

Inventory Management System For Retailers

INTRODUCTION

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.

Once sold, inventory becomes revenue. Before it sells, inventory (although reported as an asset on the balance sheet) ties up cash. Therefore, too much stock costs money and reduces cash flow.

One measurement of good inventory management is inventory turnover. An accounting measurement, inventory turnover reflects how often stock is sold in a period. A business does not want more stock than sales. Poor inventory turnover can lead to deadstock, or unsold stock.

1.1 Project Overview:

Lean Inventory Adopting lean inventory for your warehouse is just as important as it is in manufacturing. The basic premise of lean is that you only have what you need and nothing more. Possibly reduce or eliminate safety stocks, and try to get suppliers to deliver smaller quantities more frequently. That improves the good relationship between retailers and manufacturing companies. Because they can also see what the customer needs and what products we are sending out it helps retailers to know what products are in demand. so there is no wastage of space.

1.2 Purpose:

The customers are highly satisfied because of the availability of stock. When we buy frequently, it means that the product is freshly manufactured , so quality will be higher. When quality is higher, the customer will be easily satisfied.

LITERATURE SURVEY

2.1 Existing problem

Primaseller is one of the promising names in the category of retail management software. The software provides a flawless interface with incredible features. One can very easily sell the product by fetching the customers from various marketplaces like Amazon, eBay, Flipkart & many more.

A very handy feature of this software is in-store POS billing which may help you manage your physical store sales. Further, one can very easily synchronize inventory. Whether offline or online, you can always have access to your inventory irrespective of location.

The option to create digital catalogs for your products adds extra essence to the product. The user-friendly interface and responsiveness of the site add to the comfortability of the user. Also, the data is secure and safe with SSL security.

2.1 References:

“Demand uncertainty and inventory turnover performance: An empirical analysis of the US retail industry”

The purpose of this study is to investigate the impact of demand uncertainty on inventory turnover performance through empirical modeling. In particular we use the inaccuracy of quarterly sales forecasts as a proxy for demand uncertainty and study its impact on firm level inventory turnover ratios. Design/methodology/approach We use regression analysis to study the effect of various measures on inventory performance. We use a sample financial data for 304 publicly listed U.S. retail firms for the 25-year period from 1985 to 2009. Findings Controlling for the effects of retail segments and year, it is found that inventory turnover is negatively correlated with mean absolute percentage error of quarterly sales forecasts and gross margin and positively correlated with capital intensity and sales surprise. These four variables explain 73.7% of the variation across firms and over time and 93.4% of the within-firm variation in our data. Practical implications In addition to conducting an empirical investigation for the sources of variation in a major operational metric, the results in this study can also be used to benchmark a retailer's inventory

performance against its competitors. Originality/value We develop a new proxy to measure the demand uncertainty that a firm faces and show that this measure may help to explain the variation in inventory performance.

“The analysis of the impact of business process outsourcing on the profitability of commercial airline in the conditions of a global pandemic”

This article provides an economic assessment of the impact of the global pandemic COVID-19 on the economic efficiency of commercial airlines. The dominant role of airlines in the formation of flexible service supply chain and service travel chain has been identified, which increases their customer orientation and competitiveness in the air transportation market, as well as allows them to adapt more quickly to the changing logistics environment. It has been proven that the use of the outsourcing mechanism in combination with the diversification of services provided, allows, on the one hand, to create added consumer value for customers, and on the other - necessitates building complex integration relationships with business partners in service supply chains. Analysis of statistics and experience of leading airlines with different business models in the air transportation market has shown that outsourcing business processes in a global pandemic has allowed carriers to optimize costs according to the volume of work, respond flexibly to changes in consumer demand and better overcome negative impacts external logistics environment.

“Making Better Fulfillment Decisions on the Fly in an Online Retail Environment”

Relative to brick-and-mortar retailers, online retailers have the potential to offer more options to their customers, with respect to both inventory as well as delivery times. To do this entails the management of a distribution network with more decision options than a traditional retailer. The online retailer, not the customer, decides from where items will ship, by what shipping method, and how or whether multiple-item orders will be broken up into multiple shipments. What is the best way to fulfill each customer's order to minimize average outbound shipping cost? We partner with an online retailer to examine this

question. We develop a heuristic that makes fulfillment decisions by minimizing the immediate outbound shipping cost plus an estimate of future expected outbound shipping costs. These estimates are derived from the dual values of a transportation linear program (LP). In our experiments on industry data, we capture 36% of the opportunity gap assuming clairvoyance, leading to reductions in outbound shipping costs on the order of 1%. These cost savings are achieved without any deterioration in customer service levels or any increase in holding costs. The transportation LP also serves as the basis for a metric that provides information on the quality of the inventory position. Based on initial successful piloting, our industrial partner has implemented the metric as well as a version of the heuristic that it is applying to every fulfillment decision for each of its stock keeping units in North America.

“Inventory management for retail companies: A literature review and current trends”

In recent years, the correct management of inventories has become a fundamental pillar for achieving success in enterprises. Unfortunately, studies suggesting the investment and adoption of advanced inventory management and control systems are not easy to find. In this context, this article aims to analyze and present an extensive literature concerning inventory management, containing multiple definitions and fundamental concepts for the retail sector. A systematic literature review was carried out to determine the main trends and indicators of inventory management in Small and Medium-sized Enterprises (SMEs). This research covers five years, between 2015 and 2019, focusing specifically on the retail sector. The primary outcomes of this study are the leading inventory management systems and models, the Key Performance Indicators (KPIs) for their correct management, and the benefits and challenges for choosing or adopting an efficient inventory control and management system. Findings indicate that SMEs do not invest resources in sophisticated systems; instead, a simple Enterprise Resource Planning (ERP) system or even programs such as Excel or manual inventories are mainly used.

“Integrated Location- Inventory Modelling under Forward and Reverse Product Flows in the Used Merchandise Retail Sector: A Multi-Echelon Formulation”

This study presents a joint three-echelon location inventory model for a donation-demand driven industry in which the main warehouse (MW), distribution centers (DC), retail stores (RS) and donation-only centers (ADCs) exist. This unique inventory-location problem involves demand and supply uncertainties, coverage radius limitations, service level requirements, and multiple products consideration. Each retailer has two classes of products flowing from the assigned DC due to demands minus donations occurring in that retailer. The proposed model simultaneously determines the number of DCs to open, DC locations, and assignments of retailers to the open DCs for particular product types. The objective is to minimize the total annual cost including: facility location costs, transportation costs, inventory costs, and the lost sale costs. Due to the complexity of the problem, the proposed model structure allows for relaxing complicating constraints through recourse to Lagrangian relaxation. The use of robust branch-cut and price heuristics solves the mixed integer nonlinear problem to obtain a lower bound and a distance-based heuristic to get an upper bound. We formulate essential features of this novel problem, solve several numerical example problems and evaluate solution performance. We believe this is a novel problem environment, and that this initial study extends integrated location-inventory modeling to a new context.

