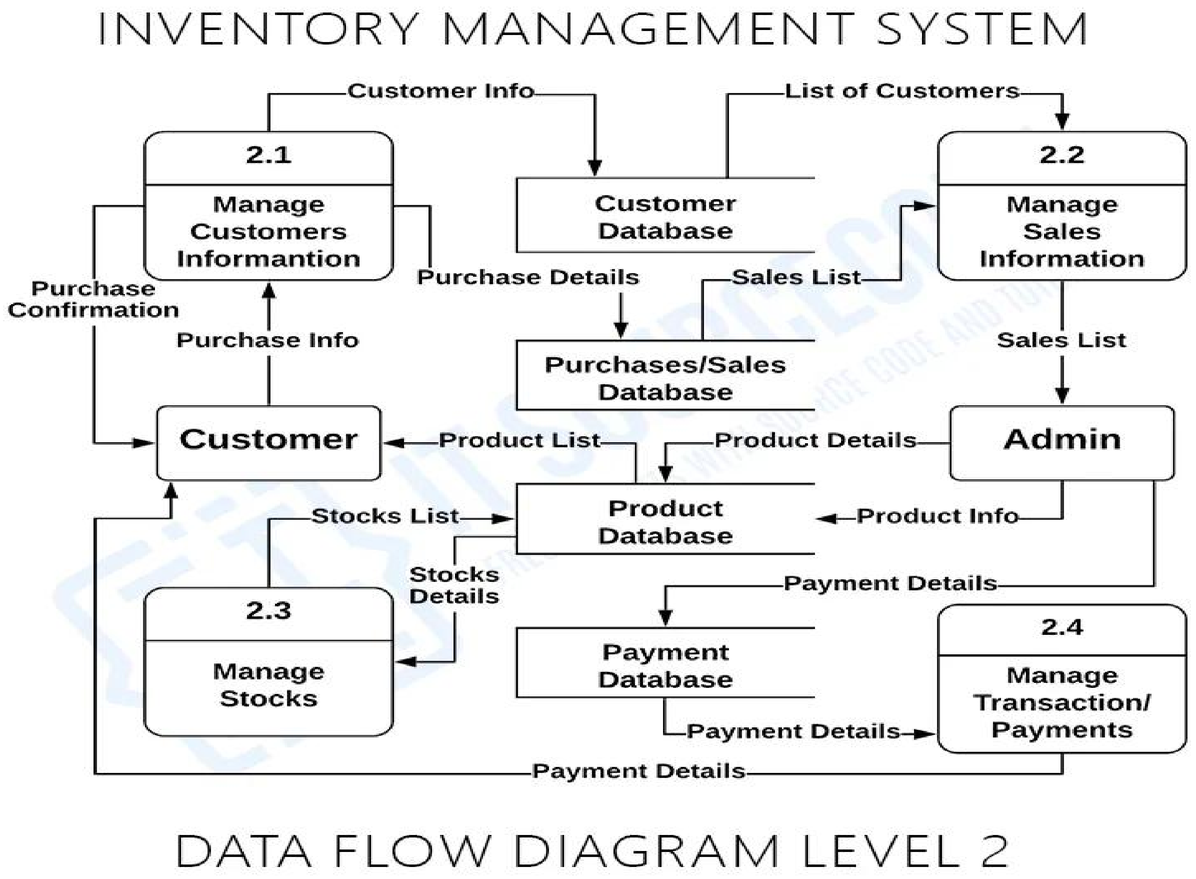


Fig 5.3

- At this point, let’s focus on a crucial aspect of the data flow diagram.

- The databases of the inventory management system are composed of:

✓ Customer Database
✓ Purchases and Sales Database
✓ Product Database
✓ Payment Database

- These data repositories are responsible for maintaining data security and accessibility. The system will only provide information to the user who requested it.

5.2 SOLUTION & TECHNICAL ARCHITECTURE

SOLUTION ARCHITECTURE

- Inventory Management is one of the basic problems in almost every company. Before computer age and integration, paper tables and paperwork solutions were being used as inventory management tools.

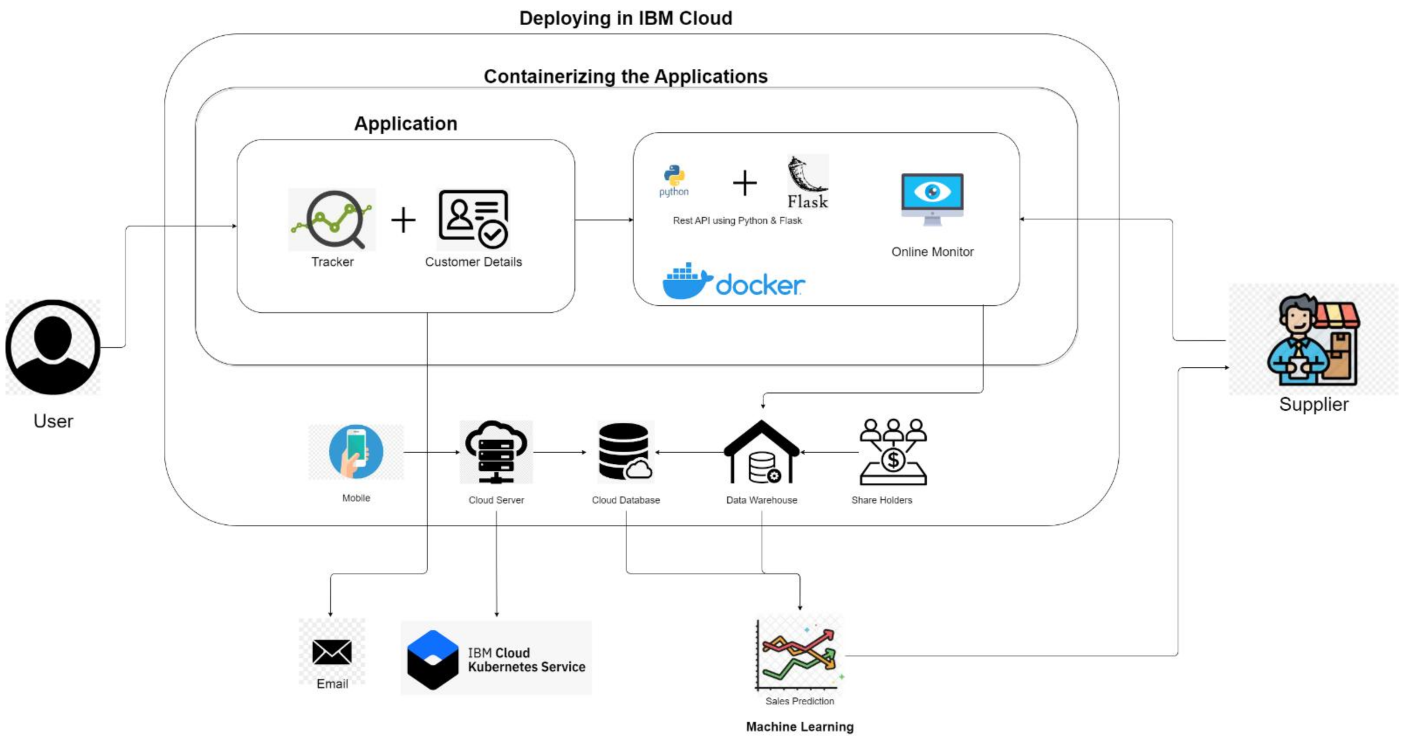


Fig 5.4

- These we very far from being a solution, took so much time, even needed employees just for this section of organization.

FLASK

- Flask is a lightweight Python web framework that provides useful tools and features for creating web applications in the Python Language. It gives developers flexibility and is an accessible framework for new developers because you can build a web application quickly using only a single Python file.

- Flask is also extensible and doesn't force a particular directory structure or require complicated boilerplate code before getting started.

DOCKER

- Docker Python is an open-source tool and a standard shipping container. This tool is used for automating the deployment of any application inside a software container.

- Docker is a containerization tool used for **spinning up isolated, reproducible application environments**. It is a popular development tool for Python developers. The tutorials and articles here will teach you how to include Docker to your development workflow and use it to deploy applications locally and to the cloud.

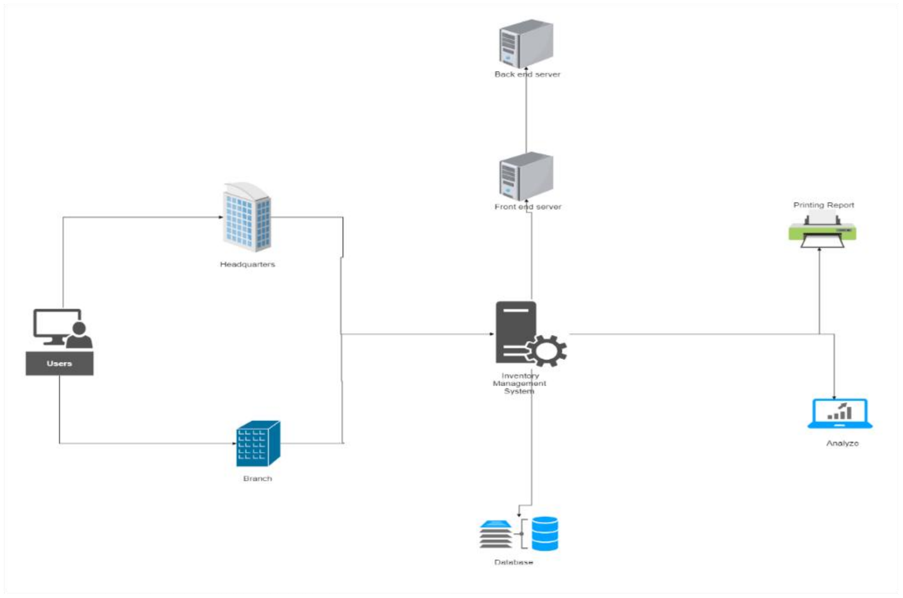


Fig 5.5

ARCHITECTURE OF INVENTORY MANAGEMENT SYSTEM

- This is an illustration of a Network Architecture for Inventory management system. Network architecture may help with security, which is becoming more critical as more consumer devices connect to the network.
- The network's design and protocols must facilitate rapid and efficient user detection and authorization.
- The Open Systems Interconnection Model, or OSI, is used in the majority of network topologies.
- This conceptual paradigm divides network jobs into seven logical levels, from the most basic to the most complex.

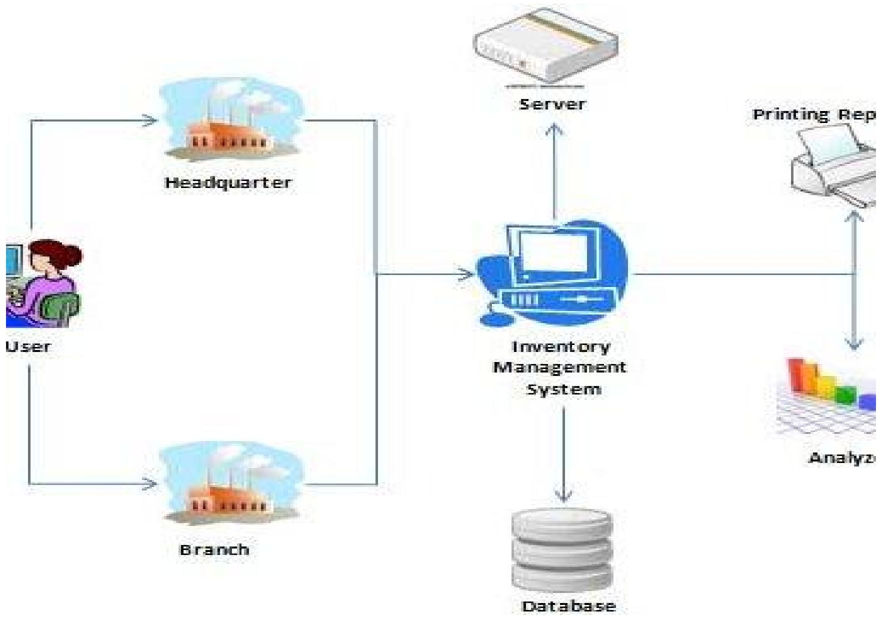


Fig 5.6

TECHNICAL ARCHITECTURE

- Technical Architecture (TA) is a form of IT architecture that is used to design computer systems.

- It involves the development of a technical blueprint regarding the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

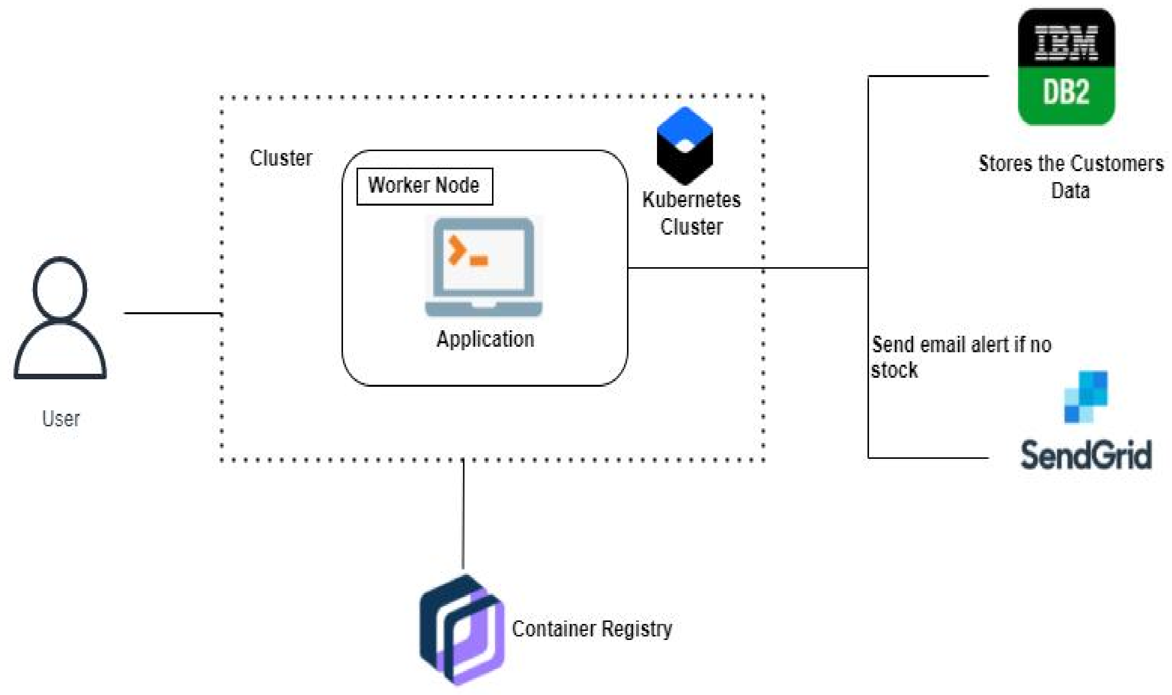


Fig 5.7

BLOCK DIAGRAM FOR INVENTORY MANAGEMENT SYSTEM

- Client sends the request operations such as adding, deleting, and upadating via internet toBusiness Logic Server

- Server to receive and process those requests and then sent via Internet to Database Server.

