

Inventory Management System for Retailers

PSG Institute of Technology and Applied Research

Report



Team Members

A S Kareshmmaa
A K Sharan Shankar
Shubham Ghosh
Vishal K

INDEX

1. INTRODUCTION

- i. Project Overview
- ii. Purpose

2. LITERATURE SURVEY

- i. Existing problem
- ii. References
- iii. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- i. Empathy Map Canvas
- ii. Ideation & Brainstorming
- iii. Proposed Solution
- iv. Problem Solution fit

4. REQUIREMENT ANALYSIS

- i. Functional requirement
- ii. Non-Functional requirements

5. PROJECT DESIGN

- i. Data Flow Diagrams
- ii. Solution & Technical Architecture
- iii. User Stories
- iv. Customer Journey Map

6. PROJECT PLANNING & SCHEDULING

- i. Sprint Planning & Estimation
- ii. Sprint Delivery Schedule
- iii. Reports from JIRA

7. CODING & SOLUTION

- i. Setting up the Application Environment
- ii. Deployment of Application in IBM Cloud
- iii. Integrating SendGrid Service
- iv. Implementing Web Application

8. TESTING

- i. Defect Analysis
- ii. Test Case Analysis

9. ADVANTAGES & DISADVANTAGES

10. RESULTS AND CONCLUSION

11. FUTURE SCOPE

12. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

PROJECT OVERVIEW:

A difficult issue in supply chain management is inventory management. The company's issue is that they have no system in place to monitor inventory data. The store finds it challenging to keep track of the inventory information. The main challenge for every inventory stock management is keeping track of how much stock is bought and how much stock is used. In this case, a tool or system to help with inventory management would be useful. The term "inventory management" describes the control of the quantity, quality, location, and transportation of a wide range of goods used in manufacture by a range of commercial enterprises or in sales by a range of retailers.

PURPOSE:

Typically, inventory management systems are constrained and restricted to a predetermined range of items and are unable to be altered and expanded in response to the needs of the client. The Inventory Management System puts a lot of effort into making it flexible and simple for the end user to use, with ongoing customer support to change the use. Inventory Management System concentrates on making it easier by adding details of different other entities that are a part of the company, in contrast to other software that offers comparable features.

2. LITERATURE SURVEY

EXISTING PROBLEM:

1. Design of a Computerized Inventory Management System for Supermarkets

Aim:

The purpose of this article is to develop a computerised inventory management system that will help managers make decisions about how much inventory to order, when to order more inventory, and how to keep track of transactions.

Scope:

This work focuses on stock control, management, and tends to fix business abnormalities. It examines the capacity to view current ones as well as the opening of new stocks and stock updates.

Implementation:

New stock, stock orders, stock updates, product or item searches to determine availability, and stock reports are all handled electronically by this system.

- To ensure security, a login page is made.
- If it is successful, the splash screen will appear, followed by an automatic display of the main menu form.
- The user can select from the New Stock, Update Stock, Search, and View options on the onscreen menu.
- Stock information will be recorded in the user form.
- Which are later fetched from the database, displayed, and kept there.

2. The inventory management system for automobile spare parts in a central warehouse

Aim:

In order to manage the inventory of car parts in a central warehouse, this research seeks to design an expanded fuzzy neural network (EFNN) based decision support system.

Scope:

To achieve greater accuracy than using an artificial neural network. This project incorporates domain experts' expertise into enhanced fuzzy neural networks (EFNN), which create connection weights based on the fuzzy analytic hierarchy process (AHP) method without laboriously and slowly rotating them.

Implementation:

Three parts make up the proposed system.

The fuzzy AHP's hierarchical structure development

The subject matter specialists are questioned regarding the parts, demand, timing, sales, and other relevant aspects that affect the supply of spare parts.

Weights determination: Based on the suggested framework, a new questionnaire is created. Comparing pairs of items from each level to each element in the level above using questionnaire surveys. There is a 7-point scale in use.

Making decisions using EFNN.

To improve accuracy, the EFNN, a five-layered hybrid neural network with the ability to self-organize its activation function, is used.

3. Design of smart inventory management system for construction sector based on IoT and cloud computing.**Aim:**

A novel approach to create a model and show how this can help construction sector in managing inventory of essential form work shuttering products.

Scope:

This research reveals that there could be an opportunity to approach barcode-based designs by amalgamating such with Cloud Computing, Arduino-based wireless station nodes, IoT and a secure form channel to access data through a dedicated web portal.

Implementation:

The proposed model is a novel Aluminium Shuttering Inventory Management System (ASIMS) consist of barcodes, Arduino-based IoT devices, wireless sensor networks and Cloud Computing to track Aluminium formwork shuttering components under actual field conditions.

- Upon receipt of Aluminium formwork shuttering components from vendor at site, a Goods Receipt Note (GRN) entry is passed in the system.
- Physical verification of the received items and GRN process have needed to be completed.
- The barcode labels for the items are generated and printed. The printed barcode labels are then affixed on the formwork shuttering components.
- Using our proposed application, the component is labelled and then mapped with corresponding geolocational coordinates to enable tracking.
- Aluminium formwork shuttering components are often cut and resized according to localized requirements. During such process of resizing, the created new items have to be checked, verified physically and logged using our proposed software. Again, new barcode labels have been generated for the new components derived from the parent item.