Author:

- 1) G. Haçerlioğulları, A. Şen, y E. A. Aktunç, “Demand uncertainty and inventory turnover performance: an empirical analysis of the US retail industry”, *International Journal of Physical Distribution and Logistics Management*, vol. 46, número. 6–7, pp. 681–708, 2016, doi: 10.1108/IJPDLM-12-2014-0303
- 2) Y. Wang, S. W. Wallace, B. Shen, y T.-M. Choi, “Service supply chain management: A review of operational models”, *European Journal of Operational Research*, vol. 247, núm. 3, pp. 685–698, 2015.
- 3) S. Mahar y P. D. Wright, “The value of postponing online fulfillment decisions in multi-channel retail/e-tail organizations”, *Computers & operations research*, vol. 36, núm. 11, pp. 3061–3072, 2009.
- 4) M. Barratt, T. J. Kull, y A. C. Sodero, “Inventory record inaccuracy dynamics and the role of employees within multi-channel distribution center inventory systems”, *Journal of Operations Management*, vol. 63, núm. 1, pp. 6–24, nov. 2018, doi: 10.1016/j.jom.2018.09.003.
- 5) A. Ross, M. Khajehnezhad, W. Otieno, y O. Aydas , “Integrated location-inventory modelling under forward and reverse product flows in the used merchandise retail sector: A multi-echelon formulation”, *European Journal of Operational Research*, vol. 259, núm. 2, pp. 664–676, 2017, doi: 10.1016/j.ejor.2016.10.036.

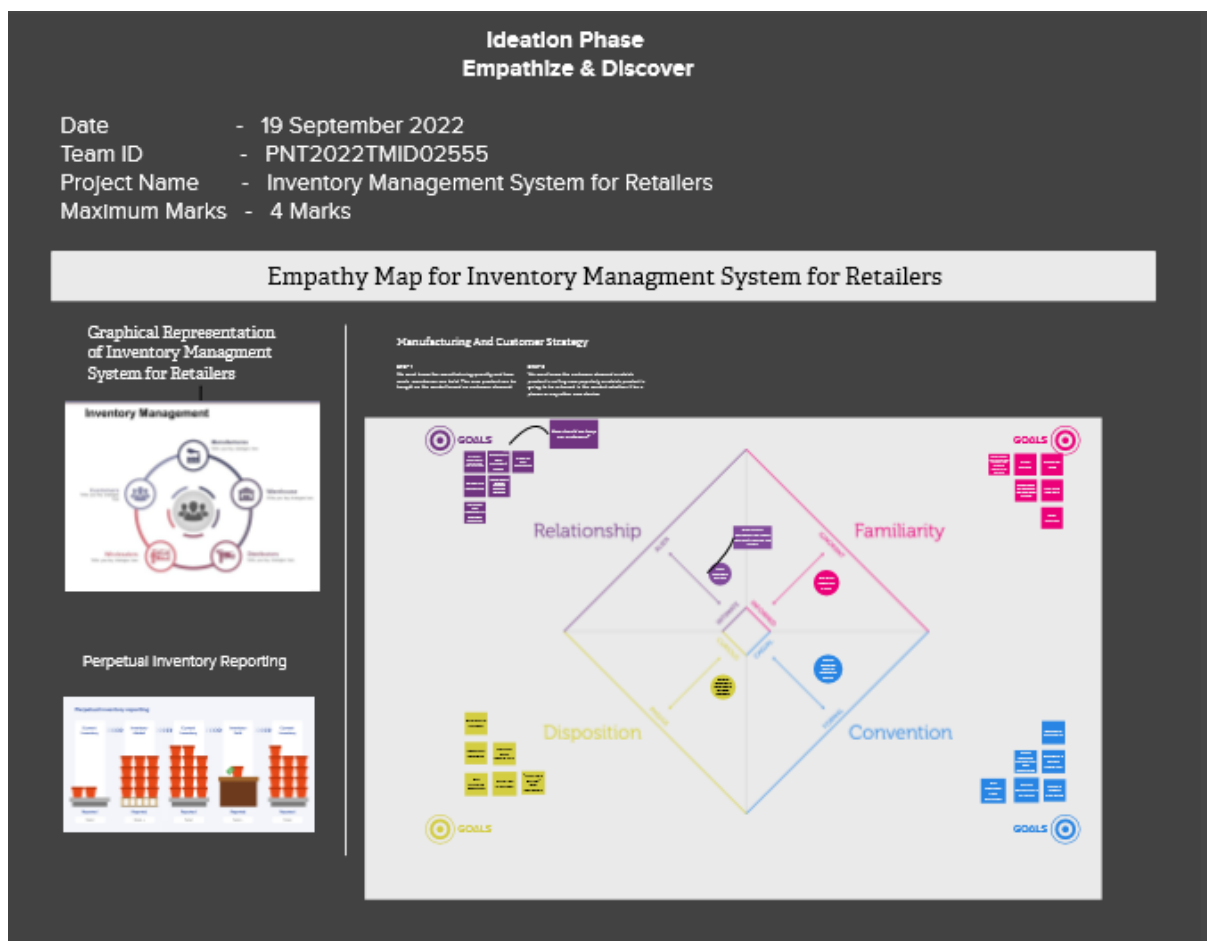
2.3 Problem Statement Definition

Inefficient Warehouse Management and Overselling.