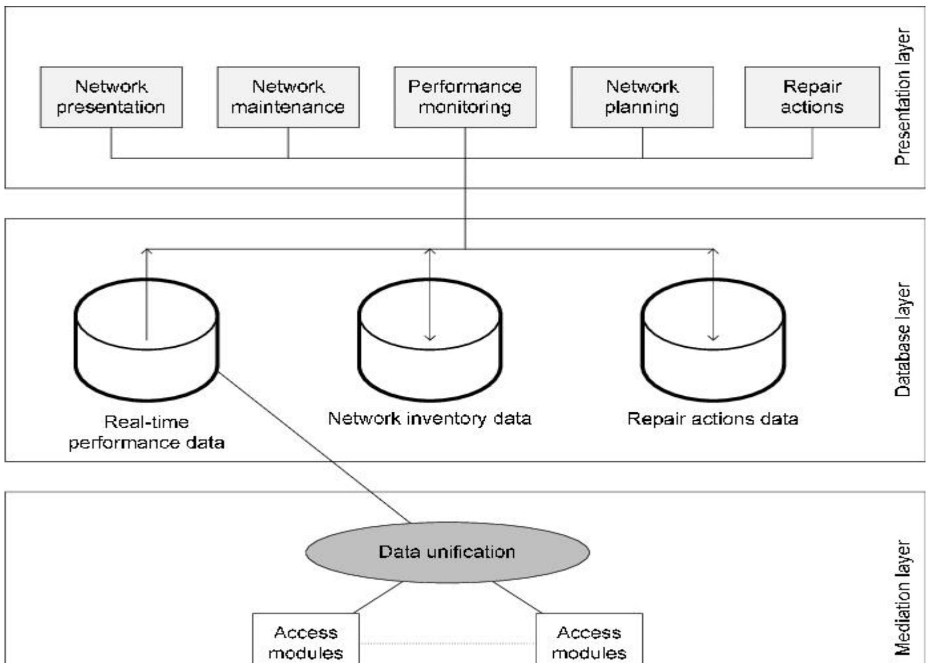


Fig 5.8

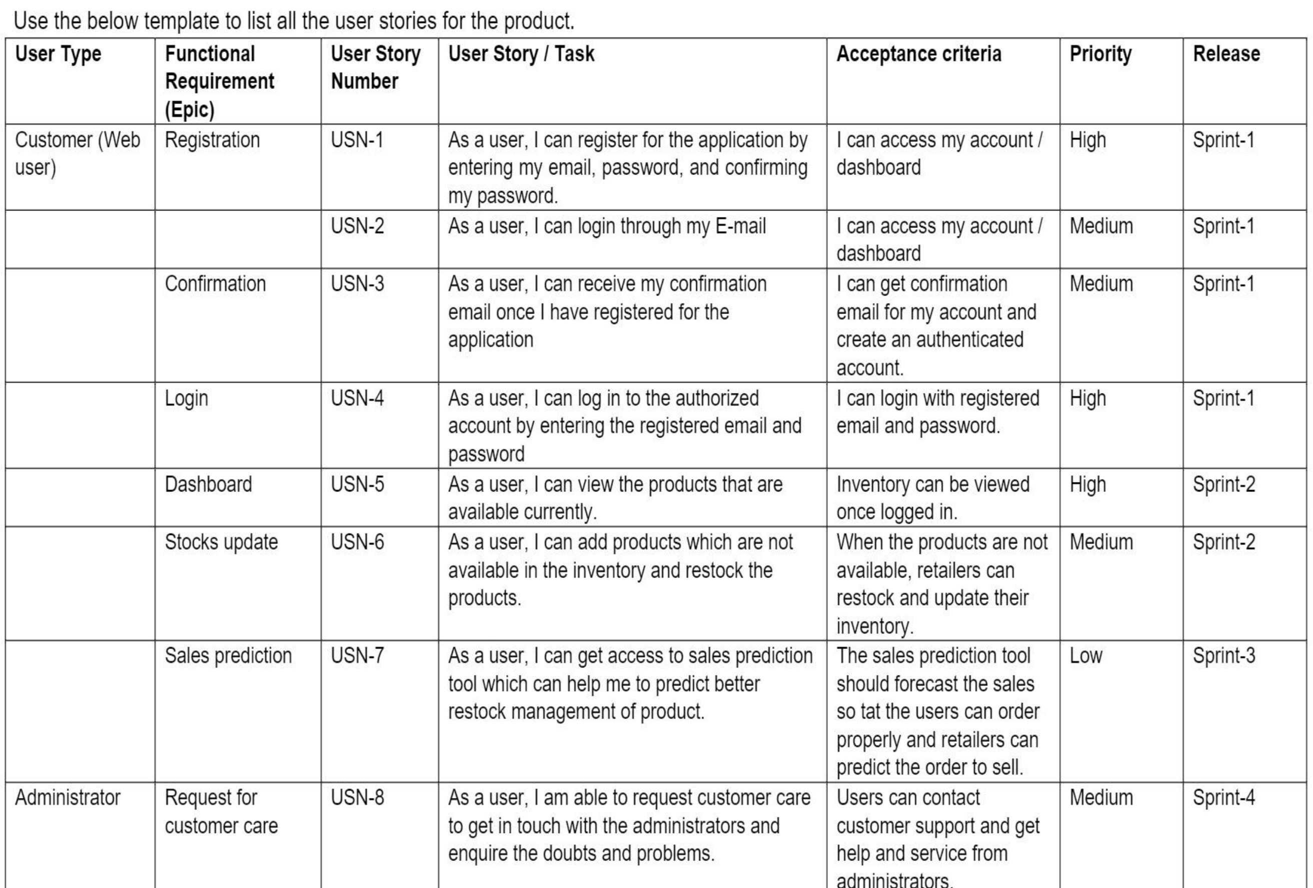
- Component diagram represents a group of related classes and other elements that work together within a self-contained logical structure in order to provide some aspect of the system

- You can easily edit this template using Create. You can export it in multiple formats like JPEG, PNG and SVG and easily add it to Word documents, PowerPoints (PPT) presentations, Excel or any other documents. You can export it as a PDF for high-quality printouts.

5.3 USER STORIES

- A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

- Perhaps the most important benefit of user stories in agile product development, is that unlike requirements or use cases, user stories are not meant to stand on their own. Instead, each user story is a placeholder for a future conversation with the development team.



Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I can login through my E-mail	I can access my account / dashboard	Medium	Sprint-1
	Confirmation	USN-3	As a user, I can receive my confirmation email once I have registered for the application	I can get confirmation email for my account and create an authenticated account.	Medium	Sprint-1
	Login	USN-4	As a user, I can log in to the authorized account by entering the registered email and password	I can login with registered email and password.	High	Sprint-1
	Dashboard	USN-5	As a user, I can view the products that are available currently.	Inventory can be viewed once logged in.	High	Sprint-2
	Stocks update	USN-6	As a user, I can add products which are not available in the inventory and restock the products.	When the products are not available, retailers can restock and update their inventory.	Medium	Sprint-2
	Sales prediction	USN-7	As a user, I can get access to sales prediction tool which can help me to predict better restock management of product.	The sales prediction tool should forecast the sales so tat the users can order properly and retailers can predict the order to sell.	Low	Sprint-3
Administrator	Request for customer care	USN-8	As a user, I am able to request customer care to get in touch with the administrators and enquire the doubts and problems.	Users can contact customer support and get help and service from administrators.	Medium	Sprint-4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

PRODUCT BACKLOG, SPRINT SCHEDULE, AND ESTIMATION

Use the below template to create product backlog and sprint schedule

Sprint	Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
SPRINT-1	REGISTRATION	USN-1	AS A USER, I CAN REGISTER FOR THE APPLICATION BY USING MY EMAIL & PASSWORD AND CONFIRMING MY LOGIN CREDENTIALS.	3	HIGH	SHEIK AMANULLAH, MOHAMED HUSSAIN KANI, AHAMED JASIR, MOHAIDDIN ABDULQADHAR
SPRINT-1		USN-2	AS A USER, I CAN LOGIN THROUGH MY E-MAIL.	3	MEDIUM	SHEIK AMANULLAH, MOHAMED HUSSAIN KANI, AHAMED JASIR, MOHAIDDIN ABDULQADHAR
SPRINT-1	CONFIRMATION	USN-3	AS A USER, I CAN RECEIVE MY CONFIRMATION EMAIL ONCE I HAVE REGISTERED FOR THE APPLICATION.	2	HIGH	SHEIK AMANULLAH, MOHAMED HUSSAIN KANI, AHAMED JASIR, MOHAIDDIN ABDULQADHAR
SPRINT-2	LOGIN	USN-4	AS A USER, I CAN LOG IN TO THE AUTHORIZED ACCOUNTBY ENTERING THE REGISTERED EMAIL AND PASSWORD.	3	MEDIUM	SHEIK AMANULLAH, MOHAMED HUSSAIN KANI, AHAMED JASIR, MOHAIDDIN ABDULQADHAR
SPRINT-2	DASHBOARD	USN-4	AS A USER, I CAN VIEW THE PRODUCTS THAT AREAVAILABLE CURRENTLY.	4	HIGH	SHEIK AMANULLAH, MOHAMED HUSSAIN KANI, AHAMED JASIR, MOHAIDDIN ABDUL QADHAR
SPRINT-3	STOCKS UPDATE	USN-4	AS A USER, I CAN ADD PRODUCTS WHICH ARE NOTAVAILABLE IN THE INVENTORY AND RESTOCK THE PRODUCTS.	3	MEDIUM	SHEIK AMANULLAH, MOHAMED HUSSAIN KANI, AHAMED JASIR, MOHAIDDIN ABDUL QADHAR
SPRINT-4	SALES PREDICTION	USN-4	AS A USER, I AM ABLE TO REQUEST CUSTOMER CARETO	6	MEDIUM	SHEIK AMANULLAH, MOHAMED

			GET IN TOUCH WITH THE ADMINISTRATORS AND ENQUIRE THE DOUBTS AND PROBLEMS.			HUSSAIN KANI, AHAMED JASIR, MOHAIDDIN ABDUL QADHAR
SPRINT-4	REQUEST FOR CUSTOMERCARE	USN-4	AS A USER, I AM ABLE TO SEND FEEDBACK FORMS REPORTING ANY IDEAS FOR IMPROVING OR RESOLVINGANY ISSUES I AM FACING TO GET IT RESOLVED.	3	MEDIUM	SHEIK AMANULLAH, MOHAMED HUSSAIN KANI, AHAMED JASIR, MOHAIDDIN ABDUL QADHAR

PROJECT TRACKER, VELOCITY & BURNDOWN CHART:

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	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint-2	7	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022
Sprint-4	7	6 Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

6.2 SPRINT DELIVERY SCHEDULE

6.2.1 SPRINT 1

Creating flask application:

Step 1

Create python flask app named app.py

Step 2:

Create templates;

additems, base, brands, dashboard, index, main, show, signin, signup, stores.

Step 3:

Add docker file

Step 4:

Add kubernetes yaml files;
Flask_deployment, flask_ingress, flask_services, ibm_deployment

Step 5:

Add DigiCertGlobalRootCA.crt file

Step 6:

Setup environment by following these steps:
(https://www.geeksforgeeks.org/sending-emails-using-api-in-flask-mail/)

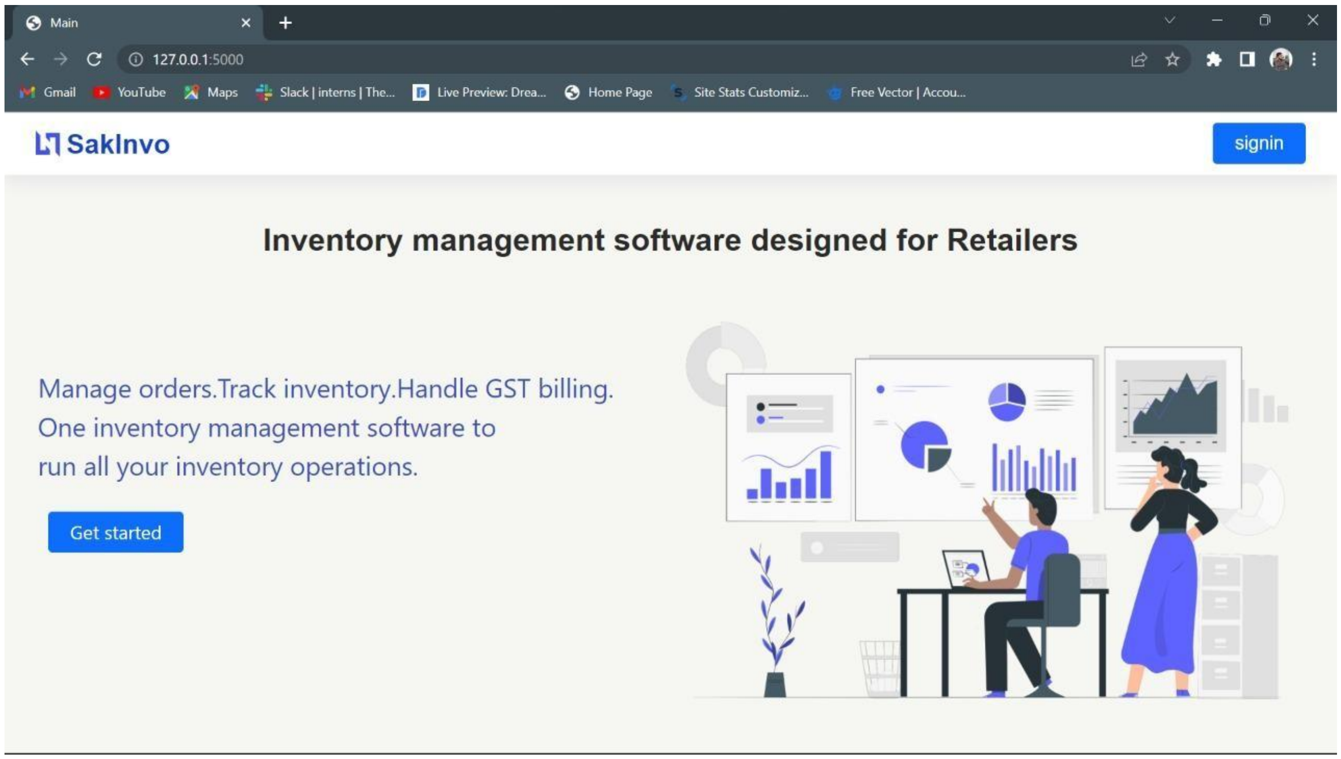
Step 7:

Add requirements.txt

Step 8:

Run the flask app by running this in
command prompt: flask --debug run

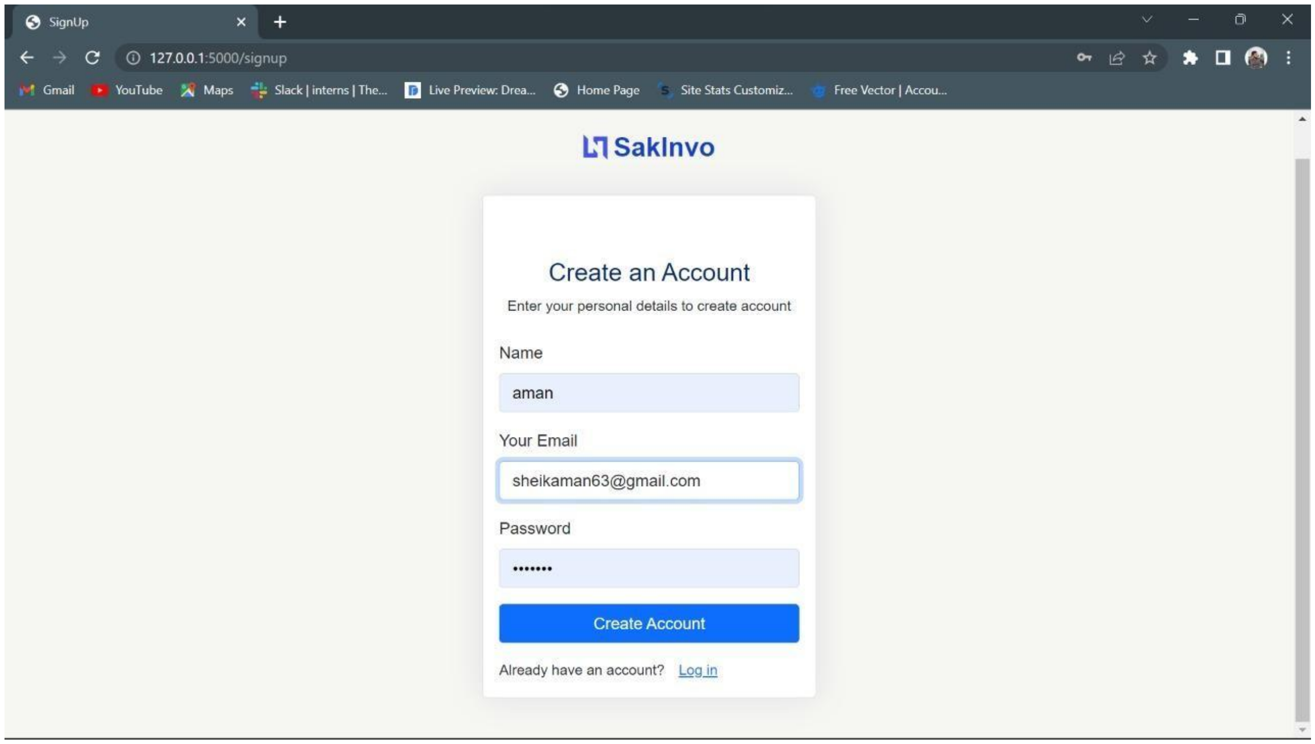
Main Page:



Inventory management software designed for Retailers

Manage orders.Track inventory.Handle GST billing.
One inventory management software to
run all your inventory operations.