4. Design of smart inventory management system for construction sector based on IoT and cloud computing.

Aim:

A novel approach to Design of smart inventory management system for construction sector based on IoT and cloud computing.

Implementation:

Inventory management and working capital management are routine activities for a specific company and firms. Inventory is one of the most significant components of current assets and current assets, in general, are part of working

capital, so it is crucial to understand the amount locked up in inventory and to manage the inventory in the best possible way. In this section of the research, we analyse and investigate the inventory management practises used by Steel Authority of India Limited.

A very important factor in the short-term liquidity position and a major factor in long-term profitability is the stock of raw materials, work in progress, finished goods, etc. In this research project, data for the previous five years is gathered from the company's annual reports, and various ratios are then applied to the data in order to quantify inventory efficiency. In addition, we used a variety of statistical tools to examine the behaviour of selected ratios.

5. Relationship Between Inventory Management and Profitability: An Empirical Analysis of Indian Cement Companies.

Aim:

An analysis on relationship Between Inventory Management and Profitability: An Empirical Analysis of Indian Cement Companies.

Implementation:

The goal of this essay is to investigate the connection between a firm's profitability and its inventory conversion period. For a sample of five top Indian cement companies over a ten-year period, from 2001–2010, the relationship between inventory management and profitability is examined. Gross operating profit is employed as the dependent variable as a measure of profitability. The impact of the inventory conversion period on gross operating profit is examined using regression analysis in this study, which also takes into account the firm's size, current ratio, and financial debt ratio as control variables. The findings show a significant negative linear association between profitability and the inventory conversion period.

The outcomes of this study corroborate earlier conclusions. According to the results, the inventory conversion period and firm profitability are inversely related; that is, as ICP days rise, a company's profitability declines, and vice

versa. It was discovered that the ratio of financial debt and corporate profitability, as determined by GOP, are negatively correlated. Inferred from this was the notion that profitability rises when the ratio of financial debt falls. According to this study's findings, there was a positive association between firm size and GOP, indicating that profitability rises as business size increases. GOP support had a negative correlation to current ratio.

REFERENCES:

1. Abisoye, O. A., Boboye, F., & Abisoye, B. O. (2013). Design of a computerized inventory management system for supermarkets.
2. Li, S. G., & Kuo, X. (2008). The inventory management system for automobile spare parts in a central warehouse. *Expert Systems with Applications*, 34(2), 1144-1153.
3. Saleem, A. (2020). Automated inventory management systems and its impact on supply chain risk management in manufacturing firms of Pakistan. *Int J Supply Chain Manag*, 9, 220-231.
4. Bose, R., Mondal, H., Sarkar, I., & Roy, S. (2022). Design of smart inventory management system for construction sector based on IoT and cloud computing. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 2, 100051.
5. Kasim, H., Zubieru, M., & Antwi, S. K. (2015). An assessment of the inventory management practices of small and medium enterprises (SMEs) in the Northern Region of Ghana. *European Journal of Business and Management*, 7(20), 28-40.
6. Panigrahi, C. M. A. (2013). Relationship between inventory management and profitability: An empirical analysis of Indian cement companies. *Asia Pacific Journal of Marketing & Management Review*, 2(7).

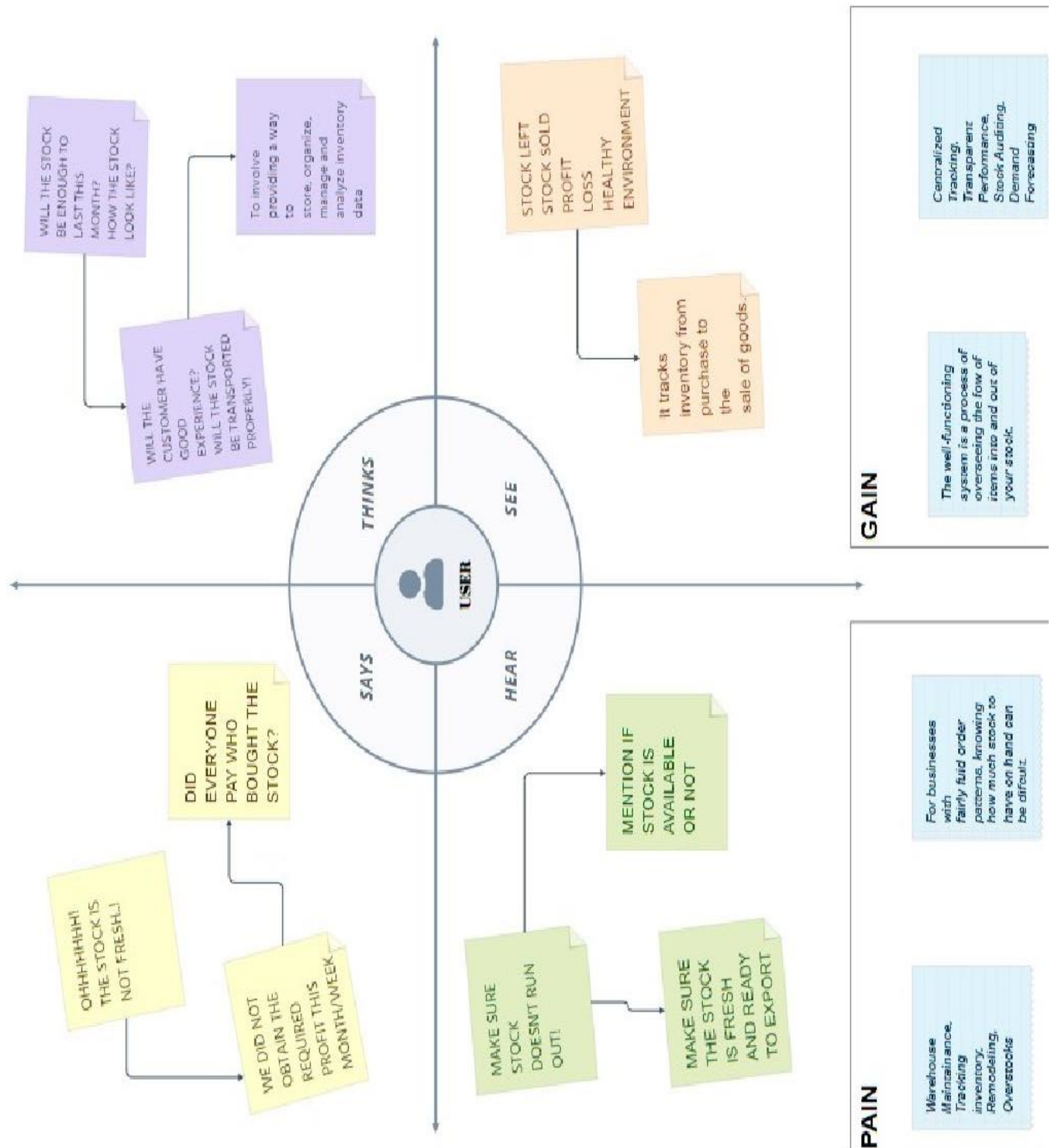
PROBLEM STATEMENT DEFINITION:

Inventory management is a challenging aspect of supply chain management. The problem for the company is that they don't have a system in place to track inventory data. It is difficult for the store to maintain track of the inventory data. Keeping track of how much stock is purchased and used is the key problem for inventory stock management. A system or tool to assist with inventory management would be helpful in this situation. The control of the quantity, quality, placement, and transportation of a large variety of items utilised in production by a variety of commercial enterprises or in sales by a variety of retailers is referred to as inventory management.

Most inventory management systems are limited to a fixed set of items and are not able to be changed or expanded in response to the client's needs. The Inventory Management System makes a lot of effort to be adaptable and user-friendly, and it offers ongoing customer assistance to adjust the use. Unlike other software that provides equivalent functions, Inventory Management System focuses on simplifying it by adding details of many other entities that are a part of the firm.

3. IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS:



IDEATION AND BRAINSTORMING:

Kareshmaa A S

Utilize a cloud-based inventory management system with real-time data backup and automated inventory updates to centralize your tracking data.	To reduce warehouse inefficiencies, track and report warehouse performance measures including inventory turnover, customer satisfaction, and order processing time.	Limited Visibility
Building a strategy that provides business - Agility, faster and flexible deployment across hyperscale providers with operational efficiency	Manual Documentation	Supply Chain Complexity
Provide workers with the appropriate inventory tools for the job.	Decentralized Design	Lack of System Optimization

Sharan Shankar A K

improve accuracy	better inventory planning and forecasting	saves your money and fulfill your customer needs
improves bussiness planning	saves time	business should strive to a sweet spot
reduce costs	improving supply chain operations	without ever running out of stock

Shubam Ghosh

better inventory control	savings in labour and procurement staff	reduced lead times
make and hold stock	increased efficiency	cost savings
secure and competitive pricing	can manage high demand	it avoids out of dates

Vishal K

Choose an appropriate fulfillment option.	Take forecasting seriously	Use EOQ for optimal order quantities
Set reorder points for each product	Give each variant a dedicated warehouse bin.	Prioritize with ABC analysis.
Sell older inventory frst	Try to implement	Automate as much as possible.

PROPOSED SOLUTION:

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none">• Customers are dissatisfied with the retailer's store because it lacks sufficient supplements and deliveries were not produced on time.• The retailers typically experience problems recording the stocks and its threshold limit available.
2.	Idea / Solution description	<ul style="list-style-type: none">• The daily update system in this proposed system will be activated once a product is sold or renewed more.• The availability of the products is monitored daily, and an alarm system is kept active to warn of any products that fall below the predetermined threshold.• All consumers can open an account by registering, and they will then receive login information they can use whenever they want to buy stocks.• The application gives clients access to information about all of the current stock options as well as when new stock will be put on sale in the store.
3.	Novelty / Uniqueness	<ul style="list-style-type: none">• Specific machine learning techniques are utilised to forecast the high-demand seasonal products that can be made available at that time.• Based on their acceptance, cost, and levels of consumer pleasure and trust, specific product predictions of the bestselling brand will be put into practise.• If a product that customers have been searching for is not available, notifications will be issued to the shops so that the product can be stocked up quickly.

4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> • The clients will be extremely satisfied because less time will be wasted looking for a product that isn't available. • Why If the system is automated every day and at every purchase, the workload of the retailers will be kept to a minimum. • As a result of receiving prompt and suitable responses from shops, customer happiness will increase.
5.	Business Model (Revenue Model)	<p>Hereby we can provide a robust and most reliable inventory management system by using:</p> <p>ML algorithms for all prediction needs employing all historical data since datasets are unquestionably abundant.</p> <p>Has the best business advertising models available.</p> <p>To develop a plan for preventing losses.</p> <p>To guarantee the system of perpetual, global product availability.</p>
6.	Scalability of the Solution	<ul style="list-style-type: none"> • The use of a system that everyone and anywhere may use can enable even the average person purchase the products. • The daily and regular updating of stock purchases to stop inventory shrinkage.