IDEATION & PROPOSED SOLUTION

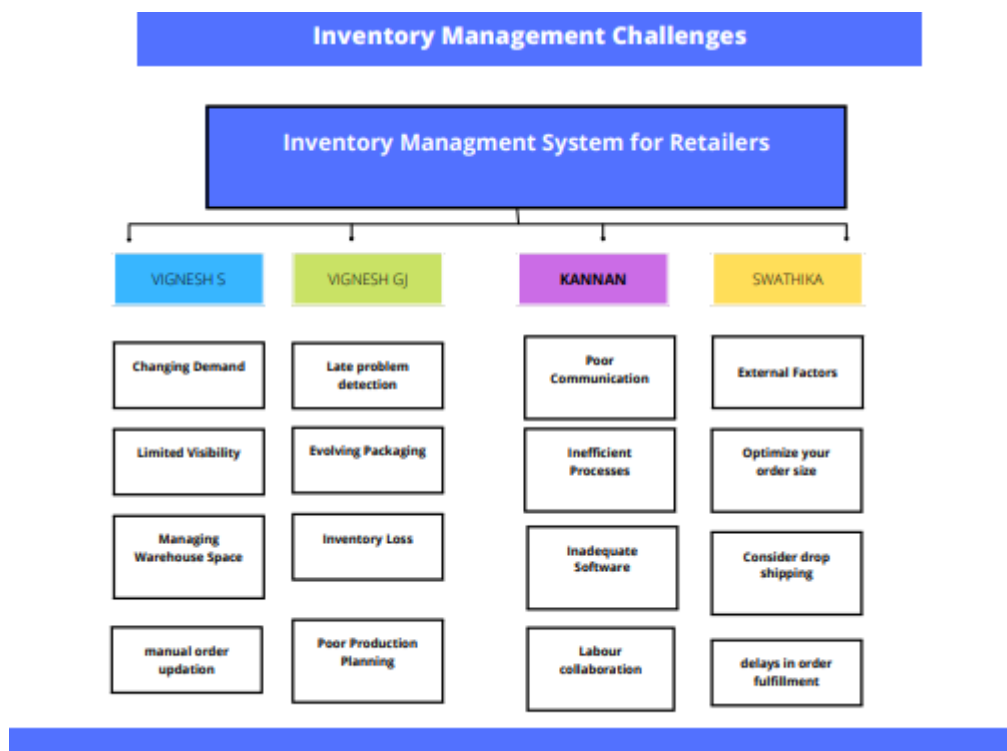
3.1 Empathy Map Canvas

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



3.2 Ideation & Brainstorming

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



3.3 Proposed Solution

Lean Inventory Adopting lean inventory for your warehouse is just as important as it is in manufacturing. The basic premise of lean is that you only have what you need and nothing more. Possibly reduce or eliminate safety stocks, and try to get suppliers to deliver smaller quantities more frequently. That improves the good relationship between retailers and manufacturing companies. Because they can also see what the customer needs and what products we are sending out it helps retailers to know what products are in demand. so there is no wastage of space.

3.4 Problem Solution fit

Frequent buying is a new thing, but it helps a lot in warehouse spacing , stock analysis, and demand. Retailers can easily change to people's demands to make a sale faster and maintain a good relationship with manufacturing companies.

The customers are highly satisfied because of the availability of stock. When we buy frequently, it means that the product is freshly manufactured , so quality will be higher. When quality is higher, the customer will be easily satisfied.

Frequently having a trade will reduce stock wastage, increase space in a warehouse, and improve data analysis. So retailers can easily find the demand of people so they can stay updated on people's demands and make more profit because they avoid buying unwanted products.

REQUIREMENT ANALYSIS

4.1 Functional requirement

Advancements in computing, applications, databases and IT operations created a market for modern inventory management systems. The shift to web services, application-based development and APIs enabled third-party application integrations that weren't possible before. Now, inventory management systems can manage a host of essential, inventory-related business functions.

Consider the following modern inventory management system features, which you can tailor to your business needs and do more than control inventory:

Cloud Infrastructure: Cloud-based software is a scalable, cost-effective solution. Cloud computing means organizations don't need to hire dedicated staff to manage and maintain those systems. Cloud infrastructure also enables automated backups, secure access and real-time collaboration across multiple locations.

Internet of Things (IoT) Integrations: The IoT movement brought automated inventory controls and reporting, digital labeling, GPS tracking and Bluetooth- and RFID-enabled tracking. The technology has also been used to improve security against theft and loss.

Digital Labeling: The move to digital labeling created features like GPS-enabled RFID tags and digital barcoding. With digital inventory data, organizations can now network handheld devices with databases for real-time tracking and paperless documentation.

Mobile Systems: Mobile applications running on wireless devices communicate with back-office systems and central databases to speed up multi-location inventory processes. Using mobile devices (running iOS or Android) allows for real-time tracking and instant inventory alerts and notifications.

Machine Learning and AI: AI-empowered virtual agents (or chatbots) are changing how businesses share inventory availability to ecommerce customers. They use machine learning algorithms and web-based services to communicate out-of-stock inventory, make recommendations based on an

active shopping cart or user history and automate backorder fulfillment. Machine learning can also report sales data trends that are critical to optimizing online customer engagement metrics. Other benefits of machine learning include the ability to improve internal inventory controls and prevent shortages. The technology can also improve forecasting with real-time data analysis and report on inventory trends that are difficult to track and predict manually.

4.2 Non-Functional requirements

Inventory management systems for manufacturing have hybrid features for material-based inventory and warehouse management. These features incorporate advanced unit conversion capabilities to handle bulk material stock, incorporate lean manufacturing methodologies, material requirements planning and bill of material invoicing.

Top inventory management system features for manufacturing include:

Unit-of-Measure Conversion: Convert inventory quantities received to a higher or lower unit of measure.

Automated Ordering: Set stock limits and place orders when inventory drops below a given quantity.

Raw Materials Planning: Track direct or indirect raw materials by purchasing and production cycles, create a bill of materials and determine the return on investment (ROI) for a given raw material.

Purchase Order Management: Assign suppliers, track product lead times and manage purchase order lifecycles.

Serialized Inventory: Drill down to individual items instead of SKUs to trace a product lifecycle, identify defective inventory and track warranty information.

Wholesale Distribution

Wholesalers with online distribution channels need inventory management features that help control and trace bulk inventory, manage multiple suppliers and customers and automate processes.

Top features for wholesale distributors include:

Sales Order Fulfillment: Use a single dashboard to view sales orders, create packing slips and automate email confirmations to customers.

Multiple Warehouse Fulfillment: Track and fulfill orders with inventory stored across multiple warehouses.

Kitting: Control, transfer and connect items normally tracked individually to a bundled sales order for offerings like a subscription box.

Lot Tracking: Track multiple quantities of perishable, time-sensitive items by manufacturer expiration date with traceable lot numbers.

Bin Management: Bin management allows for inventory to be received at the warehouse and put-away into preferred bins within the rack system of the warehouse for easy picking later during the order fulfillment process

Retail

Retailers depend on inventory management features that optimize the in-store customer experience. They need solutions that improve inventory tracking, demand forecasting and inventory alerts. Accurate, multi-location management features are essential for multi-channel operations. Point-of-sale and accounting integration features are crucial for profitable brick-and-mortar retail operations. Read our guide on retail inventory management for tips and best practices.

Top features for retail include:

Partial Receiving: Control and track partially received inventory.

Backorder Management: Create new purchase orders with consolidated sales orders by supplier and track products by type, estimated arrival or season.

Demand Forecasting: Set preferred stock levels by seasonal demand, automate multi-location replenishment and generate reports to identify future demand trends.