Sign-Up Page for Creating an Account:



Create an Account

Enter your personal details to create account

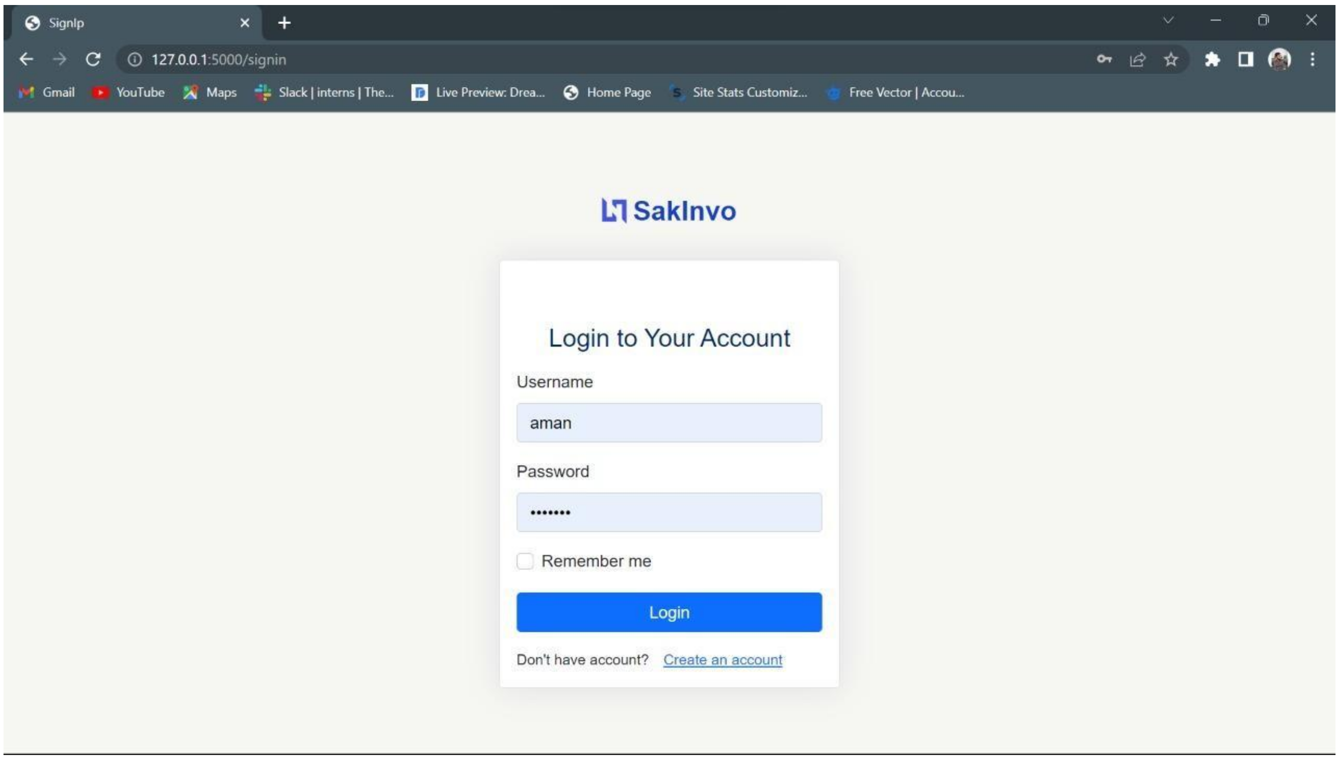
Name: aman

Your Email: sheikaman63@gmail.com

Password: •••••••

Already have an account? Log in

Sign-In Page for Logging into the Inventory Management System:


SignIp
127.0.0.1:5000/signin

SakInvo

Login to Your Account

Username: aman

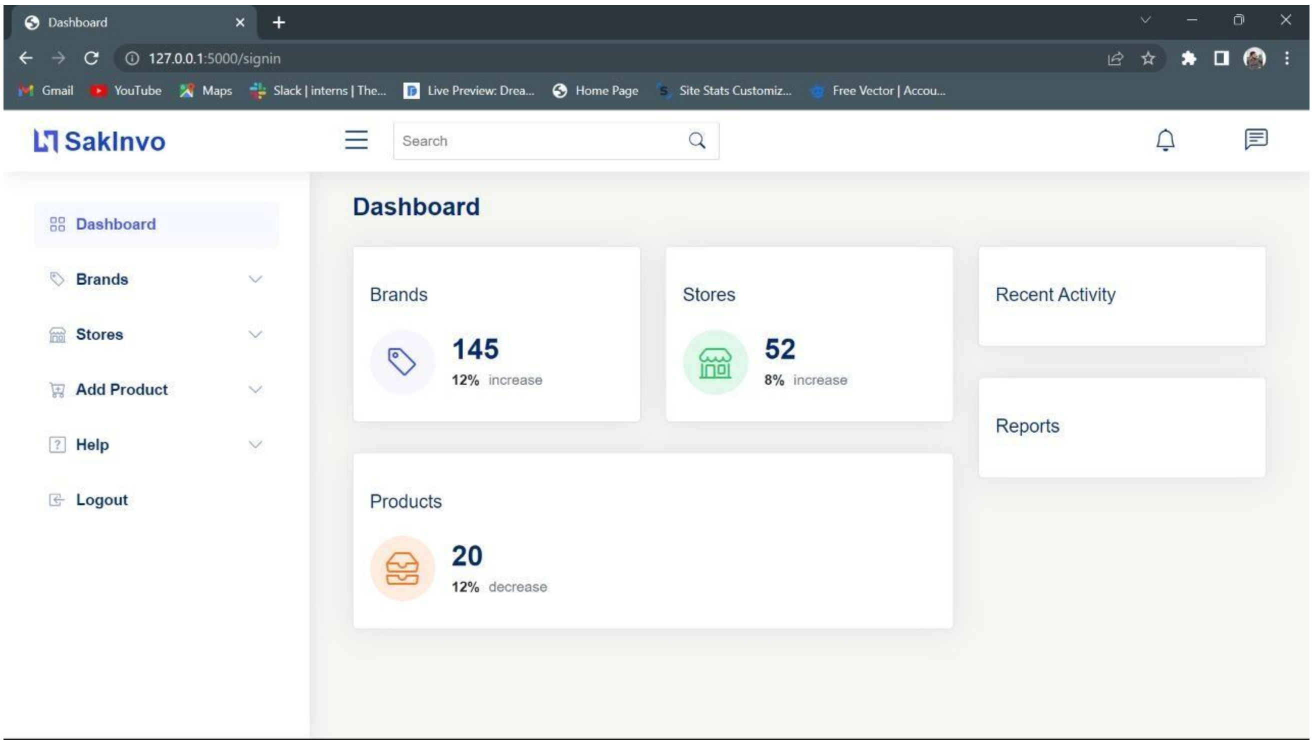
Password: •••••••

Remember me

Login

Don't have account? Create an account

Dashboard Page:


Dashboard
127.0.0.1:5000/signin

SakInvo

Dashboard

Brands
145
12% increase

Stores
52
8% increase

Recent Activity

Reports

Products
20
12% decrease

6.2.2 SPRINT 2

Creating dashboard and stocks update information:

Step 1:
To view the dashboard information, click on dashboard tab on the left

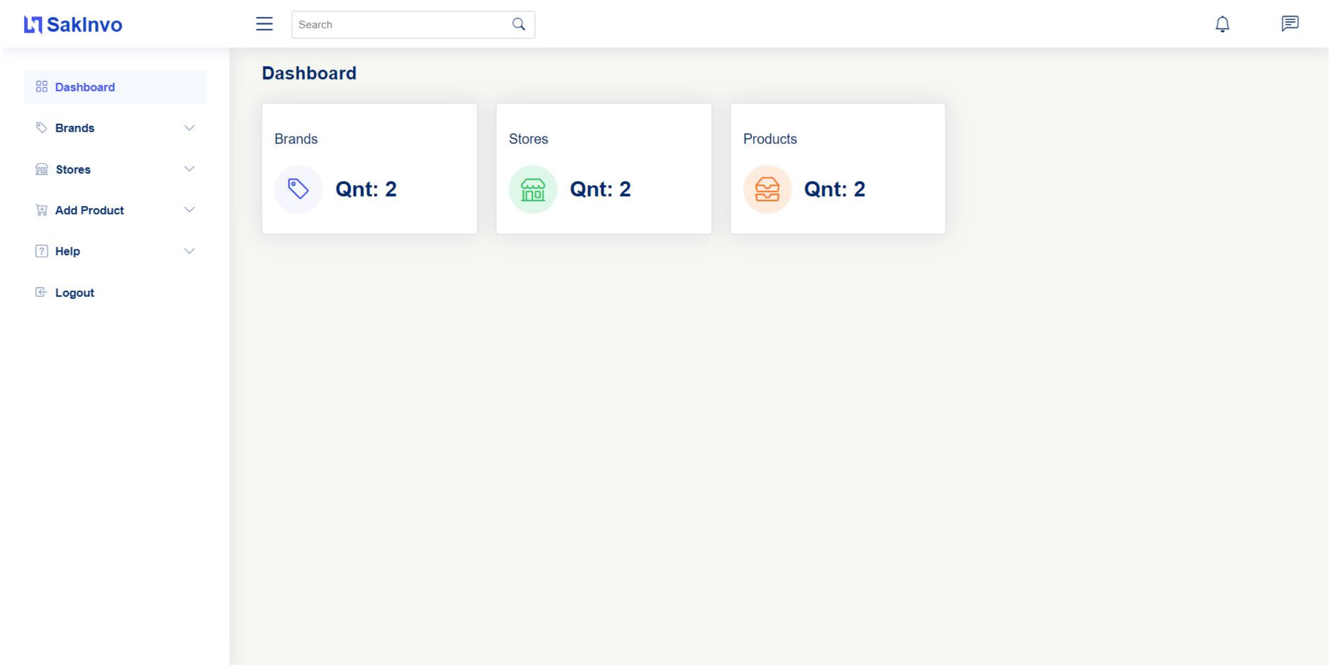
Step 2:
Add Brand, Products,Stores according to your needs in the side bar;

Step 3:
View the added items

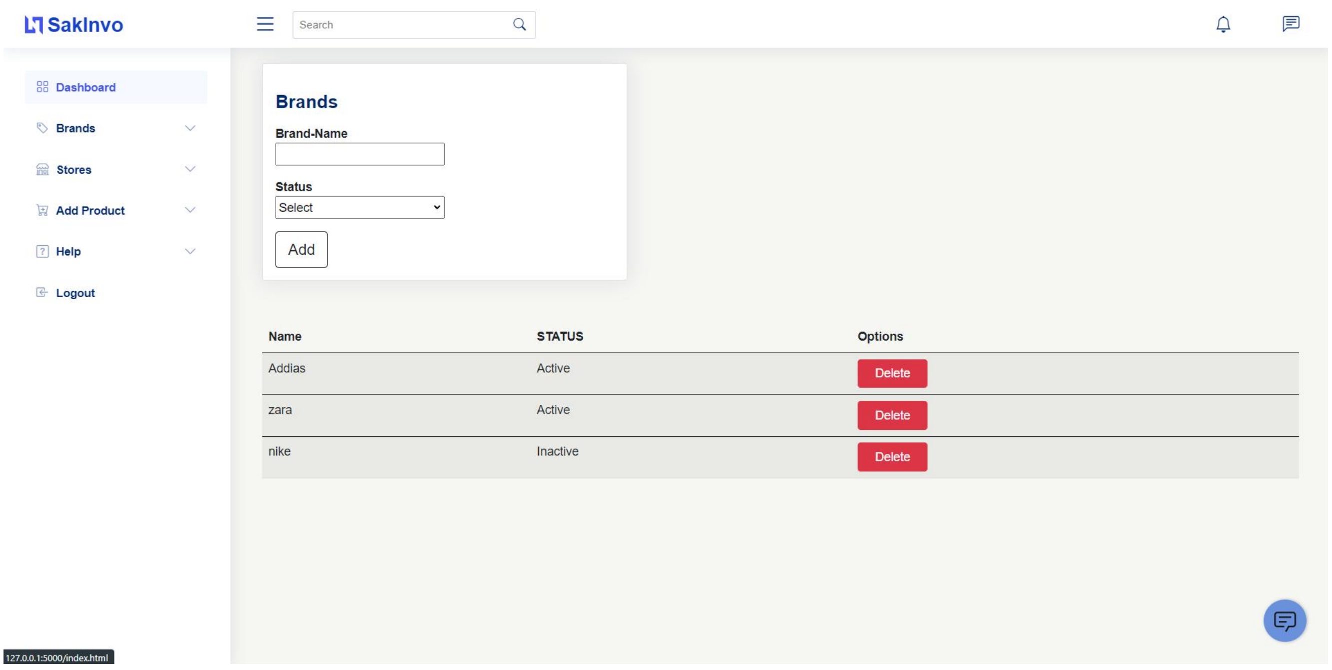
Step 4:
View the number of items on dashboard page.

Output:

Dashboard Page:



Add and view Brands:



Brands

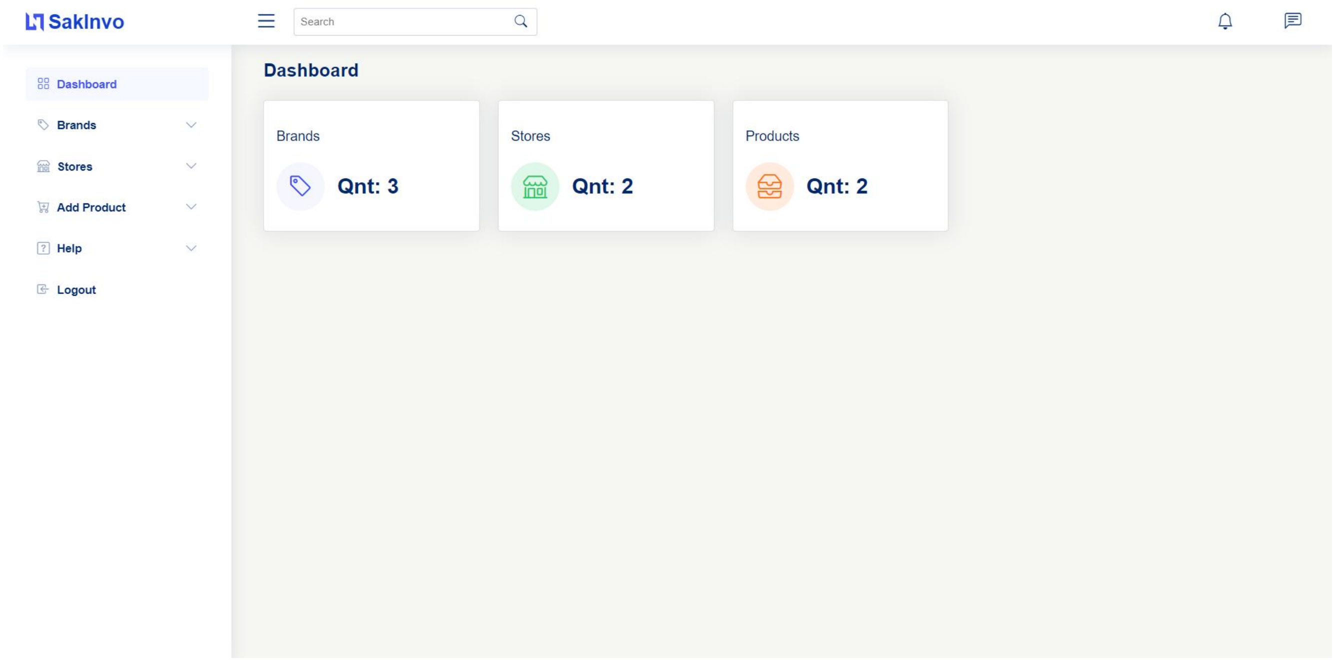
Brand-Name

Status
Select

Add

Name	STATUS	Options
Addias	Active	Delete
zara	Active	Delete
nike	Inactive	Delete

View number of Brands on dashboard:



Dashboard

Brands	Stores	Products
Qnt: 3	Qnt: 2	Qnt: 2

6.2.3 SPRINT 3

Creating IBM Db2 database and chat-box using IBM Watson Assistant:

Creating IBM Db2 Database and Connecting it:

Step 1:
Go to IBM cloud resource list and click on database.

Step 2:
Create a database and use the service credentials on your python flask to connect to IBM Db2 database services.

Step 3:
Click on ‘Go to UI’ and click ‘Data’ on the left side.

Step 4:
Click tables and select the name of your database.

Step 5:
Create new table according to the database you need.

Step 6:
Verify it is working.

Creating Chat-Box using IBM Watson Assistant:

Step 1:
Go to IBM cloud resource list and click on IBM Watson Assistant.

Step 2:
Click on launch Watson Assistant.