PROBLEM SOLUTION FIT:

10/26/22, 1:49 PM

Problem_solution_fit Sample Template - Google Docs

Project Title: INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID43303

Define CS, fit into CC		Explore AS, differentiate		
1. CUSTOMER SEGMENT(S) Business Owners, Stock holders, Departmental Store.	CS	6. CUSTOMER CONSTRAINTS One of the most limiting factors is <u>BUDGET</u> . Low powered systems, Not enough technical knowledge to operate the high level and sophisticated management systems.	CC	
		5. AVAILABLE SOLUTIONS AS User usually don't have a proper and standardised method of keeping track of the stocks/products. The pre-existing solutions are either not user friendly or not scalable. Most of the pre-existing solutions are platform dependent and run on a local device. Most available solutions are highly sophisticated and are not required by small to medium sized business.	AS	
Focus on J&P, tap into BE, understand RC		Focus on J&P, tap into BE, understand RC		
3. TRIGGERS In Traditional pen and paper method, tracking stocks for a highly dynamic business becomes very difficult and mistakes in calculations and stock management become highly likely. In pre-existing digital solutions, Remote management is not possible. Business owners tend to travel to different places for which the stated point becomes relevant.	TR	2. JOBS-TO-BE-DONE / PROBLEMS J&P The solution that we are proposing will be suitable for small to medium size business. It will be platform independent and won't require powerful systems to RUN. This will be more user-friendly and easy to access. It keeps the data in more systematic way.	9. PROBLEM ROOT CAUSE RC Most of the businesses in INDIA are not very large and hence lack funds. Which forces them to use traditional pen and paper methodologies. Most solution developers don't focus on platform independence. Businesses tend to have large variety of stocks which cannot be tracked by pre-existing methods.	
		4. EMOTIONS: BEFORE / AFTER EM BEFORE: CONFUSED, STRESSED, LACK OF CONFIDENCE, NOT MANAGEABLE. AFTER: LUCID, UNDERSTATED, CONFIDENCE, MANAGEABLE.	7. BEHAVIOUR BE The customer usually hires external working force for keeping track of STOCK DATA. Some customers also approach various developers to develop a management system specifically catering to their needs. These softwares usually involve the aforementioned problems.	
Identify strong TR & EM		Identify strong TR & EM		
8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE They can manage the stock data remotely from anywhere with the help of software from any of their device. Compare their sales, profits in various periods of time. 8.2 OFFLINE Has to verify the authenticity of entered values. Take actions based on their performance (SALES).		10. YOUR SOLUTION SL The solution that we are proposing will be suitable for small to medium size business. It will be platform independent and won't require powerful systems to RUN. This will be more user-friendly and easy to access. It keeps the data in more systematic way.		

https://docs.google.com/document/d/1MxfWIAzKq04b2b-7ZjAgX73s_JGAbbkSrR3FcURLt8/edit

1/2

4. REQUIREMENT ANALYSIS PHASE

FUNCTIONAL REQUIREMENTS:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registering through a form Registering through mail
FR-2	User Confirmation	Email confirmation OTP confirmation
FR-3	Login	Log in to the application by entering required credentials (email ID and password)
FR-4	Dashboard	View the products details (Name, quantity)
FR-5	Add items to the Inventory list	Users can add items that they wish to buy to the inventory
FR-6	Stock Updation	Increasing the availability of a particular product

NON-FUNCTIONAL REQUIREMENTS:

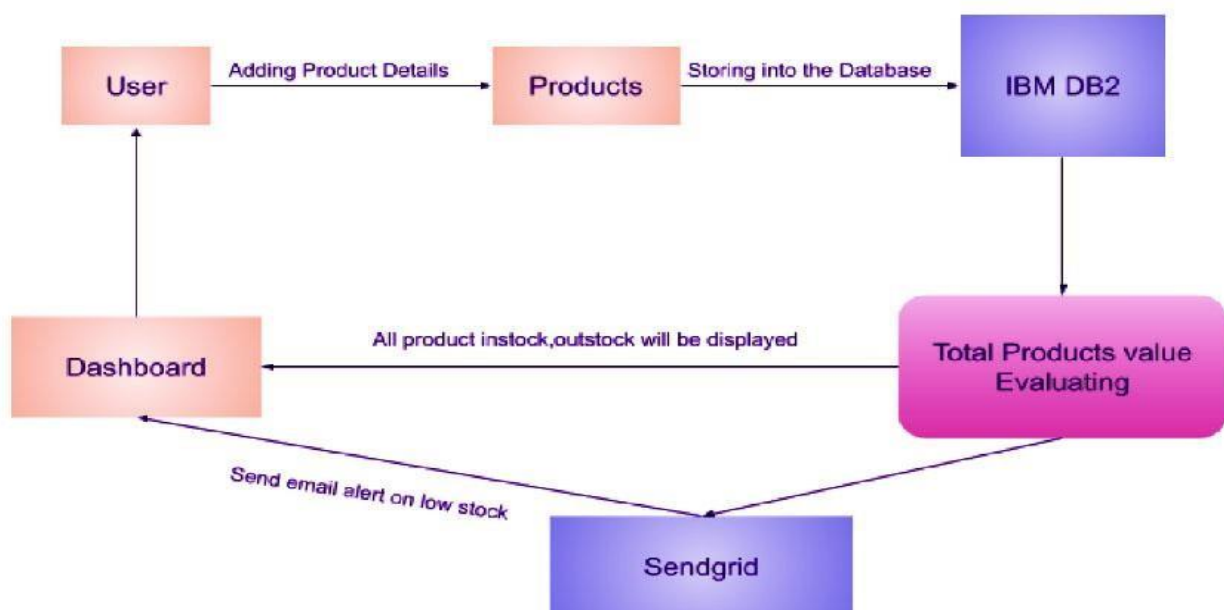
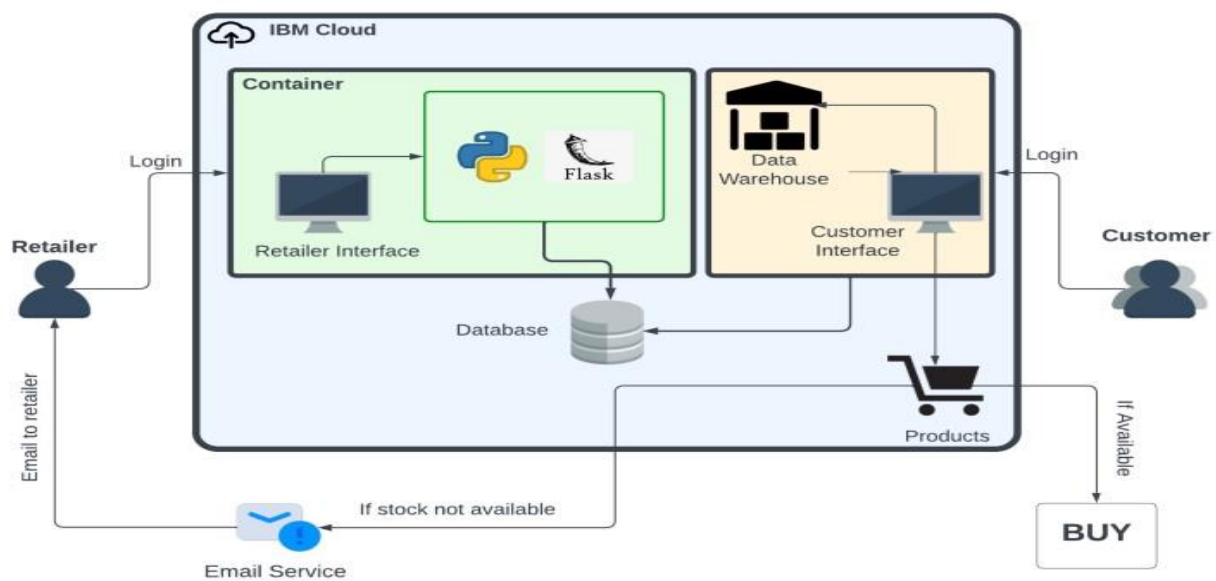
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	If the system has a steep learning curve, then it would mostly not be purchased by the company needing an inventory management system. <ul style="list-style-type: none">• The UI is simple and easy to navigate <input type="checkbox"/>• Consistent design and colours are used.• The webpages are responsive• Email delivery is to be fast
NFR-2	Security	Security refers to the safety and management of the inventory of a company such that only authorised personnel are allowed to access them. <ul style="list-style-type: none">• Login system is used to provide authentication.• Users need to create account and verify it with their email OTP.• Cookie based security is user for authentication on client side.
NFR-3	Reliability	<ul style="list-style-type: none">• Exception handling will be done at the code level to ensure that the app performs well even when errors happen in the runtime• Multiple instances of the App would be online ensure continued operation

NFR-4	Performance	<p>Performance of an inventory management system depends on the efficiency with which various tasks in it can be executed.</p> <ul style="list-style-type: none"> • Reduces manpower, cost and saves time. Emails will be sent automatically when stocks are not available. • Makes the business process more efficient. • Improves organizations performance. • It will be performed fast and secure even at the lower bandwidth
NFR-5	Availability	The use of IBM DB2 ensures high availability
NFR-6	Scalability	<p>The scalability of an inventory management system refers to the extensibility of its operations.</p> <ul style="list-style-type: none"> • DB2 is highly Scalable • The code is developed efficiently to easily add new features without many changes by reusing the code. • Docker in IBM Container registry is used which is highly scalable