Cycle Counting: Categorize inventory by volume, value or transactions and receive automated prompts for inventory counts to establish safety stock levels. Read our [cycle counting guide](#) to learn more about this process.

Ecommerce

Ecommerce inventory systems need to support accurate inventory controls and automated processes that deliver products to customers without interruption. Accounting integrations that incorporate online payment processing, email invoicing and inventory tracking are popular ecommerce features.

Top inventory management system features for ecommerce include:

Returns Handling: Send customer email updates, assign warehouses to receive a returned item and generate reports on return trends.

Automatic Restocking: Generate automated purchase orders when stock goes below a pre-set level.

Multi-Carrier Shipping: Store contact information and data on preferred shipping carriers and manage multiple logistics providers.

Virtual Stock Control: Create virtual inventory and manage drop-ship logistics with supplier inventory stored elsewhere.

Display inventory availability on website: Accurately reflect the availability of stock on ecommerce platforms.

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey	In this activity you are expected to gather/collect the relevant information on project use case, refer the existing solutions, technical papers, research publications etc	3 SEP 2022
Prepare Empathy Map	In this activity you are expected to prepare the empathy map canvas to capture the user Pains & Gains, Prepare list of problem statements.	10 SEP 2022
Ideation	In this activity you are expected to list the ideas (at least 4 per each team member) by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	17 SEP 2022

Project Design Phase -I

TITLE	DESCRIPTION	DATE
Proposed Solution	In this activity you are expected to prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	20 SEP 2022
Problem- Solution Fit	Prepare problem - solution fit document & Solution Architecture	24 SEP 2022
Solution Architecture	In this activity you are expected to prepare solution architecture document	1 OCT 2022

Project Design Phase –II

TITLE	DESCRIPTION	DATE
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	3 OCT 2022
Functional Requirement	to prepare the functional requirement document.	10 OCT 2022
Data Flow Daigrams	Prepare the Functional Requirement Document & DataFlow Daigrams	15 OCT 2022
Technology Architecture	Prepare Technology Architecture of the solution	18 OCT 2022

Project Planning Phase

TITLE	DESCRIPTION	DATE
Prepare Milestone & Activity List	Prepare Milestone & Activity List	22 OCT 2022
Sprint delivery plan	, Sprint Delivery Plan	26 OCT 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA
Sprint-1		USN-3	As a user, I can register for the application through Facebook	2	Low	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA
Sprint -1	Dashboard	USN-6	As a user, I will go through the dashboard	2	Low	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA
Sprint-2	Create UI to interact with the application <ul style="list-style-type: none"> Registration Page Login Page Display Items in the Dashboard Adding Items Test it 	USN-7	As a user, I will test the UI interact	2	High	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA
Sprint-3	Integrating SendGrid service	USN-8	Integrating SendGrid service to the application. Follow the documentation provided in the reference link.	2	High	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA
Sprint-4	Deployment of app in The IBM cloud.	USN-9	Containerize a Flask application by using Docker and deploy it to the IBM Cloud Kubernetes Service	2	High	VIGNESH S VIGNESH GJ KANNAN T SWATHIKA

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	8	5 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

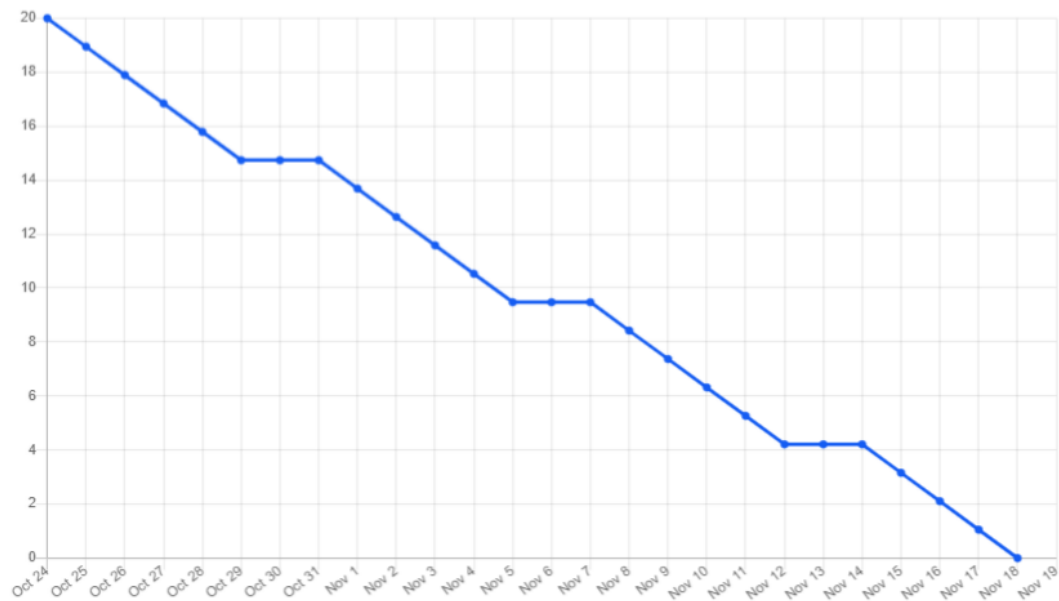
$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

$$AV = (10+8+6+7)/24 = 31/24$$

$$= 1.289$$

6.3 Reports from JIRA

Burndown Chart



CODING & SOLUTIONING

7.1 Feature 1

Frequent buying is a new thing, but it helps a lot in warehouse spacing , stock analysis, and demand. Retailers can easily change to people's demands to make a sale faster and maintain a good relationship with manufacturing companies

The customers are highly satisfied because of the availability of stock. When we buy frequently, it means that the product is freshly manufactured , so quality will be higher. When quality is higher, the customer will be easily satisfied.

7.2 Feature 2

Frequently having a trade will reduce stock wastage, increase space in a warehouse, and improve data analysis. So retailers can easily find the demand of people so they can stay updated on people's demands and make more profit because they avoid buying unwanted products.

Overall, this idea's scalability will be rated as a 6 because adopting it will take more time and it will increase memory usage more when compared to the previous one. For frequent trading, more data is wanted to be stored accurately, like stock in and stock out.

TESTING

8.1 Test Cases

Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup displayed or not	Inventory_management localhost	Login/Signup popup should display	Working as expected	Pass				
Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery	Inventory_management localhost	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery	Working as expected	Partial			BUG-1235	VIGNESH S
Home page	Verify user is able to log into application with Valid credentials		1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	username:vigneshramani27@gmail.com password:vignesh	User should navigate to user account homepage	Working as expected	pass				
Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button		Application should show 'Incorrect email or password' validation message.	Working as expected	Partial	need to verify		bug-1236	VIGNESH GJ
Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	username:vigneshramani27@gmail.com password:vignesh	Application should show 'Incorrect email or password' validation message.	Working as expected	pass				
Login page	Verify user is able to log into application with Invalid credentials		1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	username:vigneshramani27@gmail.com password:vignesh	Application should show 'Incorrect email or password' validation message.						