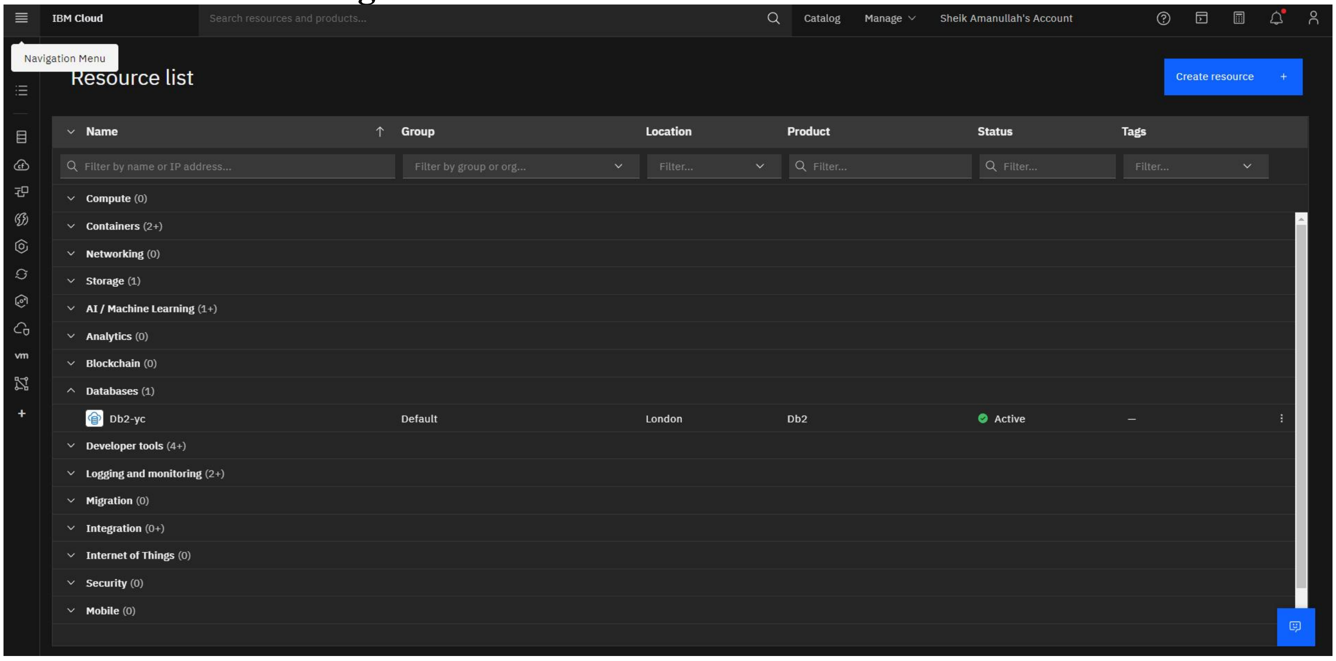
Step 3:
Build your virtual assistant.

Step 4:
Add the script to your python file.

Step 5:
Verify on our application page.

IBM Db2 Database Output:

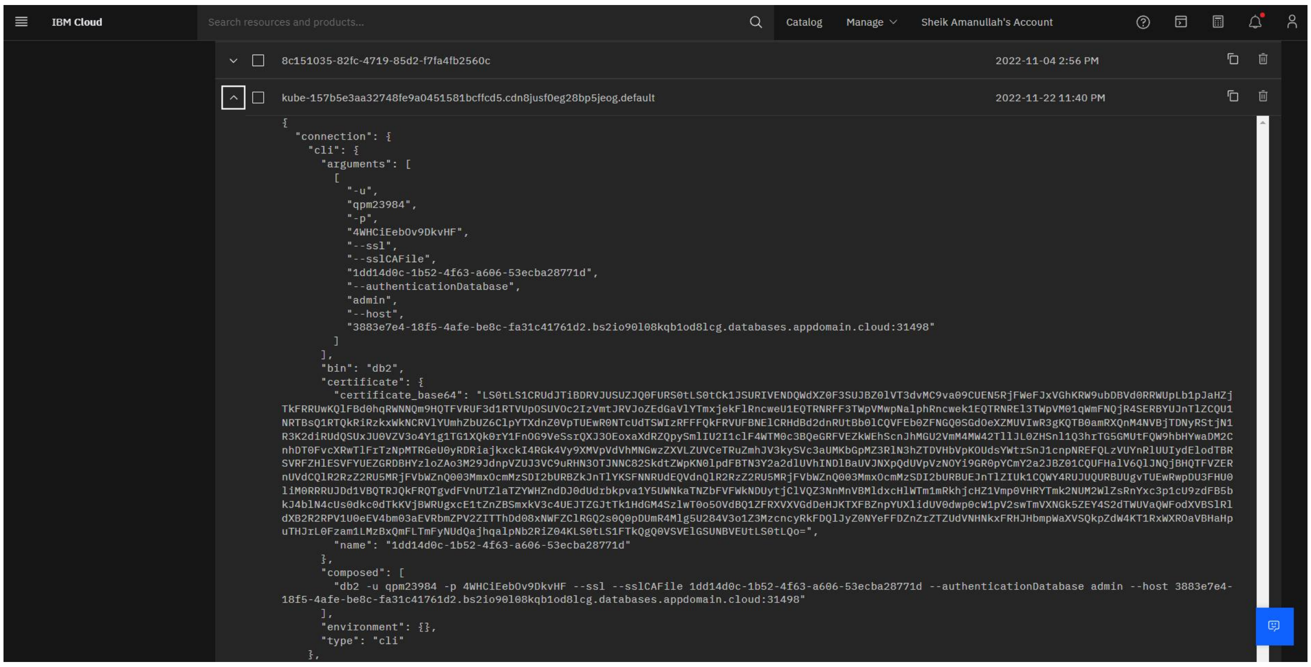
IBM Resource List Page:



Resource list

Name	Group	Location	Product	Status	Tags
Compute (0)					
Containers (2+)					
Networking (0)					
Storage (1)					
AI / Machine Learning (1+)					
Analytics (0)					
Blockchain (0)					
Databases (1)					
Db2-yc	Default	London	Db2	● Active	–
Developer tools (4+)					
Logging and monitoring (2+)					
Migration (0)					
Integration (0+)					
Internet of Things (0)					
Security (0)					
Mobile (0)					

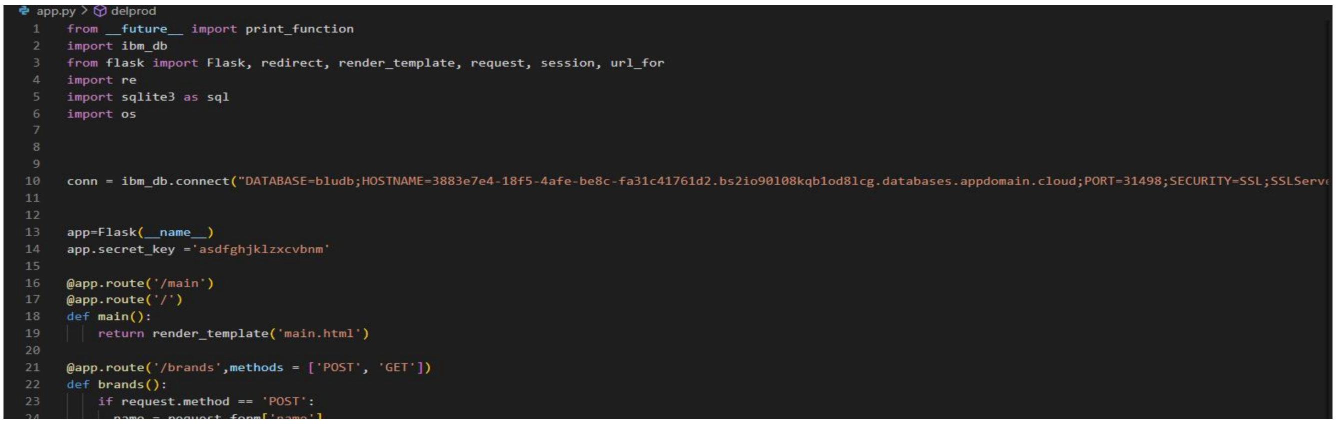
Db2 Service Credentials:



```
8c151035-82fc-4719-85d2-f7fa4fb2560c                                  2022-11-04 2:56 PM
kube-157b5e3aa32748fe9a0451581bcffcd5.cdn8jusf0eg28bp5jeog.default    2022-11-22 11:40 PM
```

```
{
  "connection": {
    "cli": {
      "arguments": [
        [
          "-u",
          "qpm23984",
          "-p",
          "4WHCiEebOv9DkvHF",
          "--ssl",
          "--sslCAFile",
          "1dd14d0c-1b52-4f63-a606-53ecba28771d",
          "--authenticationDatabase",
          "admin",
          "--host",
          "3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498"
        ]
      ],
      "bin": "db2",
      "certificate": {
        "certificate_base64": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURIVENDQWdXZ0F3SUJBZ0lVT3dvMC9va09CUEN5RjFWeFJxVGhKRW9ubDBVd0RRWUpLb1pJaHZj
TkFRRUwKQlFBd0hqRWNNQm9HQTFVRUF3d1RTVUpOSUVOc2IzVmtJRVJoZEdGaVlYTmxjekFlRncweU1EQTRNRFF3TWpVMwpNalphRncwek1EQTRNREl3TWpVM01qWmFNQjR4SERBYUJnTlZCQU1
NRTBsQ1RTQkRiRzkxWkNCRVlYUmhZbUZ6ClpYTXdnZ0VpTUEwR0NTcUdTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBb0lCQVFEb0ZFNGQ0SGdOeXZMUVIwR3gKQTB0amRXQnM4NVBjTDNyRStjN1
R3K2diRUdQSUxJU0VZV3o4Y1g1TG1XQk0rY1FnOG9VeSsrQXJ3OEoxaXdRZQpySmlIU2I1clF4WTM0c3BQeGRFVEZkWEhScnJhMGU2VmM4MW42TllJL0ZHSnl1Q3hrTG5GMUtFQW9hbHYwaDM2C
nhDT0FvcXRwTlFrTzNpMTRGeU0yRDRiajkxckI4RGk4Vy9XMVpVdVhMNGwzZXVLZUVCeTRuZmhJV3kySVc3aUMKbGpMZ3RlN3hZTDVHbVpKOUdsYWtrSnJ1cnpNREFQLzVUYnRlUUIydElodTBR
SVRFZHlESVFYUEZGRDBHYzloZAo3M29JdnpVZUJ3VC9uRHN3OTJNNC82SkdtZWpKN0lpdFBTN3Y2a2dlUVhINDlBaUVJNXpQdUVpVzNOYi9GR0pYCmY2a2JBZ01CQUFHalV6QlJNQjBHQTFVZER
nUVdCQlR2RzZ2RU5MRjFVbWZnQ003MmxOcmMzSDI2bURBZkJnTlYKSFNNRUdEQVdnQlR2RzZ2RU5MRjFVbWZnQ003MmxOcmMzSDI2bURBUEJnTlZIUk1CQWY4RUJUQURBUUgvTUEwRwpDU3FHU0
liM0RRRUJDd1VBQTRJQkFRQTgvdFVnUTZlaTZYWHZndDJ0dUdrbkpva1Y5UWNkaTNZbFVFWkNDUytjClVQZ3NnMnVBMldxcHlWTm1mRkhjcHZ1Vmp0VHRYTmk2NUM2WlZsRnYxc3p1cU9zdFB5b
kJ4blN4cUs0dkc0dTkKVjBWRUgxcE1tZnZBSmxkV3c4UEJTZGJtTk1HdGM4SzlwT0o5OVdBQ1ZFRXVXVGdDeHJKTXFBZnpYUXlidUV0dwp0cW1pV2swTmVXNGk5ZEY4S2dTWUVaQWFodXVBSlRl
dXB2R2RPV1U0eEV4bm03aEVRbmZPV2ZITThDd08xNWFZClRGQ2s0Q0pDUmR4Mlg5U284V3o1Z3MzcncyRkFDQlJyZ0NYeFFDZnZrZTZUdVNHNkxFRHJHbmpWaXVSQkpZdW4KT1RxWXROaVBHaHp
uTHJrL0Fzam1LMzBxQmFLTmFyNUdQajhqalpNb2RiZ04KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=",
        "name": "1dd14d0c-1b52-4f63-a606-53ecba28771d"
      },
      "composed": [
        "db2 -u qpm23984 -p 4WHCiEebOv9DkvHF --ssl --sslCAFile 1dd14d0c-1b52-4f63-a606-53ecba28771d --authenticationDatabase admin --host 3883e7e4-
18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498"
      ],
      "environment": {},
      "type": "cli"
    },
```

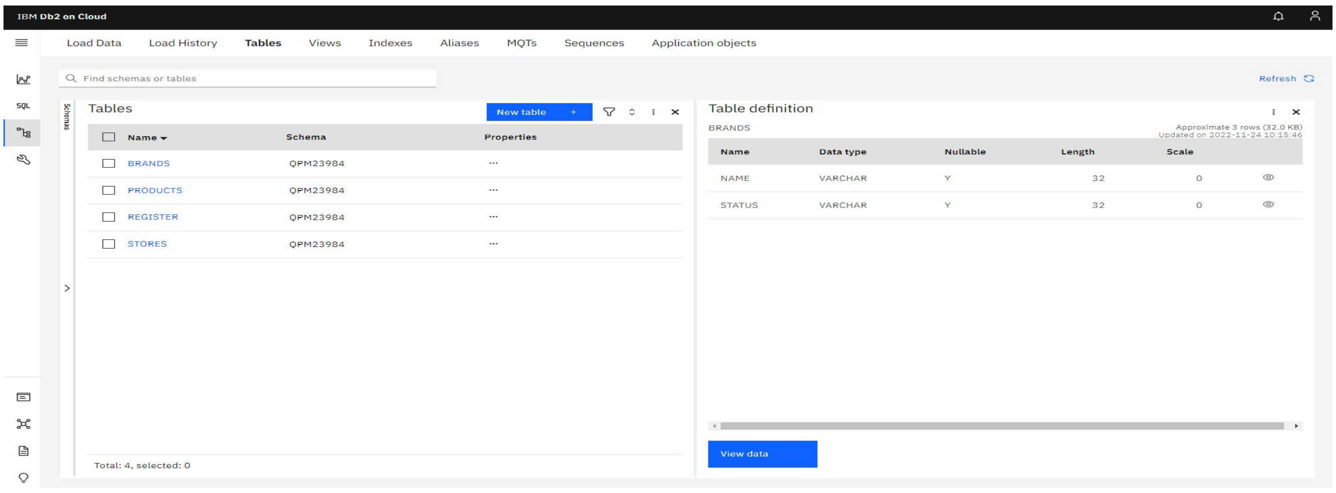
Linking our python flask application with IBM Db2:



app.py > delprod

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



conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31498;SECURITY=SSL;SSLServe


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

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

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

Going to IBM Db2 database :



Tables

Name	Schema	Properties
BRANDS	QPM23984	···
PRODUCTS	QPM23984	···
REGISTER	QPM23984	···
STORES	QPM23984	···

Total: 4, selected: 0

Table definition

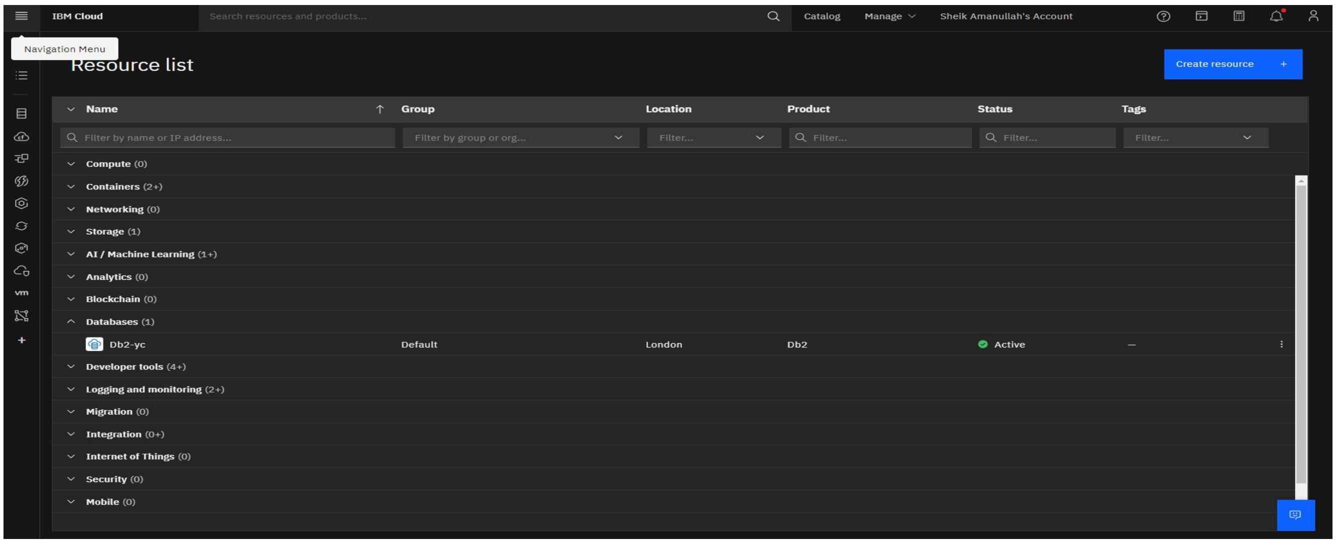
BRANDS

Approximate 3 rows (32.0 KB)
Updated on 2022-11-24 10:15:46

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	32	0
STATUS	VARCHAR	Y	32	0

Chat-Box Output:

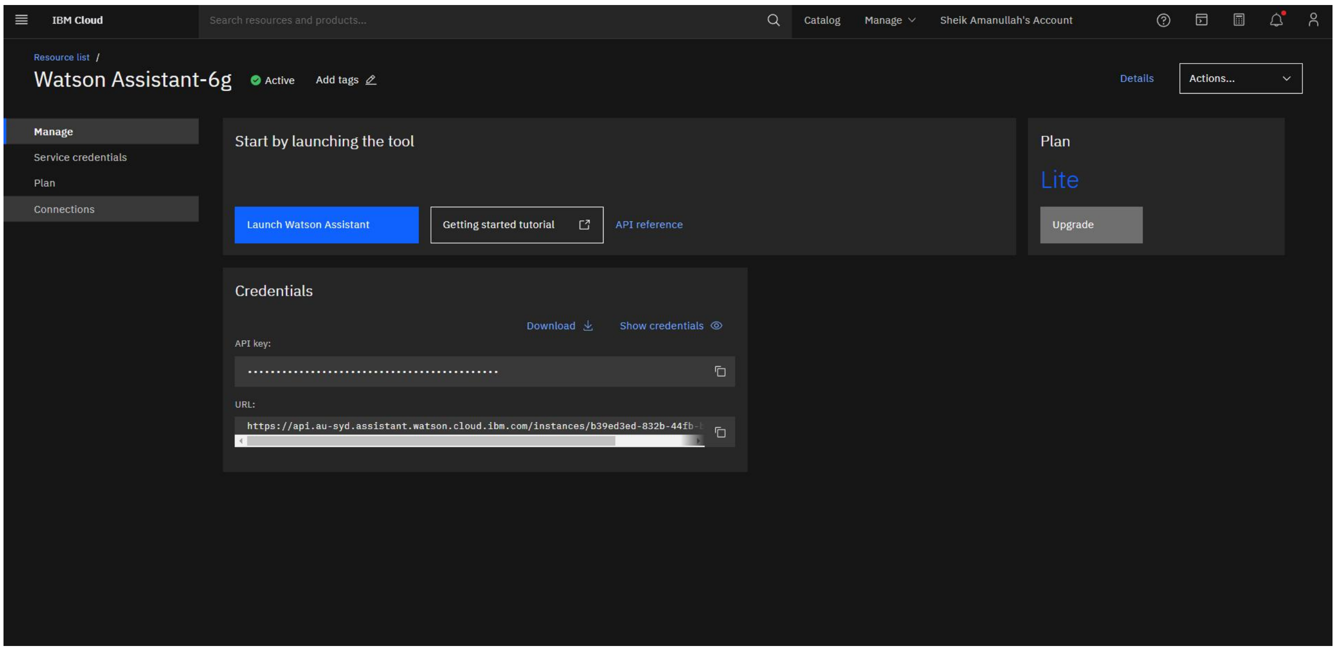
IBM Resource List Page:



Resource list

Name	Group	Location	Product	Status	Tags
Compute (0)					
Containers (2+)					
Networking (0)					
Storage (1)					
AI / Machine Learning (1+)					
Analytics (0)					
Blockchain (0)					
Databases (1)					
Db2-yc	Default	London	Db2	Active	–
Developer tools (4+)					
Logging and monitoring (2+)					
Migration (0)					
Integration (0+)					
Internet of Things (0)					
Security (0)					
Mobile (0)					

IBM Watson Service Page on IBM Cloud:



Watson Assistant-6g Active Add tags

Start by launching the tool

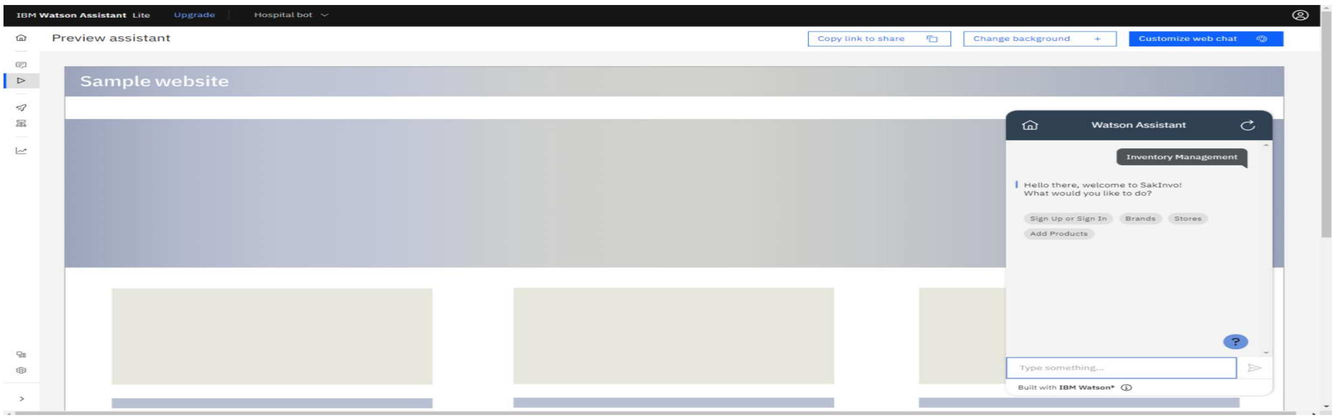
Plan
Lite

Credentials

API key:
···

URL:
https://api.au-syd.assistant.watson.cloud.ibm.com/instances/b39ed3ed-832b-44fb-

Launch Watson Assistant and create a virtual chat-box:

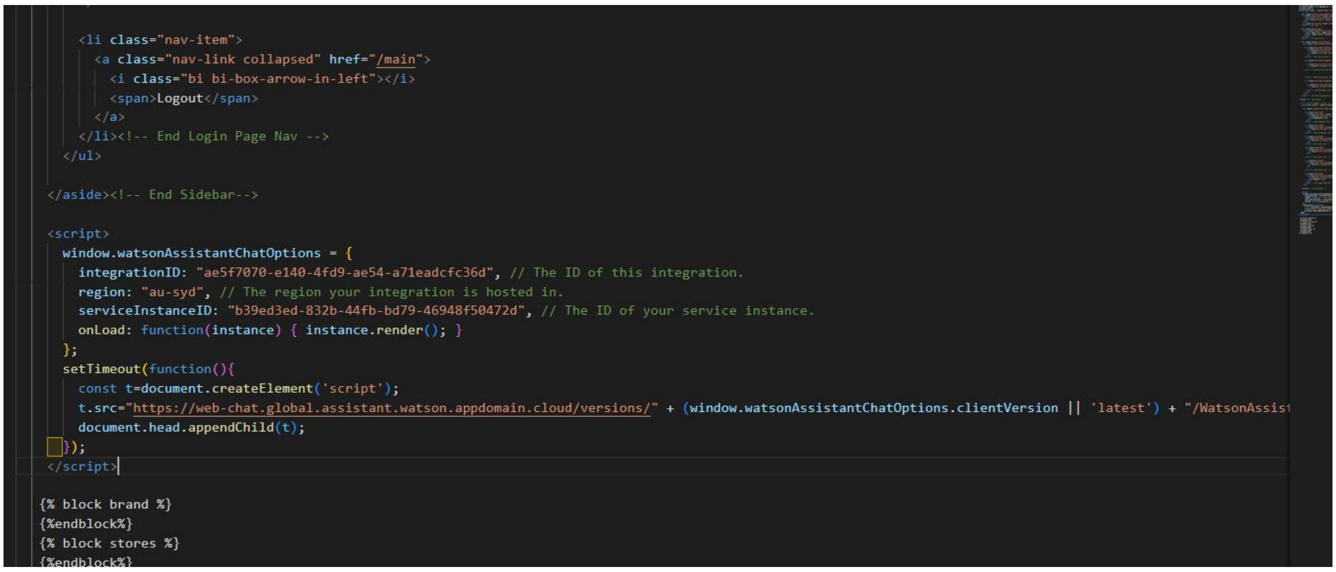


Inventory Management

Hello there, welcome to SakInvo!
What would you like to do?

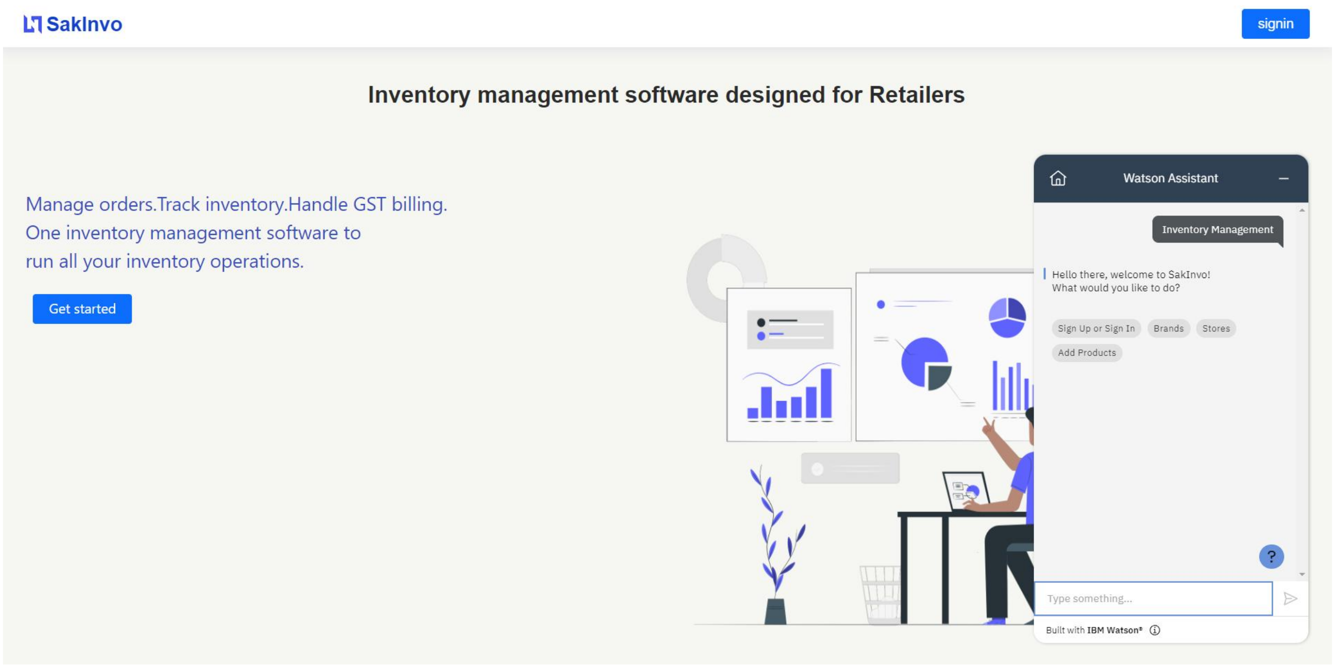
Sign Up or Sign In
Brands
Stores
Add Products

Embed it on your python file:



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

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

  <script>
    window.watsonAssistantChatOptions = {
      integrationID: "ae5f7070-e140-4fd9-ae54-a71eadcfc36d", // The ID of this integration.
      region: "au-syd", // The region your integration is hosted in.
      serviceInstanceID: "b39ed3ed-832b-44fb-bd79-46948f50472d", // The ID of your service instance.
      onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
      const t=document.createElement('script');
      t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssist
      document.head.appendChild(t);
  });
  </script>

 {% block brand %}
 {%endblock%}
 {% block stores %}
 {%endblock%}
```

Verifying on our page:



Inventory management software designed for Retailers

Manage orders.Track inventory.Handle GST billing.
One inventory management software to
run all your inventory operations.

Hello there, welcome to SakInvo!
What would you like to do?

6.2.4 SPRINT 4

Deploying it on our Kubernetes services:

Step 1:

Open command prompt in the location where the project file is in.

Step 2:

Login to docker, ibmcloud and ibmcloud cr by using [docker login], [ibmcloud login] and [ibmcloud cr login] commands.

Step 3:

Build the image to docker hub.

Step 4:

Tag the docker image with our IBM container register namespaces.

Step 5:

Push to the namespaces of IBM container registry.

Step 6:

View the pushed image on IBM container registry.

Step 5:

Launch Kubernetes service and connect via CLI

Step 5:

Apply the yaml files using command prompt.

Step 5:

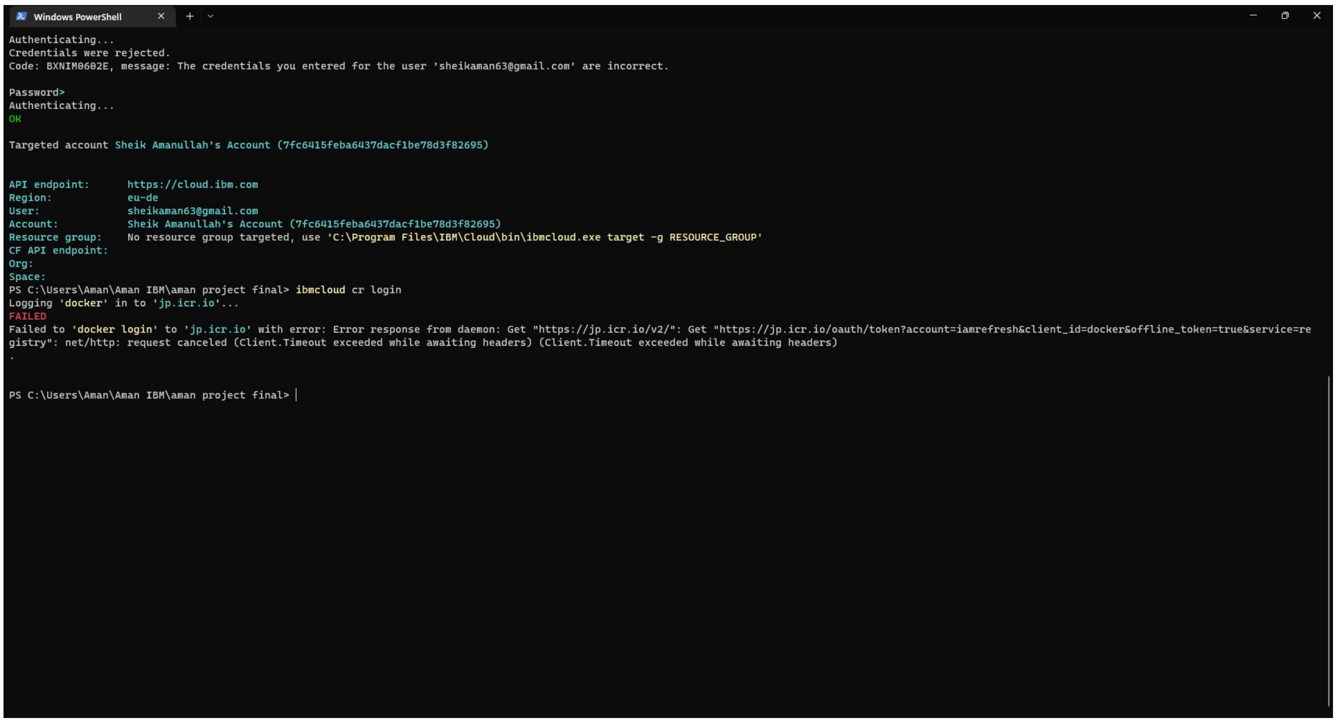
View the public IP and port number.

Step 5:

Go to the Public IP with the respected port number.

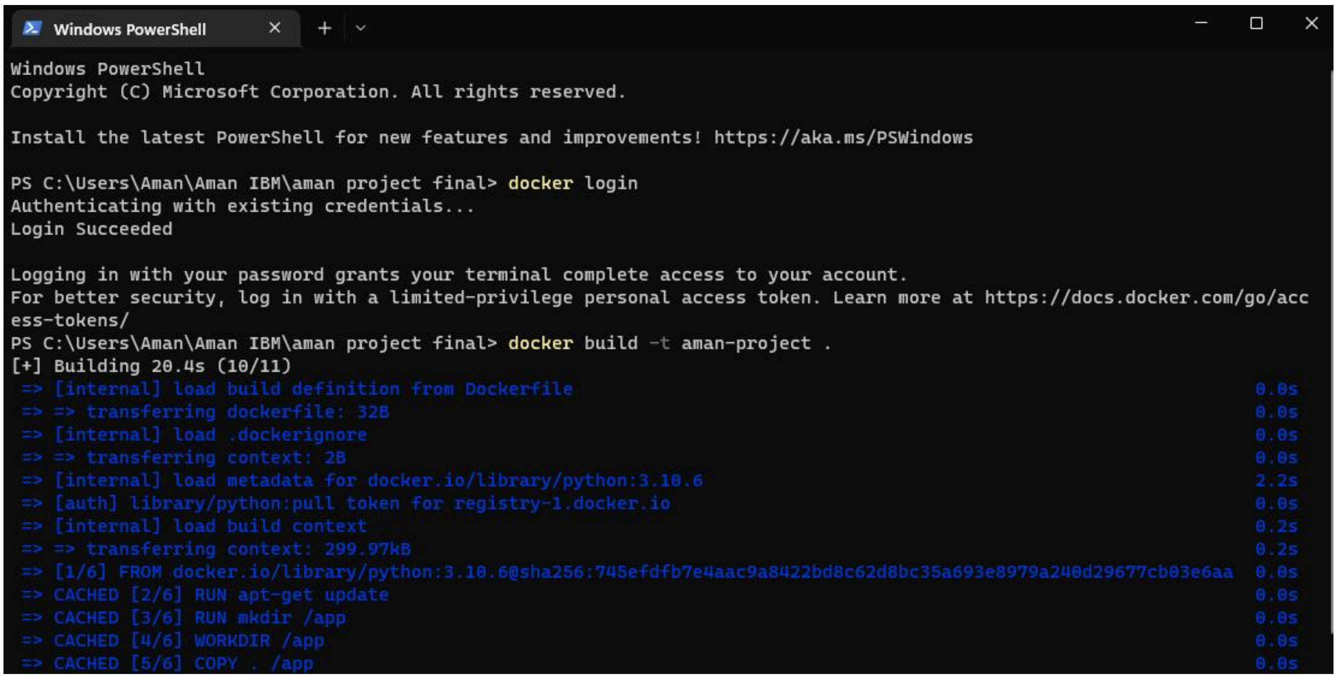
Output:

Login to the docker, ibmcloud & ibmcloud cr in Command Prompt:

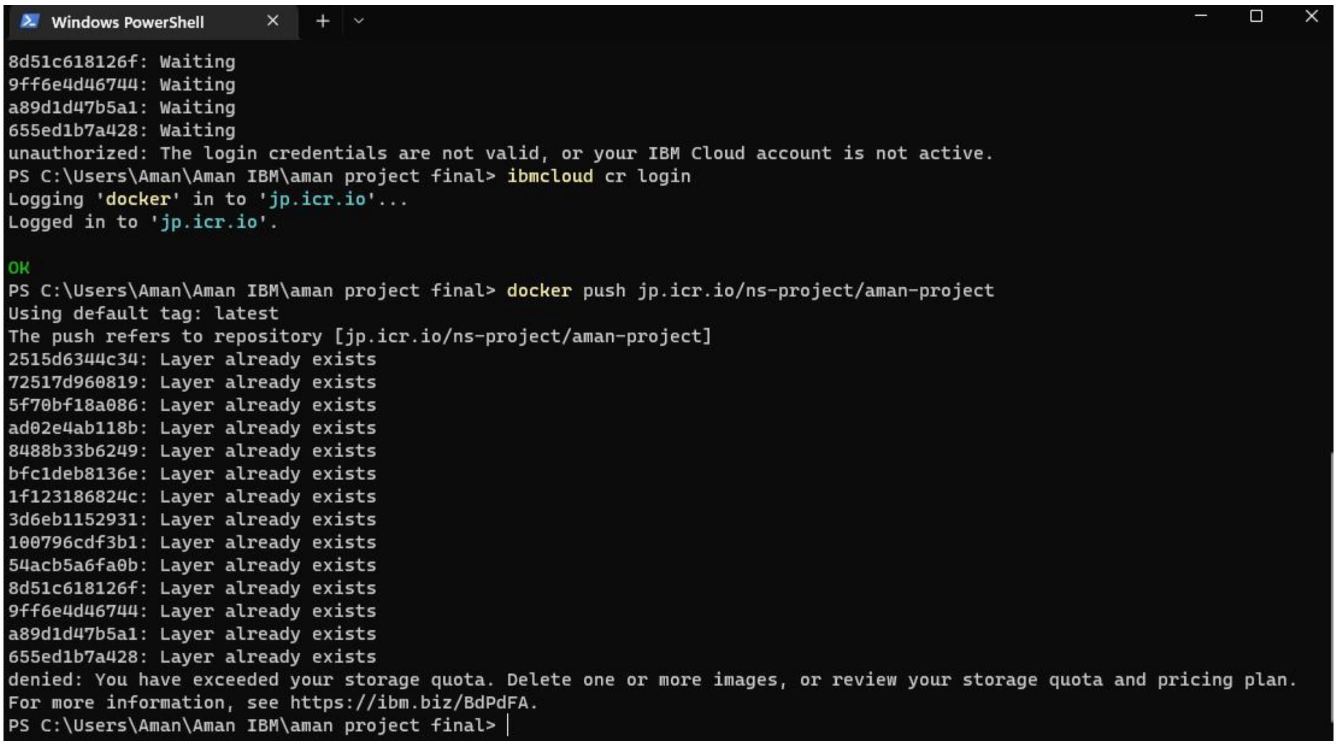