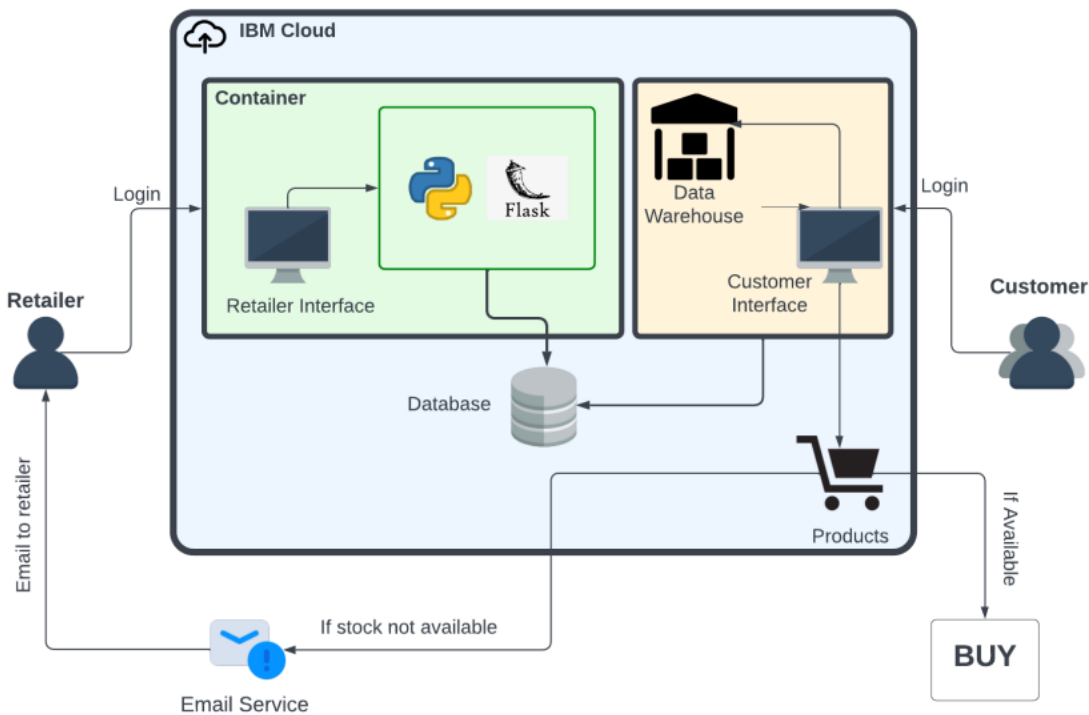
5. PROJECT DESIGN

DATA FLOW DIAGRAMS:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



SOLUTION ARCHITECTURE:



TECHNOLOGY STACK:

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	General purpose language	Python
3.	Application Logic-2	Containerization of platform	Docker
4.	Application Logic-3	Micro web framework	Flask
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Authentication and authorization using password and usernames.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	3-tier architecture- User, Retailer, Cloud Service


USER STORIES:

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 (Mobile 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 will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can register for the application through Gmail	High	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password	I can log into the application by entering email & password	High	Sprint-1
		USN-5	As a user, I should be able to recover my account in case I forget my credentials.	I am able to recover my account in case I forget my credentials.	High	Sprint-1
	Dashboard	USN-6	As a user, I should be able to view and edit my profile.	I am able to view and edit my profile.	Medium	Sprint-2
		USN-7	As a user, I should be able to view the stock details and modify them.	I am able to view the stock details and modify them.	High	Sprint-2
		USN-8	As a user, I would like a user-friendly and mobile-friendly interface.	The UI suits my preference.	Low	Sprint-3
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 will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can register for the application through Gmail	High	Sprint-1


User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Login	USN-4	As a user, I can log into the application by entering email & password	I can log into the application by entering email & password	High	Sprint-1
		USN-5	As a user, I should be able to recover my account in case I forget my credentials.	I am able to recover my account in case I forget my credentials.	High	Sprint-1
	Dashboard	USN-6	As a user, I should be able to view and edit my profile.	I am able to view and edit my profile.	Medium	Sprint-2
		USN-7	As a user, I should be able to view the stock details and modify them.	I am able to view the stock details and modify them.	High	Sprint-2
		USN-8	As a user, I would like a user friendly and minimalistic interface.	The UI suits my preference.	Low	Sprint-3
	Handling services	USN-1	As a customer care executive, I should be able to receive the user queries (services).	I am able to receive the user queries	Low	Sprint-4
		USN-2	As a customer care executive, I should be able to satisfy the user's needs via different modes.	I am able to satisfy the user's needs via different modes.	Low	Sprint-4
	Handling Problems	USN-1	As an administrator, I should be able to solve the problems reported by the other users.	I am able to solve the problems reported by the other users.	Medium	Sprint-5
		USN-2	As an administrator, I should be able to monitor the system in a simplistic manner.	I am able to monitor the system in a simplistic manner.	Medium	Sprint-5

CUSTOMER JOURNEY MAP:



Customer experience journey map

Use this framework to better understand customer needs, motivations, and obstacles by illustrating a key scenario or process from start to finish. When possible, use this map to document and summarize interviews and observations with real people rather than relying on your hunches or assumptions.

Created in partnership with  Product School


[Show template feedback](#)

Document an existing experience

Narrow your focus to a specific scenario or process within an existing product or service. In the **Steps** row, document the step-by-step process someone typically experiences, then add detail to each of the other rows.

TIP
As you fill steps in the **Steps** row, document the step-by-step process someone typically experiences, then add detail to each of the other rows.






Entice	Enter	Engage	Exit	Extend
How does someone initially become aware of this core need?	What do people experience as they begin the process?	In the core moments in the process, what happens?	What do people typically experience as they progress forward?	What happens after the experience is over?
Entice Research, building awareness, and enter a lead (if you have)	Enter What do people experience as they begin the process?	Engage In the core moments in the process, what happens?	Exit What do people typically experience as they progress forward?	Extend What happens after the experience is over?
Steps What steps does a person go through to fulfill their need?	Interactions What interactions do they have with the product or service?	Goals & motivations What goals and motivations drive a person's behavior?	Positive moments What steps does a person experience that are particularly satisfying, delightful, or exciting?	Negative moments What steps does a person experience that are particularly frustrating, confusing, surprising, costly, or time-consuming?
Area of opportunity How might we make each step better? What are the biggest opportunities?				