8.2 User 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	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	8	4	2	4	18
Duplicate	1	0	3	0	4
External	5	2	0	1	8
Fixed	9	3	3	19	34
Not Reproduced	0	0	2	0	2
Skipped	0	0	1	2	3
Won't Fix	2	5	2	1	10
Totals	24	14	14	26	79

3. TestCaseAnalysis

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

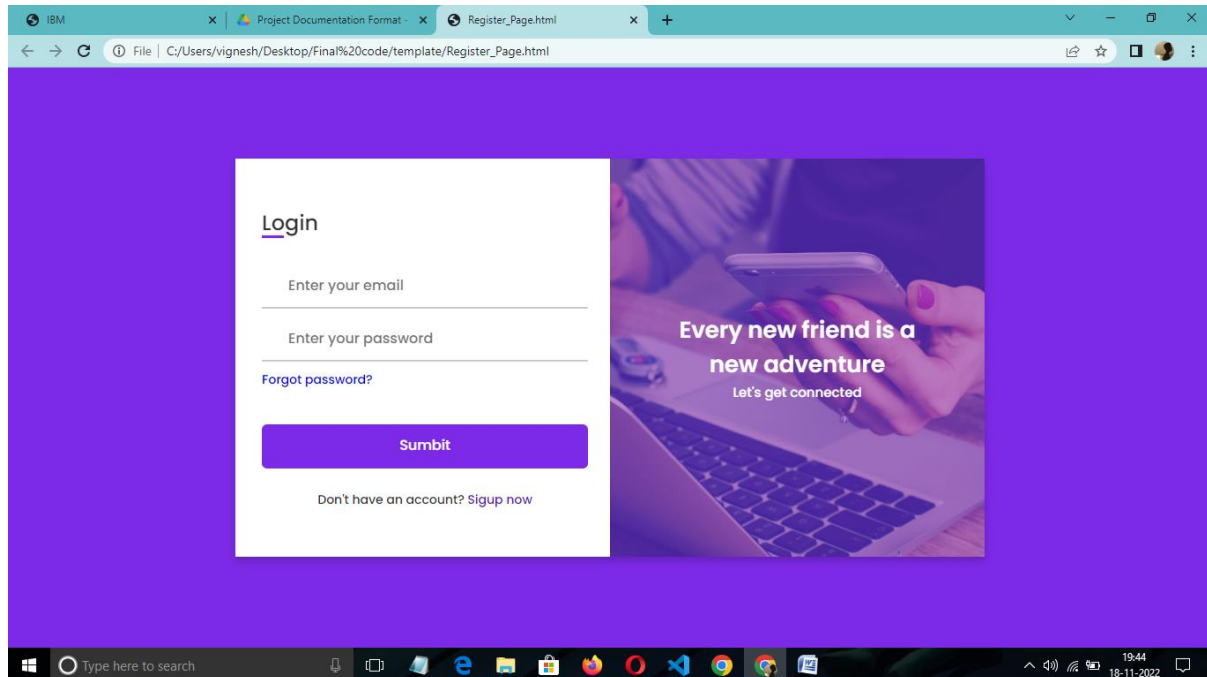
Section	TotalCases	Not Tested	Fail	Pass
PrintEngine	6	0	0	6
ClientApplication	51	0	0	51
Security	2	0	0	2
OutsourceShipping	3	0	0	3