```
Authenticating...
Credentials were rejected.
Code: BXNIM0602E, message: The credentials you entered for the user 'sheikaman63@gmail.com' are incorrect.

Password>
Authenticating...
OK

Targeted account Sheik Amanullah's Account (7fc6415feba6437dacf1be78d3f82695)


API endpoint:      https://cloud.ibm.com
Region:            eu-de
User:              sheikaman63@gmail.com
Account:           Sheik Amanullah's Account (7fc6415feba6437dacf1be78d3f82695)
Resource group:    No resource group targeted, use 'C:\Program Files\IBM\Cloud\bin\ibmcloud.exe target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:
PS C:\Users\Aman\Aman IBM\aman project final> ibmcloud cr login
Logging 'docker' in to 'jp.icr.io'...
FAILED
Failed to 'docker login' to 'jp.icr.io' with error: Error response from daemon: Get "https://jp.icr.io/v2/": Get "https://jp.icr.io/oauth/token?account=iamrefresh&client_id=docker&offline_token=true&service=re
gistry": net/http: request canceled (Client.Timeout exceeded while awaiting headers) (Client.Timeout exceeded while awaiting headers)
.


PS C:\Users\Aman\Aman IBM\aman project final> |
```

Building, Tagging and Pushing the Image to Container Registry:

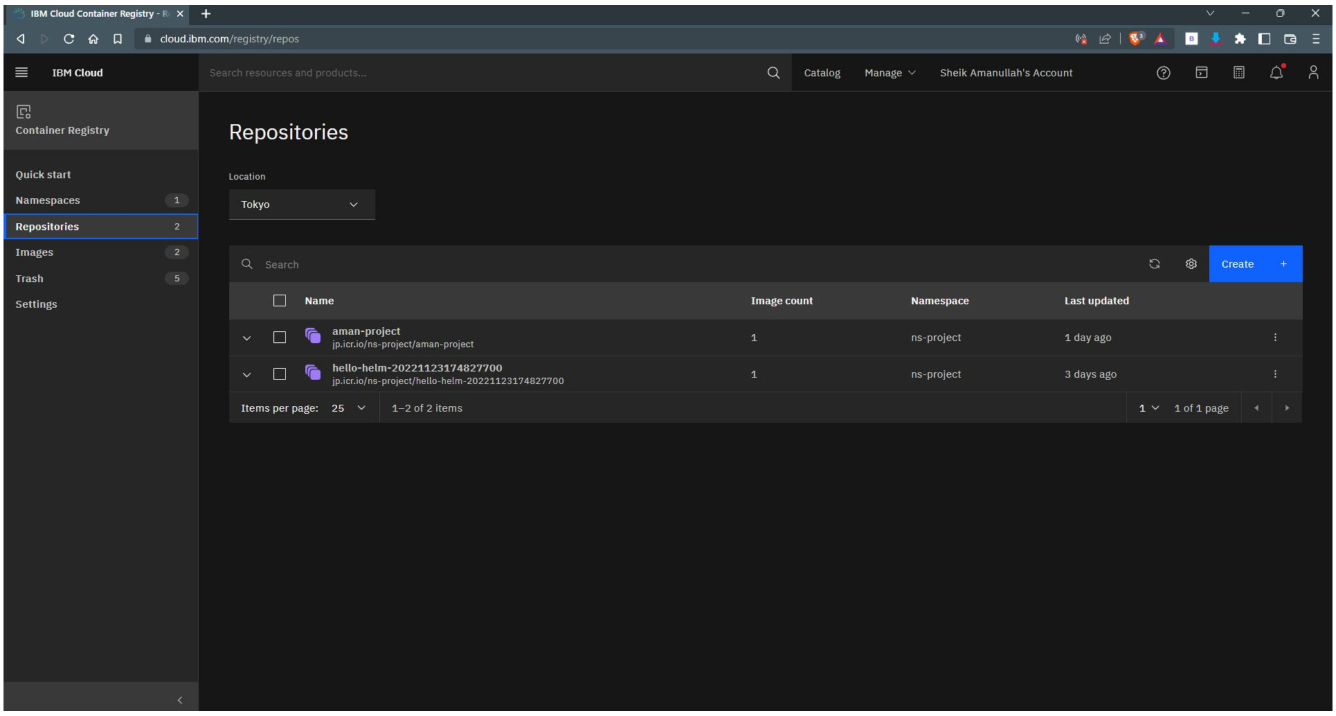


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Aman\Aman IBM\aman project final> docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/acc
ess-tokens/
PS C:\Users\Aman\Aman IBM\aman project final> docker build -t aman-project .
[+] Building 20.4s (10/11)
 => [internal] load build definition from Dockerfile                                                           0.0s
 => => transferring dockerfile: 32B                                                                            0.0s
 => [internal] load .dockerignore                                                                              0.0s
 => => transferring context: 2B                                                                                0.0s
 => [internal] load metadata for docker.io/library/python:3.10.6                                               2.2s
 => [auth] library/python:pull token for registry-1.docker.io                                                  0.0s
 => [internal] load build context                                                                              0.2s
 => => transferring context: 299.97kB                                                                          0.2s
 => [1/6] FROM docker.io/library/python:3.10.6@sha256:745efdfb7e4aac9a8422bd8c62d8bc35a693e8979a240d29677cb03e6aa  0.0s
 => CACHED [2/6] RUN apt-get update                                                                            0.0s
 => CACHED [3/6] RUN mkdir /app                                                                                0.0s
 => CACHED [4/6] WORKDIR /app                                                                                  0.0s
 => CACHED [5/6] COPY . /app                                                                                   0.0s
```



```
8d51c618126f: Waiting
9ff6e4d46744: Waiting
a89d1d47b5a1: Waiting
655ed1b7a428: Waiting
unauthorized: The login credentials are not valid, or your IBM Cloud account is not active.
PS C:\Users\Aman\Aman IBM\aman project final> ibmcloud cr login
Logging 'docker' in to 'jp.icr.io'...
Logged in to 'jp.icr.io'.

OK
PS C:\Users\Aman\Aman IBM\aman project final> docker push jp.icr.io/ns-project/aman-project
Using default tag: latest
The push refers to repository [jp.icr.io/ns-project/aman-project]
2515d6344c34: Layer already exists
72517d960819: Layer already exists
5f70bf18a086: Layer already exists
ad02e4ab118b: Layer already exists
8488b33b6249: Layer already exists
bfc1deb8136e: Layer already exists
1f123186824c: Layer already exists
3d6eb1152931: Layer already exists
100796cdf3b1: Layer already exists
54acb5a6fa0b: Layer already exists
8d51c618126f: Layer already exists
9ff6e4d46744: Layer already exists
a89d1d47b5a1: Layer already exists
655ed1b7a428: Layer already exists
denied: You have exceeded your storage quota. Delete one or more images, or review your storage quota and pricing plan.
For more information, see https://ibm.biz/BdPdFA.
PS C:\Users\Aman\Aman IBM\aman project final> 
```

Viewing the Pushed Image on Container Registry:

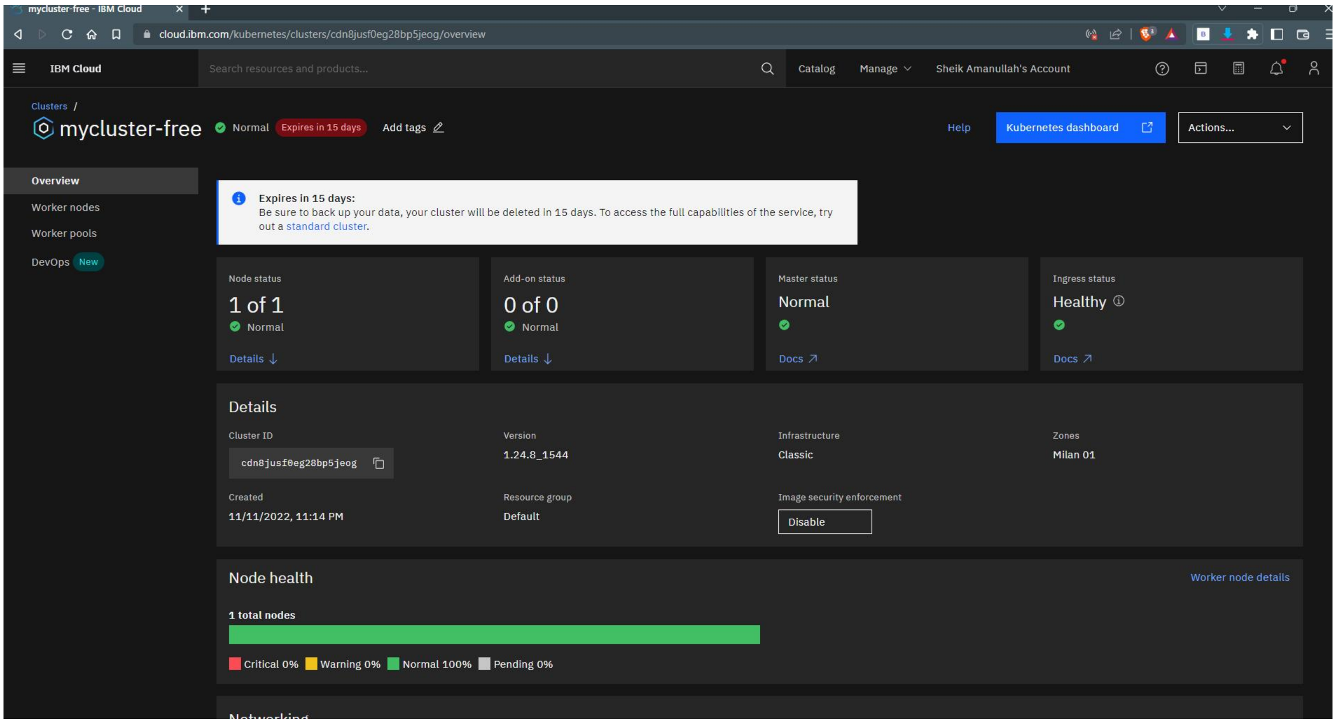


Repositories

Name	Image count	Namespace	Last updated
aman-project jp.icr.io/ns-project/aman-project	1	ns-project	1 day ago
hello-helm-20221123174827700 jp.icr.io/ns-project/hello-helm-20221123174827700	1	ns-project	3 days ago

Items per page: 25 1–2 of 2 items 1 1 of 1 page

Launch Kubernetes Service and Connect Via CLI:



mycluster-free · Normal · Expires in 15 days

Expires in 15 days:
Be sure to back up your data, your cluster will be deleted in 15 days. To access the full capabilities of the service, try out a standard cluster.

Node status: 1 of 1 Normal · Add-on status: 0 of 0 Normal · Master status: Normal · Ingress status: Healthy

Details

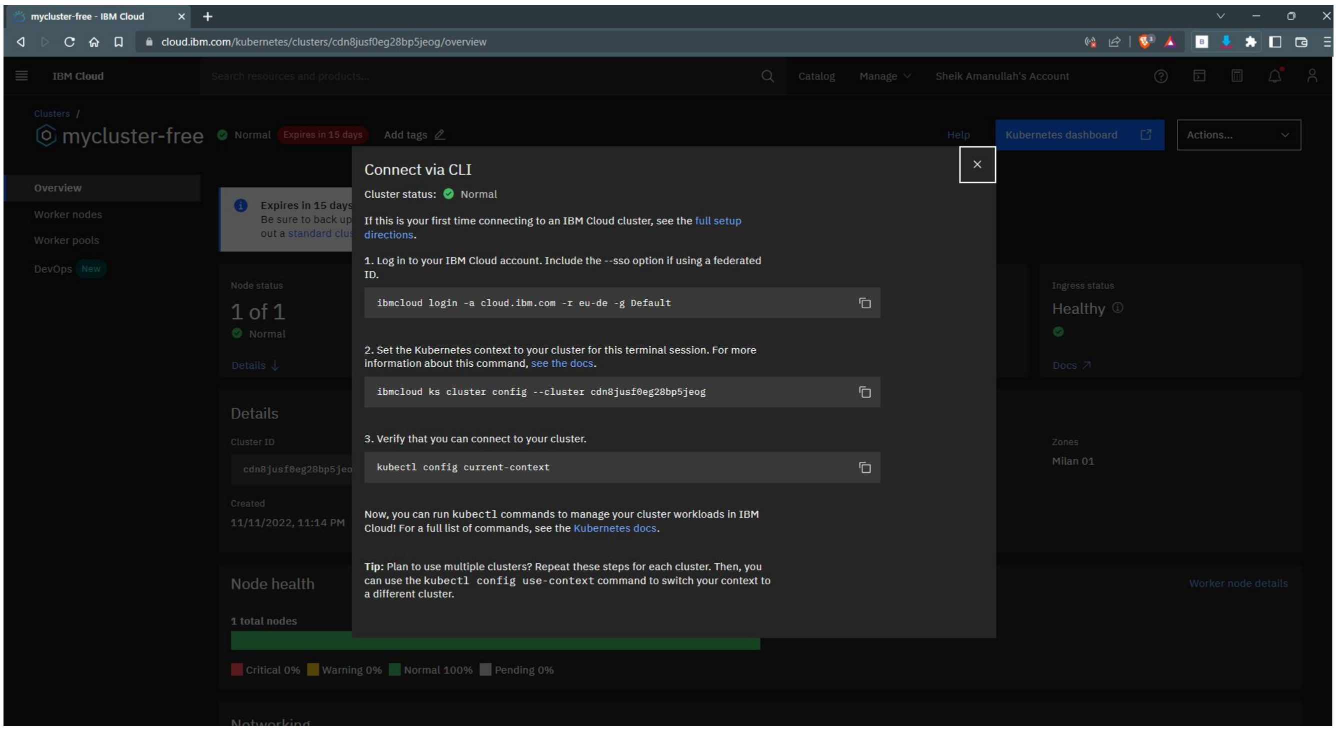
Cluster ID: cdn8jusf0eg28bp5jeog · Version: 1.24.8_1544 · Infrastructure: Classic · Zones: Milan 01

Created: 11/11/2022, 11:14 PM · Resource group: Default · Image security enforcement: Disable

Node health

1 total nodes

🟥 Critical 0% 🟨 Warning 0% 🟩 Normal 100% ⬜ Pending 0%



Connect via CLI

Cluster status: Normal

If this is your first time connecting to an IBM Cloud cluster, see the full setup directions.