Need some inspiration?

Check out these examples of how others have used this template to create their own journey maps.

[Open examples](#)

6. PROJECT PLANNING & SCHEDULING

SPRINT PLANNING AND EXECUTION:

Activity Number	Activity Name	Detailed Activity Description	Assigned To	Duration (Start to End Date)	Status
1	Create Flask Project	An application Frame work written in Python.	Kareshmmaa	Completed	
2	Create IBM Cloud	Create and log into IBM Cloud.	Sharan		
3	Install IBM Cloud CLI	General-Purpose developer tool that provides access to your IBM Cloud Account.	Vishal		
4	Docker CLI	Use Docker CLI configuration to customize settings.	Shubham		
5	Create Account in SendGrid	Create account in SendGrid to send mails.	Kareshmmaa		
Implementing Web Application					
6	Create UI to interact with Application	Pages such as Registration, Login page, Displaying items etc.	Sharan	Completed	
7	Create IBM Db2 and connect with Python	Create IBM Db2 service in IBM Cloud and connect with python code using DB.	Shubham		
8	SendGrid Integration with Python Code	To send emails from the applications we need to integrate the SendGrid Service.	Vishal	Completed	
Deployment of App in IBM Cloud					
9	Containerize the App	Need to create Docker Image of the application and push into the IBM Container Registry.	Kareshmmaa	Completed	
10	Upload Image to IBM Container Registry	Upload the Image to IBM Container Registry.	Sharan		
11	Deploy in Kubernetes Cluster	Once the image is uploaded the IBM Container registry deploy the image to IBM Kubernetes Cluster.	Shubham		
Ideation Phase					
12	Literature Survey on the Selected Project and Information Gathering	Gather information about various other inventory management systems	Vishal	Completed	
13	Prepare Empathy Map	Analyse the gain and pain of the proposed project	Kareshmmaa		
14	Ideation and Brainstorming	Team members pool in their ideas on the project	Sharan		
Project Design Phase I					
15	Proposed Solution	Layout the solution to be implemented	Shubham	Completed	
16	Problem Solution Fit	Explore, identify and propose the use cases	Vishal		
17	Solution Architecture	Diagrammatically depict the solution	Kareshmmaa		

	Project design Phase II			
18	Customer Journey	Tells about how customers feel about the modules	Sharan	Completed
19	Functional Requirement	Specifies both functional and non-functional requirements	Shubham	
20	Dataflow Diagram	Flowgraph of the project	Vishal	
21	Technology Architecture	Tech stacks used in the model is listed	Kareshmmaa	
	Project Planning Phase			
22	Prepare Milestone and Activity List	Roles and responsibilities of each team member	Sharan	Completed
23	Sprint Delivery Plan	Various stages of the project is divided into modules	Shubham	
	Project Development Phase			
24	Sprint 1	Login and Registration details	Vishal	Completed
25	Sprint 2	Dashboard and add items into the cart	Kareshmmaa	
26	Sprint 3	Stock update	Sharan	
27	Sprint 4	Request to customer care and contact administrator	Shubham	

SPRINT DELIVERY SCHEDULE:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	4
Sprint-1		USN-2	As a user, I can register for the application through Email.	1	Medium	4
Sprint-1	Confirmation	USN-3	As a user, I will receive confirmation email once I have registered for the application.	2	Medium	4
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	2	High	4

Sprint-2	Dashboard	USN-5	As a user, I can view the products which are available	4	High	4
Sprint-2	Add items to cart	USN-6	As a user, I can add the products I wish to buy to the carts.	5	Medium	4
Sprint-3	Stock Update	USN-7	As a user, I can add products which are not available in the dashboard to the stock list.	5	Medium	4
Sprint-4	Request to Customer Care	USN-8	As a user, I can contact the Customer Care Executive and request any services I want from the customer care.	5	Low	4
Sprint-4	Contact Administrator	USN-9	I can be able to report any difficulties I experience as a report	5	Medium	4