ExceptionReporting	9	0	0	9
FinalReportOutput	4	0	0	4
VersionControl	2	0	0	2

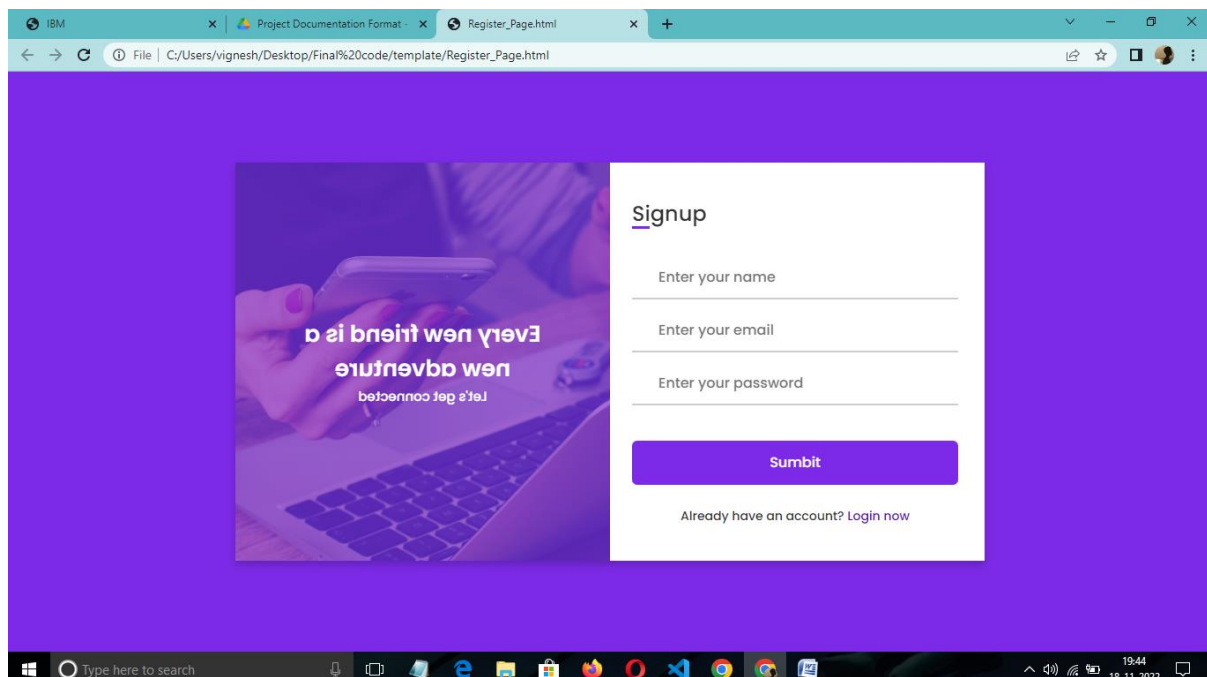
RESULTS

9.1 Performance Metrics

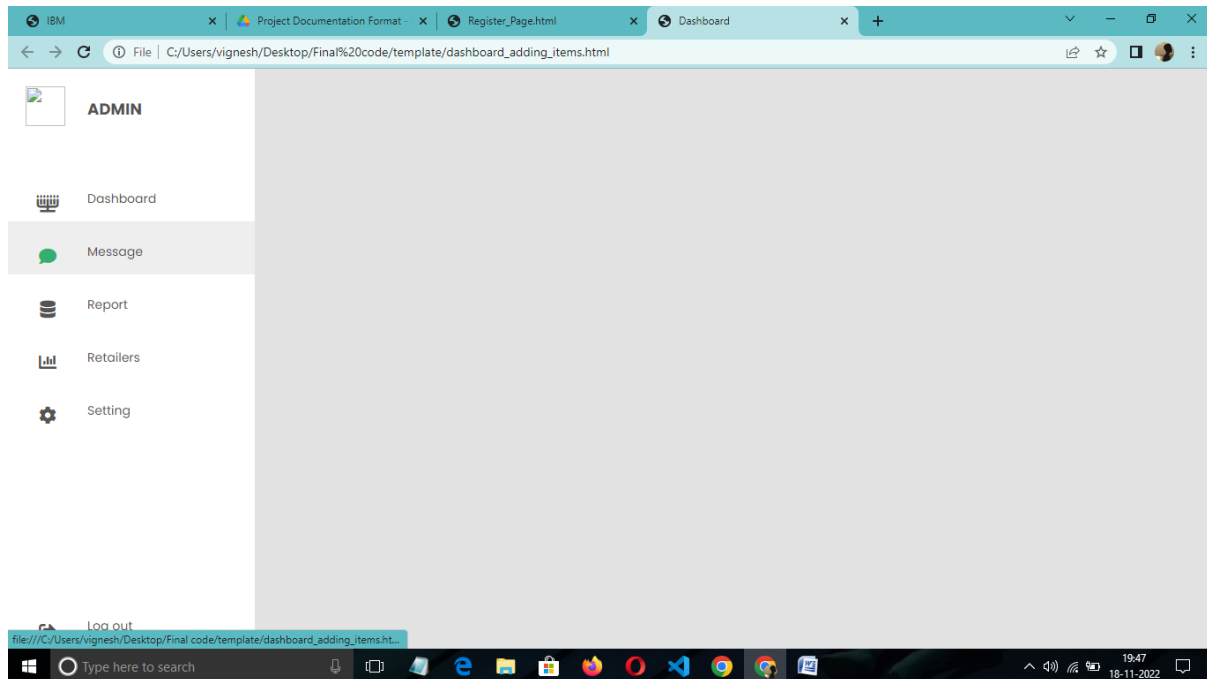
Output



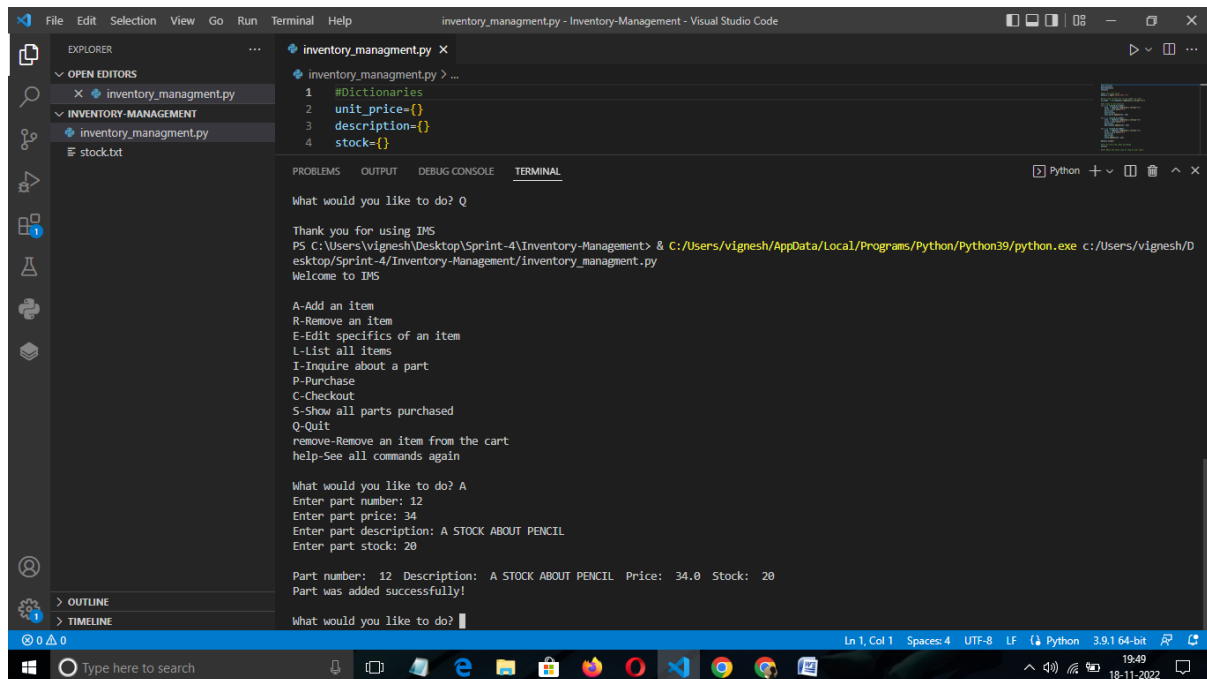
Signup page

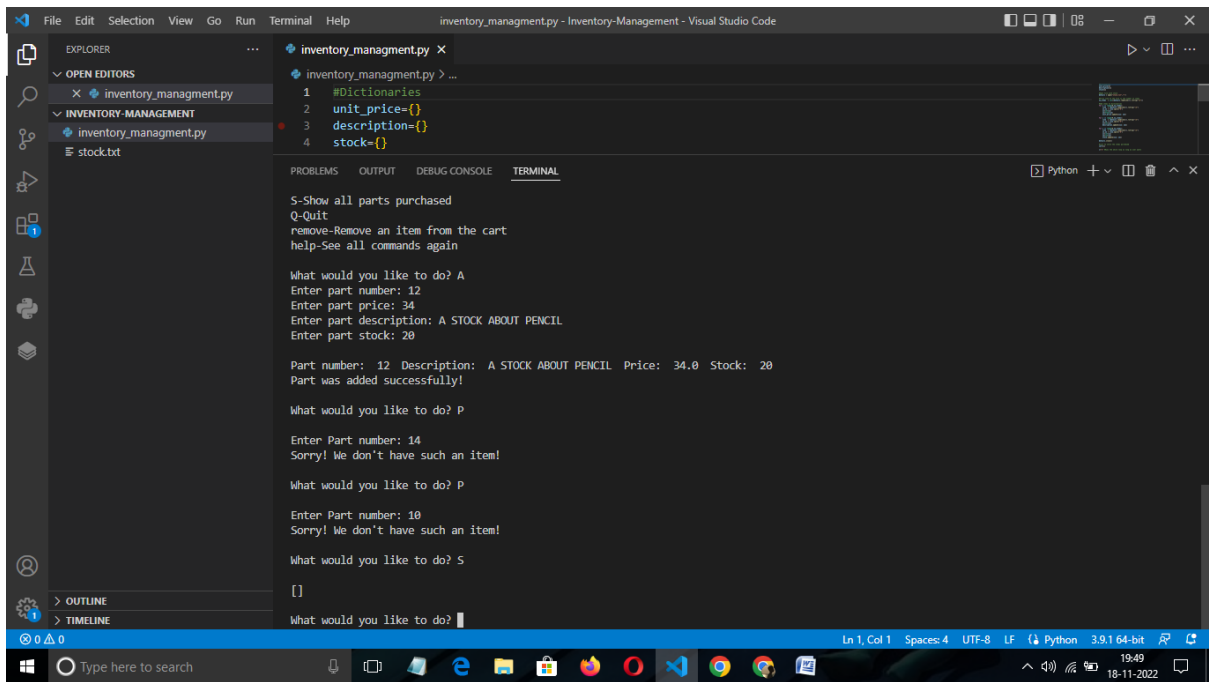


Dashboard



Code output





ADVANTAGES & DISADVANTAGES

Advantages

It helps to maintain the right amount of stocks: contrary to the belief that is held by some people, inventory management does not seek to reduce the amount of inventory that you have in stock, however, it seeks to maintain an equilibrium point where your inventory is working at a maximum efficiency and you do not have to have many stocks or too few stocks at hand at any particular point in time. The goal is to find that zone where you are never losing money in your inventory in either direction. With the aid of an efficient inventory management strategy, it is easy to improve the accuracy of inventory order.

It leads to a more organized warehouse: with the aid of a good inventory management system, you can easily organize your warehouse. If your warehouse is not organized, you will find it very difficult to manage your inventory. A lot of businesses choose to optimize their warehouse by putting the items that have the highest sales together in a place that is easy to access in the warehouse. This ultimately helps to speed up order fulfilment and keeps clients happy.

If you enter into a warehouse or a facility without proper inventory management, it is immediately apparent. Managers and business owners who do not implement an effective inventory management system usually have troubles keeping track of assets and executing work order so they just end up not being kept in the right place. It is not just an eyesore but it can also lead to a variety of safety hazards as well.

It saves time and money: an effective inventory management system can translate to time and money saved on the part of the business. By keeping track of the product that you already have at hand, you can save yourself the hassles of having to do an inventory recount in order to ensure your records are accurate. It also allows you to save cash that would have otherwise been spent on slow moving products.

Disadvantage

Bureaucracy: even though inventory management allows employees at every level of the company to read and manipulate company stock and product inventory, the infrastructure required to build such a system adds a layer of bureaucracy to the whole process and the business in general. In instances where inventory control is in-house, this includes the number of new hires that are not present to regulate the warehouse and facilitate transactions. In instances where the inventory management is in the hands of a third party, the cost is a subscription price and a dependence on another separate company to manage its infrastructure. No matter the choice you go for, it translates to a higher overhead cost and more layers of management between the owner and the customer. From the view point of the customer, a problem that requires senior management to handle will take a longer period of time before it will be trashed out.