1. Log in to your IBM Cloud account. Include the --sso option if using a federated ID.

```
ibmcloud login -a cloud.ibm.com -r eu-de -g Default
```

2. Set the Kubernetes context to your cluster for this terminal session. For more information about this command, see the docs.

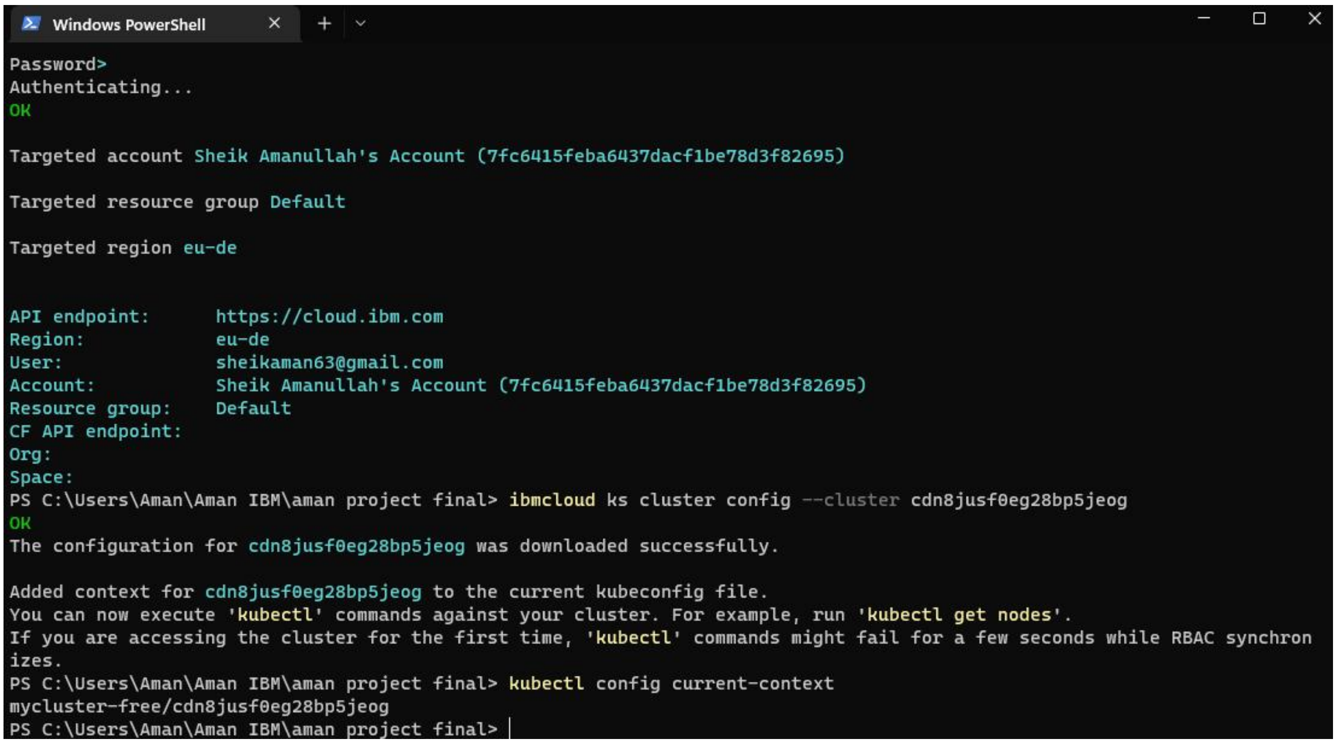
```
ibmcloud ks cluster config --cluster cdn8jusf0eg28bp5jeog
```

3. Verify that you can connect to your cluster.

```
kubectl config current-context
```

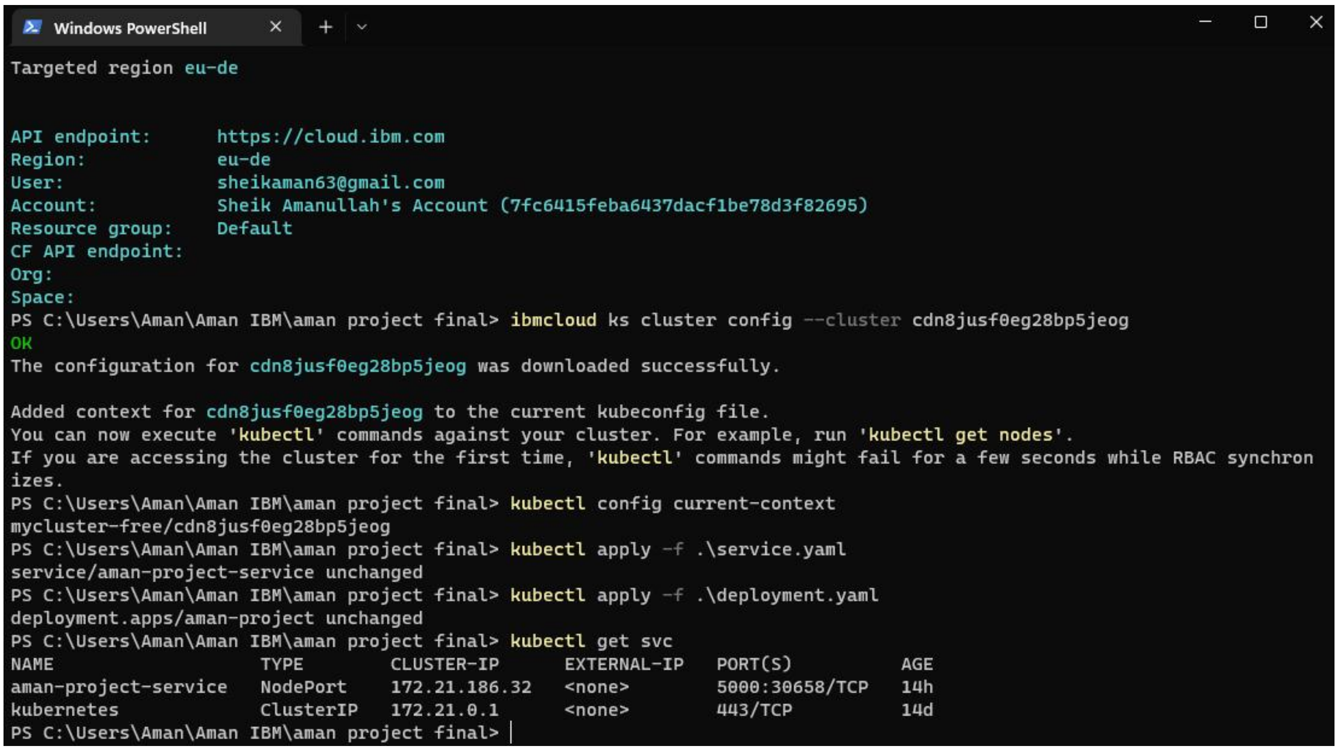
Now, you can run `kubectl` commands to manage your cluster workloads in IBM Cloud! For a full list of commands, see the Kubernetes docs.

Tip: Plan to use multiple clusters? Repeat these steps for each cluster. Then, you can use the `kubectl config use-context` command to switch your context to a different cluster.



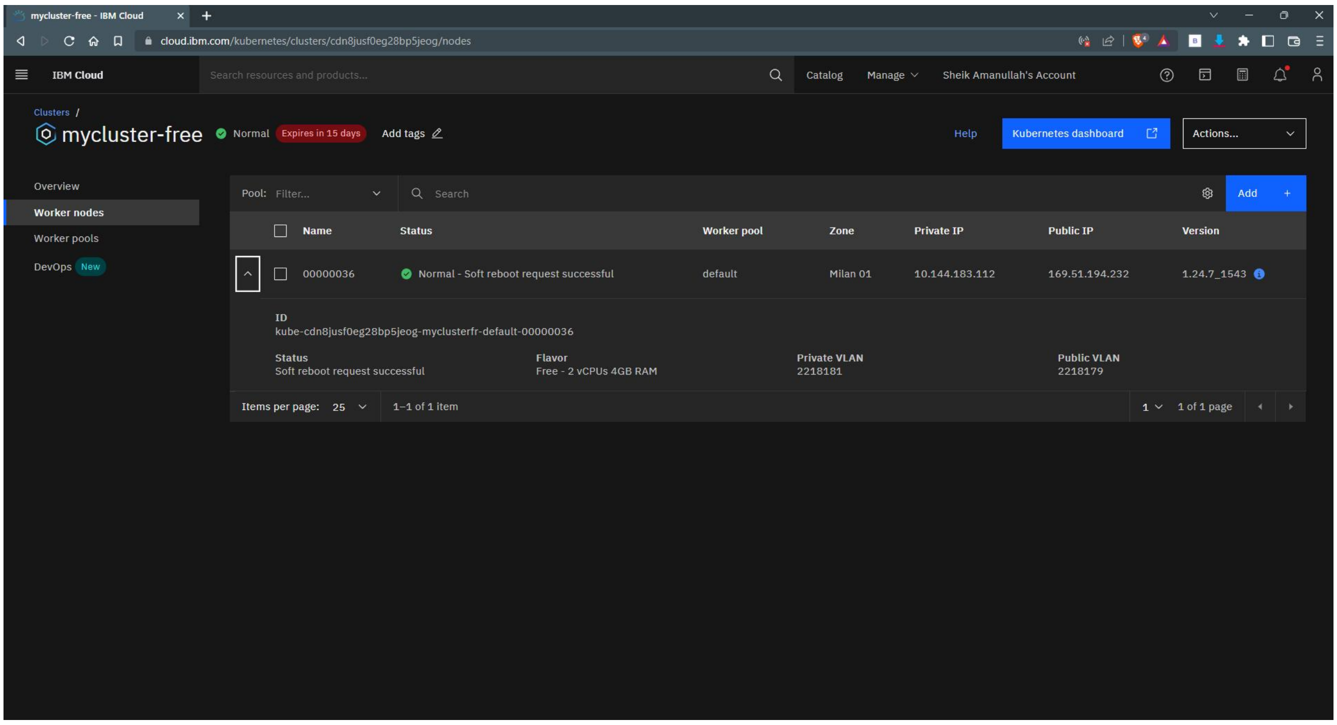
```
Password>
Authenticating...
OK

Targeted account Sheik Amanullah's Account (7fc6415feba6437dacf1be78d3f82695)

Targeted resource group Default

Targeted region eu-de


API endpoint:      https://cloud.ibm.com
Region:            eu-de
User:              sheikaman63@gmail.com
Account:           Sheik Amanullah's Account (7fc6415feba6437dacf1be78d3f82695)
Resource group:    Default
CF API endpoint:
Org:
Space:
PS C:\Users\Aman\Aman IBM\aman project final> ibmcloud ks cluster config --cluster cdn8jusf0eg28bp5jeog
OK
The configuration for cdn8jusf0eg28bp5jeog was downloaded successfully.

Added context for cdn8jusf0eg28bp5jeog to the current kubeconfig file.
You can now execute 'kubectl' commands against your cluster. For example, run 'kubectl get nodes'.
If you are accessing the cluster for the first time, 'kubectl' commands might fail for a few seconds while RBAC synchron
izes.
PS C:\Users\Aman\Aman IBM\aman project final> kubectl config current-context
mycluster-free/cdn8jusf0eg28bp5jeog
PS C:\Users\Aman\Aman IBM\aman project final>
```

Apply the yaml Files Using Command Prompt:



```
Targeted region eu-de


API endpoint:      https://cloud.ibm.com
Region:            eu-de
User:              sheikaman63@gmail.com
Account:           Sheik Amanullah's Account (7fc6415feba6437dacf1be78d3f82695)
Resource group:    Default
CF API endpoint:
Org:
Space:
PS C:\Users\Aman\Aman IBM\aman project final> ibmcloud ks cluster config --cluster cdn8jusf0eg28bp5jeog
OK
The configuration for cdn8jusf0eg28bp5jeog was downloaded successfully.

Added context for cdn8jusf0eg28bp5jeog to the current kubeconfig file.
You can now execute 'kubectl' commands against your cluster. For example, run 'kubectl get nodes'.
If you are accessing the cluster for the first time, 'kubectl' commands might fail for a few seconds while RBAC synchron
izes.
PS C:\Users\Aman\Aman IBM\aman project final> kubectl config current-context
mycluster-free/cdn8jusf0eg28bp5jeog
PS C:\Users\Aman\Aman IBM\aman project final> kubectl apply -f .\service.yaml
service/aman-project-service unchanged
PS C:\Users\Aman\Aman IBM\aman project final> kubectl apply -f .\deployment.yaml
deployment.apps/aman-project unchanged
PS C:\Users\Aman\Aman IBM\aman project final> kubectl get svc
NAME                   TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
aman-project-service   NodePort    172.21.186.32   <none>        5000:30658/TCP   14h
kubernetes             ClusterIP   172.21.0.1      <none>        443/TCP          14d
PS C:\Users\Aman\Aman IBM\aman project final>
```

View the Public IP and Port Number:


mycluster-free - IBM Cloud

cloud.ibm.com/kubernetes/clusters/cdn8jusf0eg28bp5jeog/nodes

mycluster-free Normal Expires in 15 days Add tags

Pool: Filter... Search

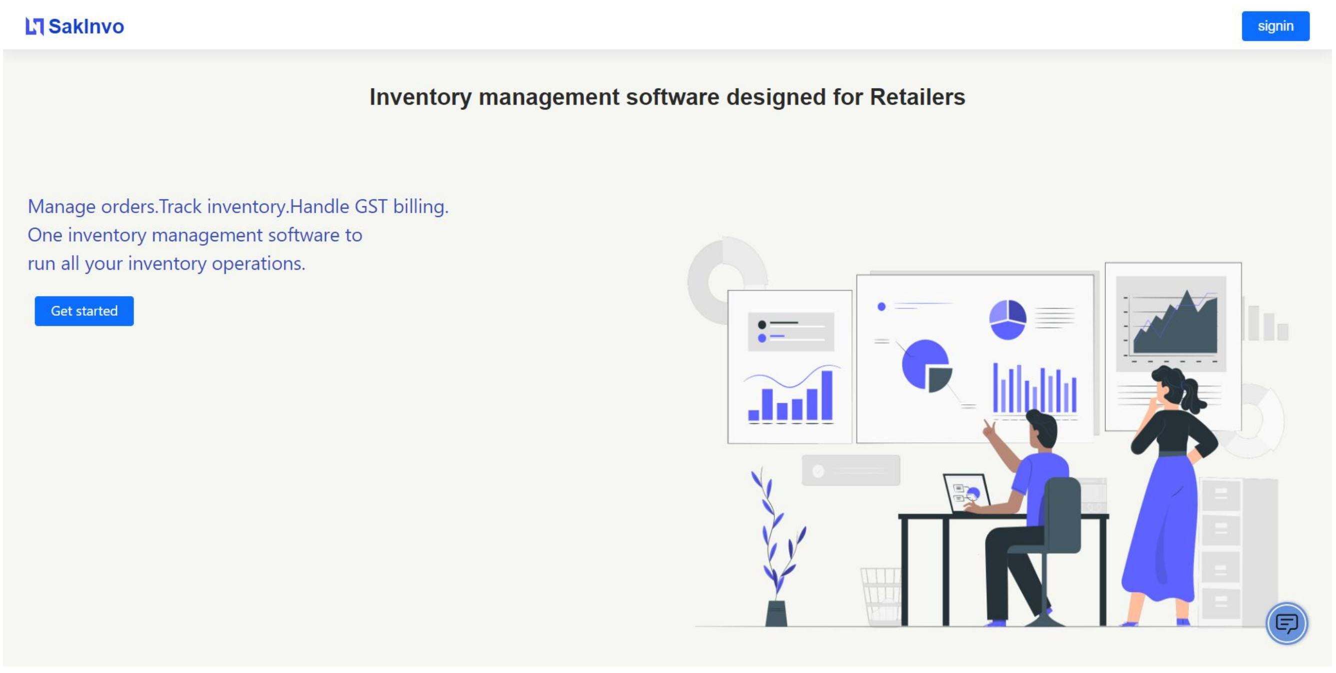
Name	Status	Worker pool	Zone	Private IP	Public IP	Version
00000036	Normal - Soft reboot request successful	default	Milan 01	10.144.183.112	169.51.194.232	1.24.7_1543

ID
kube-cdn8jusf0eg28bp5jeog-myclusterfr-default-00000036

Status	Flavor	Private VLAN	Public VLAN
Soft reboot request successful	Free - 2 vCPUs 4GB RAM	2218181	2218179

Items per page: 25 1–1 of 1 item 1 1 of 1 page

Visit the Public IP and Port Number of Project:


SakInvo signin

Inventory management software designed for Retailers

Manage orders.Track inventory.Handle GST billing.
One inventory management software to
run all your inventory operations.