Production problem: even though inventory management can reveal to you the amount of stock you have at hand and the amount that you have sold off, it can also hide production problems that could lead to customer service disasters. Since the management places almost all of its focus on inventory management to the detriment of quality control, broken or incorrect items that would normally be discarded are shipped along with wholesome items.

Increased space is need to hold the inventory: in order to hold inventory, you will need to have space so unless the goods you deal in are really small in size, then you will need a warehouse to store it. In addition, you will also need to buy shelves and racks to store your goods, forklifts to move around the stock and of course staff. The optimum level of inventory for a business could still be a lot of goods and they will need space to be stored in and in some cases additional operational costs to manage the inventory. This will in turn increase cost and impact negatively on the amount of profit the business makes.

Complexity: some methods and strategies of inventory management can be relatively complex and difficult to understand on the part of the staff. This may result in the need for employees to undergo training in order to grasp how the system works.

CONCLUSION

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

FUTURE SCOPE

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

Legend has it that Tim Cook of Apple once exclaimed: “Inventory is fundamentally evil!” The Toyota Production System (TPS), and Lean Thinking sees inventory as “muda” (waste). The message is clear, but it is also an exaggeration for effect. Insightful companies will see that too much inventory is waste, but that inventory, deployed in the right quantities at the right time in the right places, can provide substantial benefit to businesses and enhance competitive advantage.

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

The supply chain concept has been encouraging practitioners for decades to knock down the walls of communication and destroy functional silos that interfere with collaboration. Visionary companies will see that there is a lot more work to do to fully embrace this ethic.

It is only by taking a system-wide view of inventory investment that practitioners can hope to optimize its deployment. At each stage of production, from raw materials to finished consumer goods, and at all transportation activities in between, the economies and trade-offs need to be analyzed. After analysis, the results need to be synthesized and impacts of various solutions assessed from the perspective of benefit to the entire supply chain.

Such holistic analysis requires effective and efficient communication, and cross-functional understanding. This can be enabled by education and training, along with leveraging technology.

APPENDIX

Source Code

Core code

```
#Dictionaries
unit_price={}
description={}
stock={}

#Open file with stock
details = open("stock.txt","r")

#First line of the file is the number of items
no_items = int((details.readline()).rstrip("\n"))

#Add items to dictionaries
for i in range(0,no_items):
    line = (details.readline()).rstrip("\n")
    x1,x2 = line.split("#")
    x1=int(x1)
    x2=float(x2)
    unit_price.update({x1: x2})

for i in range(0,no_items):
    line = (details.readline()).rstrip("\n")
    x1,x2 = line.split("#")
    x1=int(x1)
    description.update({x1: x2})

for i in range(0,no_items):
    line = (details.readline()).rstrip("\n")
    x1,x2 = line.split("#")
    x1=int(x1)
    x2=int(x2)
    stock.update({x1: x2})

details.close()

#List to store the items purchased
cart=[]

c="y" #Runs the while loop as long as user wants

#Instructions
print("Welcome to IMS")
print()
```

```

print("A-Add an item")
print("R-Remove an item")
print("E-Edit specifics of an item")
print("L-List all items")
print("I-Inquire about a part")
print("P-Purchase")
print("C-Checkout")
print("S-Show all parts purchased")
print("Q-Quit")
print("remove-Remove an item from the cart")
print("help-See all commands again")
print()

total_cost=0
flag=0 #To check if they have checked out

while(c!= "q" or c!= "Q"):
    c= input("What would you like to do? ")

    if(c=="q" or c=="Q"):
        break

    elif(c=="A" or c=="a"):#Add a part
        p_no = int(input("Enter part number: "))
        p_pr = float(input("Enter part price: "))
        p_desc = input("Enter part description: ")
        p_stock = int(input("Enter part stock: "))

        m=0
        for i in range(0,len(unit_price)):
            if(p_no in unit_price):
                p_no+=1
                m=1
        if(m==1):
            print()
            print("That part number already exists :(, changing value to
",p_no)

        unit_price.update({p_no: p_pr})
        description.update({p_no: p_desc})
        if(p_stock > -1):
            stock.update({p_no: p_stock})
        else:
            p_stock = 0
            stock.update({p_no: p_stock})
            print("The stock of an item cannot be negative, the stock has been
set to 0.")

```

```

        print()
        print("Part number: ",p_no," Description: ",description.get(p_no),"
Price: ",unit_price.get(p_no)," Stock: ",stock.get(p_no))
        print("Part was added successfully!")
        print()

    elif(c=="E" or c=="e"):#Edit a part
        print()
        p_no = int(input("Enter part number: "))
        if(p_no in unit_price):
            p_pr = float(input("Enter part price: "))
            p_desc = input("Enter part description: ")
            p_stock = int(input("Enter part stock: "))

            unit_price.update({p_no: p_pr})
            description.update({p_no: p_desc})
            stock.update({p_no: p_stock})

        else:
            print("That item does not exist, to add an item use a")
            print()

    elif(c=="R" or c=="r"):#Remove a part
        print()
        p_no = int(input("Enter part number: "))
        if(p_no in unit_price):
            are_you_sure = input("Are you sure you want to remove that
item(y/n)? ")
            if(are_you_sure=="y" or are_you_sure=="Y"):
                unit_price.pop(p_no)
                description.pop(p_no)
                stock.pop(p_no)
                print("Item successfully removed!")
            print()
        else:
            print("Sorry, we don't have such an item!")
            print()

    elif(c=="L" or c=="l"):#List all the parts
        print()
        print("Parts and their prices: ",unit_price)
        print("Descriptions: ",description)
        print("Stock left of Item: ",stock)
        print()

    elif(c=="I" or c=="i"):#Inquire about a part
        print()

```

```

        p_no=int(input("Enter Part Number: "))
        if(p_no in unit_price):
            print()
            print("Part number: ",p_no," Description:
",description.get(p_no)," Price: ",unit_price.get(p_no)," Stock:
",stock.get(p_no))
            if(stock.get(p_no)<3 and stock.get(p_no)!=0):
                print("Only ",stock.get(p_no)," remaining! Hurry!")
            print()
        else:
            print("Sorry we don't have such an item!")
            print()

    elif(c=="P" or c=="p"):#Purchase a part
        print()
        p_no = int(input("Enter Part number: "))
        if(p_no in unit_price):
            if(flag==1):
                flag=0
            stock_current = stock.get(p_no)
            if(stock_current>0):
                stock_current = stock.get(p_no)
                stock[p_no] = stock_current-1
                item_price = unit_price.get(p_no)
                total_cost = total_cost+item_price
                print(description.get(p_no),"added to cart: ", "$",item_price)
                cart.append(p_no)#Stores item in cart
            else:
                print("Sorry! We don't have that item in stock!")
        else:
            print("Sorry! We don't have such an item!")
        print()

    elif(c=="C" or c=="c"):#Check out
        print()
        print("You bought the following parts: ",cart)
        print("Total: ", "$",round(total_cost,2))
        tax= round(0.13*total_cost,2)
        print("Tax is 13%: ", "$",tax)
        total = round(total_cost+tax,2)
        print("After Tax: ", "$",total)
        total_cost=0
        flag=1
        print()
        print("You can still purchase items after check out, your cart has
been reset. To quit press q")
        print()

```



```

elif(c=="help"): #Display all commands
    print()
    print("Help Centre")
    print("A-Add an item")
    print("R-Remove an item")
    print("E-Edit specifics of an item")
    print("L-List all items")
    print("I-Inquire about a part")
    print("P-Purchase")
    print("C-Checkout")
    print("S-Show all parts purchased")
    print("remove-Remove an item from the cart")
    print("help-See all commands again")
    print("If you have any other questions or concerns please contact the
manager.")
    print()

elif(c=="remove" or c=="Remove"): #To remove an item from the cart
    print()
    are_you_sure = input("Are you sure you want to remove an item from the
cart(y/n)? ")
    if(are_you_sure=="y"):
        p_no = int(input("Enter part number to remove from cart: "))
        if(p_no in cart):
            stock_current = stock.get(p_no)
            stock[p_no] = stock_current+1
            item_price = unit_price.get(p_no)
            total_cost = total_cost-item_price
            j=0
            for i in range(0,len(cart)): #To find the index of the part in
the list cart
                if(i==p_no):
                    j=i

            cart.pop(j)
            print(description.get(p_no),"removed from cart: ")
            print()
        else:
            print()
            print("That item is not in your cart!")
            print()

elif(c=="s" or c=="S"): #prints list cart
    print()
    print(cart)
    print()

else:

```

```

        print()
        print("ERROR! Contact manager for help!")
        print()

#Outputs total if the user quits without checking out
if(total_cost>0 and flag==0):
    print()
    print("You bought: ",cart)
    print("Total: ", "$",round(total_cost,2))
    tax= round(0.13*total_cost,2)
    print("Tax is 13%: ", "$",tax)
    total = round(total_cost+tax,2)
    print("After Tax: ", "$",total)

print()
print("Thank you for using IMS")

#Write the updated inventory to the file
details = open("stock.txt","w")
no_items=len(unit_price)
details.write(str(no_items)+"\n")
for i in range(0,no_items):
    details.write(str(i+1)+"#" +str(unit_price[i+1])+"\n")

for i in range(0,no_items):
    details.write(str(i+1)+"#" +description[i+1]+"\n")

for i in range(0,no_items):
    details.write(str(i+1)+"#" +str(stock[i+1])+"\n")

details.close()

```

github link

<https://github.com/IBM-EPBL/IBM-Project-20699-1659760919>

project demo video link

<https://youtu.be/E4V2XlfM3cc>