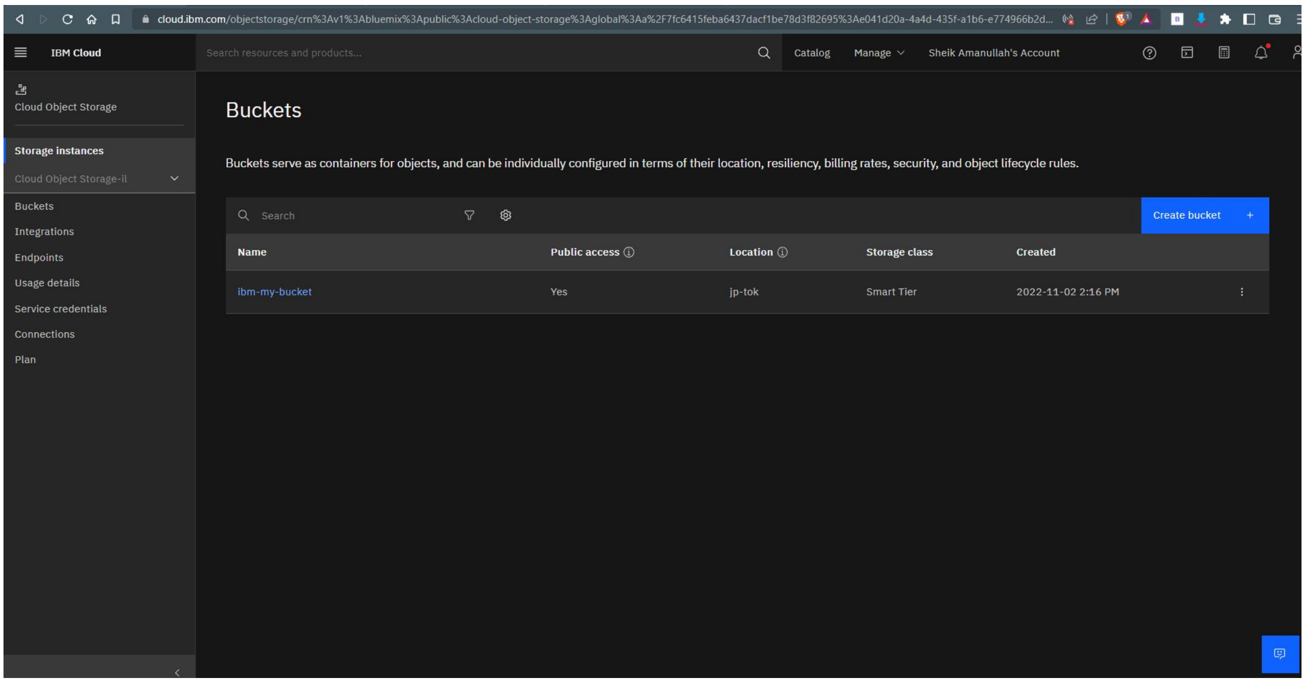
Get started

CHAPTER 7

CODING & SOLUTIONING

7.1 IBM Object Storage Service

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


cloud.ibm.com/objectstorage/crn%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F7fc6415feba6437dacf1be78d3f82695%3Ae041d20a-4a4d-435f-a1b6-e774966b2d...
IBM Cloud
Search resources and products...
Catalog
Manage
Sheik Amanullah's Account
Cloud Object Storage
Storage instances
Cloud Object Storage-il
Buckets
Integrations
Endpoints
Usage details
Service credentials
Connections
Plan
Search
Create bucket +

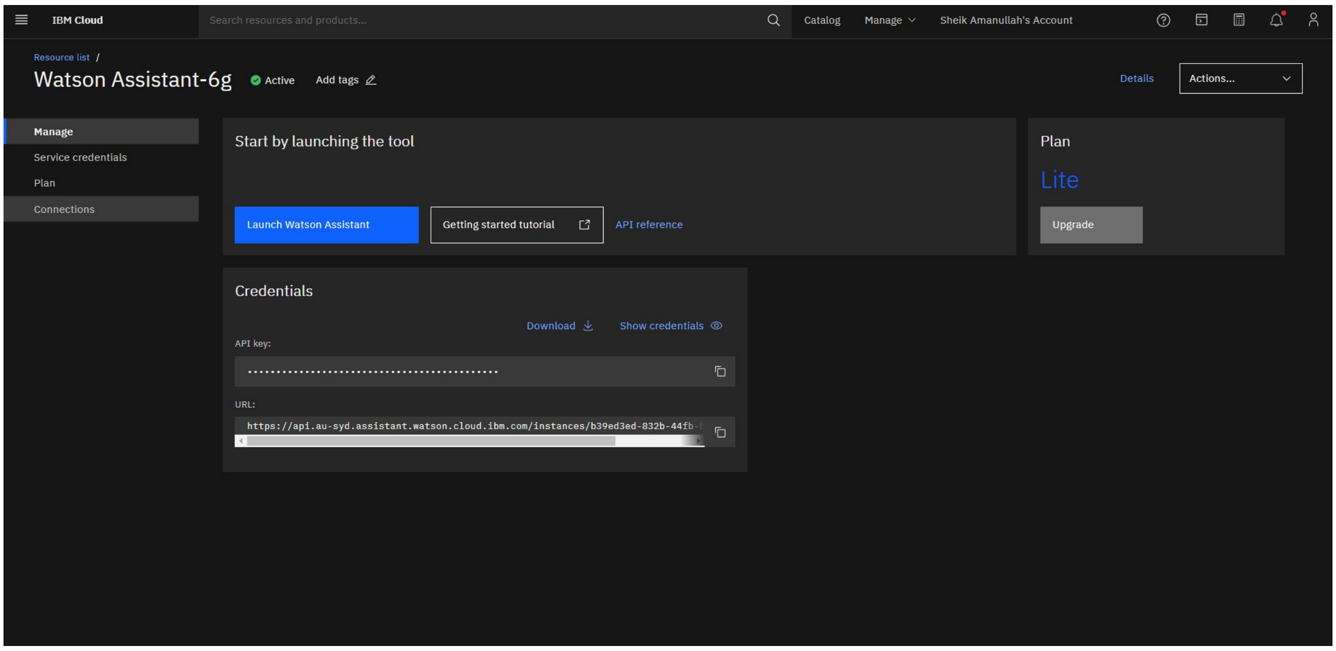
Buckets

Buckets serve as containers for objects, and can be individually configured in terms of their location, resiliency, billing rates, security, and object lifecycle rules.

Name	Public access	Location	Storage class	Created
ibm-my-bucket	Yes	jp-tok	Smart Tier	2022-11-02 2:16 PM

7.2 IBM Watson Assistant Service

IBM Watson Assistant uses artificial intelligence that understands customers in context to provide fast, consistent, and accurate answers across any application, device, or channel. Remove the frustration of long wait times, tedious searches, and unhelpful chatbots with the leader in trustworthy AI.



Watson Assistant-6g Active Add tags

Start by launching the tool

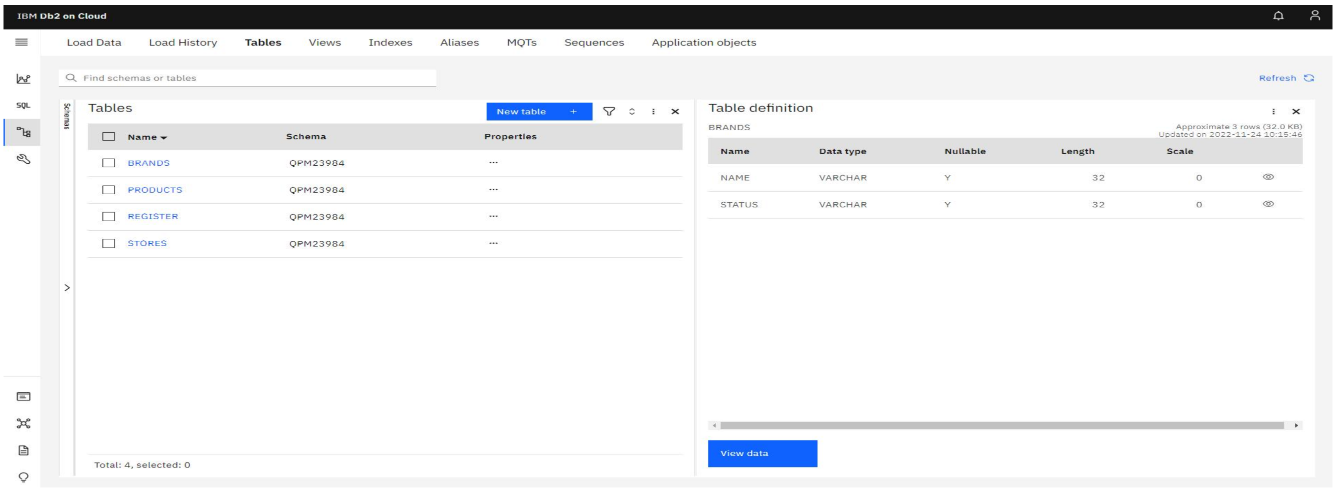
Plan
Lite

Credentials

API key:
```
••••••••••••••••••••••••••••••••••••••••••••
```
URL:
```
https://api.au-syd.assistant.watson.cloud.ibm.com/instances/b39ed3ed-832b-44fb-
```

7.3 Database Schema(IBM DB2)

IBM Db2 Database on IBM Cloud combines a proven, AI-infused, enterprise- ready data management system with an integrated data and AI platform built on the security-rich, scalable Red Hat® OpenShift® foundation. Derive insights with machine learning embedded into query processing. Cut costs with the multi-model capability that eliminates the need for data replication and migration. And enhance agility by runningDb2 on any cloud vendor.



Tables

Name	Schema	Properties
BRANDS	QPM23984	...
PRODUCTS	QPM23984	...
REGISTER	QPM23984	...
STORES	QPM23984	...

Total: 4, selected: 0

Table definition

BRANDS

Approximate 3 rows (32.0 KB)
Updated on 2022-11-24 10:15:46

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	32	0
STATUS	VARCHAR	Y	32	0

CHAPTER 8

TESTING

8.1TEST CASES

- The main objective of a test case is to ensure if different features within an application are working as expected. It helps the tester to validate if the application is free of defects and if it is working as per the expectations of the end-users.

8.2USER ACCEPTANCE TESTING

1.PURPOSE OF DOCUMENT

- The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2.DEFECT ANALYSIS

- This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

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

3.TEST CASE ANALYSIS

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9 RESULTS

Inventory management helps companies identify which and how much stock to order at what time. It tracks inventory from purchase to the sale of goods. The practice identifies and responds to trends to ensure there's always enough stock to fulfill customer orders and proper warning of a shortage.

- Easy Inventory Processing.
- Reduces Aged Inventory and Deadstock.
- Automate Manual Inventory Tasks.
- Better Customer Service.
- Lessens the Chance of Stock-Outs.

9.1 PERFORMANCE METRICS

- Inventory Performance is a measure of how effectively and efficiently inventory is used and replenished. The goal of inventory performance metrics is to compare actual on-hand dollars versus forecasted cost of goods sold.

- We've put together a list of four crucial metrics that you should keep a close eye on over the course of the year: inventory turnover, average days to sell, return on investment, and inventory carrying costs.

10 ADVANTAGES & DISADVANTAGES

ADVANTAGES FOR IMS

- Improves Accuracy. Real-time inventory tracking helps you improve inventory management and ensures that you have optimal stock available to fulfill orders.
- Reduces costs.
- Saves Time.
- Improves Business Planning.
- Improves Customer Service.

DISADVANTAGES FOR IMS

- Even with an efficient inventory management method, you can control but not eliminate business risk.

- The control of inventory is complex because of the many functions it performs. It should thus be viewed as a shared responsibility.

- Holding inventory can result to a greater risk of loss to devaluation (changes in price).

11 CONCLUSIONS

Inventory management **helps companies identify which and how much stock to order at what time**. It tracks inventory from purchase to the sale of goods. The practice identifies and responds to trends to ensure there's always enough stock to fulfill customer orders and proper warning of a shortage

12 FUTURE SCOPE

- **The Fourth Industrial Revolution will continue to drive technological change that will impact the way that we manage inventories.**

- **Successful companies will view inventory as a strategic asset, rather than an aggravating expense or an evil to be tolerated.**

- **Collaboration with supply chain partners, coupled with a holistic approach to supply chain management, will be key to effective inventory management.**

- **The nature of globalization will change, impacting inventory deployment**

decisions dramatically.

- **Increased focus on supply chain security, and concerns about the quality of inventory itself, will be primary motivators to changing supply chain and inventory strategy.**

13. APPENDIX

Public IP Deployment on IBM Kubernetes:

http://169.51.194.232:30658/

PROJECT DEVELOPMENT PHASE LINK

https://github.com/IBM-EPBL/IBM-Project-50861-1660927231/tree/main/Project%20Development%20Phase

DEMO VIDEO DOWNLOAD LINK

https://youtu.be/sljjfILm-9w

SOURCE CODE:

APP.PY

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


conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31498;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=qpm23984;PWD=4WHCiEebOv9DkvHF","","")

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

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

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


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

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

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


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

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

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

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

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

@app.route('/addproduct',methods = ['POST', 'GET'])
def addproduct():
    if request.method == 'POST':
     name = request.form['product-name']
     quantity = request.form['quantity']
     price= request.form['price']
     insert_sql = "INSERT INTO products VALUES (?,?,?)"
     prep_stmt = ibm_db.prepare(conn, insert_sql)
     ibm_db.bind_param(prep_stmt, 1, name)
     ibm_db.bind_param(prep_stmt, 2, quantity)
     ibm_db.bind_param(prep_stmt, 3, price)
     ibm_db.execute(prep_stmt)

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


@app.route('/dashboard')
def dashboard():

     sql = "SELECT COUNT(*) FROM stores"
     stmt = ibm_db.exec_immediate(conn, sql)
     tuple = ibm_db.fetch_both(stmt)
     number=tuple['1']
     sql = "SELECT COUNT(*) FROM products"
     stmt = ibm_db.exec_immediate(conn, sql)
     prod = ibm_db.fetch_both(stmt)
     numprod=prod['1']
     sql = "SELECT COUNT(*) FROM brands"
     stmt = ibm_db.exec_immediate(conn, sql)
     brand = ibm_db.fetch_both(stmt)
     numbrand=brand['1']

     if tuple | prod | brand :
      return render_template('dashboard.html', numstore= number,numbrand = numbrand,numprod =
numprod )

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

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

@app.route('/signin',methods =['GET', 'POST'])
def signin():
    global userid
    msg = ''
    if request.method == 'POST' :
       un = request.form['username']
```

```
        pd = request.form['password']
        sql = "SELECT * FROM register WHERE username =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,un)
        ibm_db.bind_param(stmt,2,pd)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            session['loggedin'] = True
            session['id'] = account['USERNAME']
            userid=  account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'

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


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

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

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

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

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

DOCKER FILE

```
FROM python:3.10.6
LABEL maintainer="Sheik Amanullah, sheikaman63@gmail.com"
RUN apt-get update
RUN mkdir /app
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
EXPOSE 5000
ENTRYPOINT [ "python" ]
CMD [ "app.py", "--host=0.0.0.0"]
```

YAML FILES

SERVICE.YAML

```
apiVersion: v1
kind: Service
metadata:
  name: aman-project-service
spec:
  type: NodePort
  ports:
    - port: 5000
  selector:
    app: aman-project
```

DEPLOYMENT.YAML

```
apiVersion: apps/v1

kind: Deployment

metadata:
    name: aman-project
    labels:
        app: aman-project

spec:
```

```
selector:
   matchLabels:
      app: aman-project
replicas: 1

template:
   metadata:
      labels:
         app: aman-project

   spec:
      containers:
         - name: aman-project

           image: jp.icr.io/ns-project/aman-project

           imagePullPolicy: Always

           ports:
              - containerPort: 5000
           env:
              - name: DISABLE_WEB_APP
                value: "false"
```