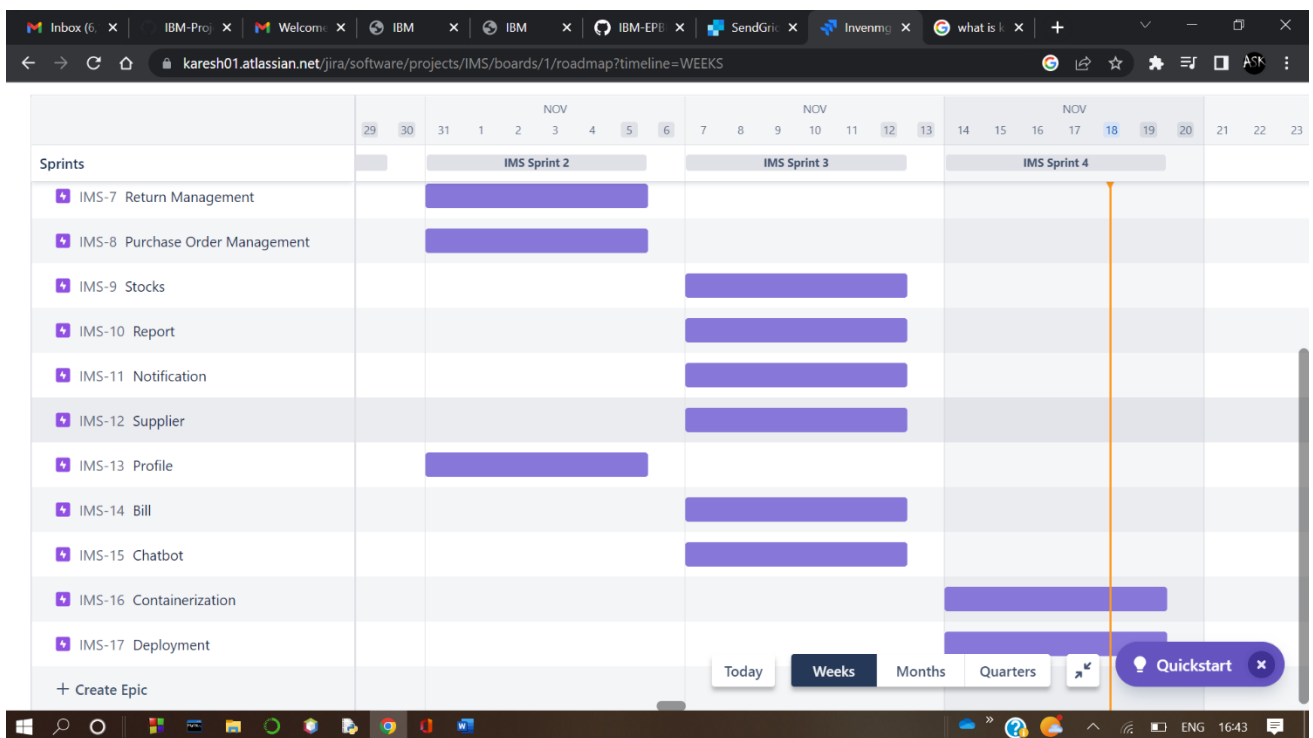
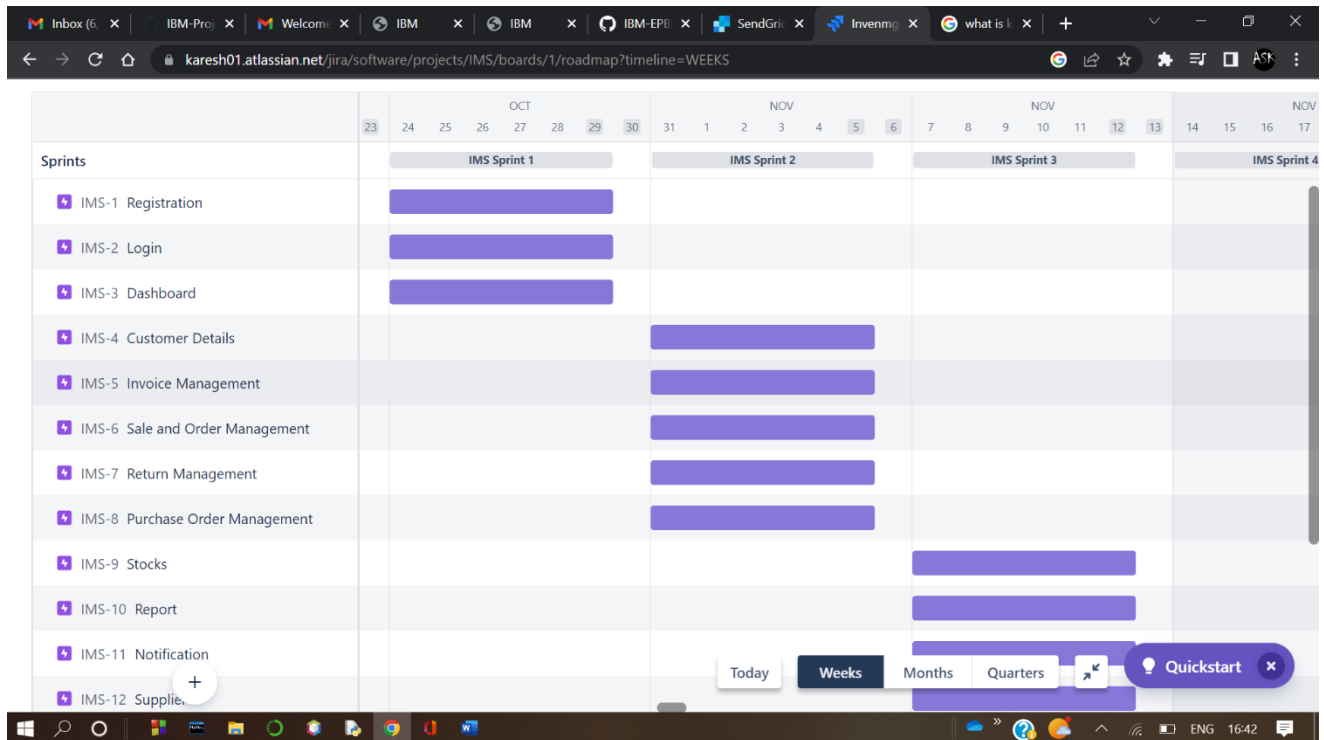
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	9	6 Days	31 Oct 2022	05 Nov 2022	9	05 Nov 2022
Sprint-3	5	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

Velocity:

Sprints	Sprint Duration	Velocity	Actual Velocity
Sprint-1	6	7	0.85
Sprint-2	6	9	0.66
Sprint-3	6	5	1.2
Sprint-4	6	10	0.6

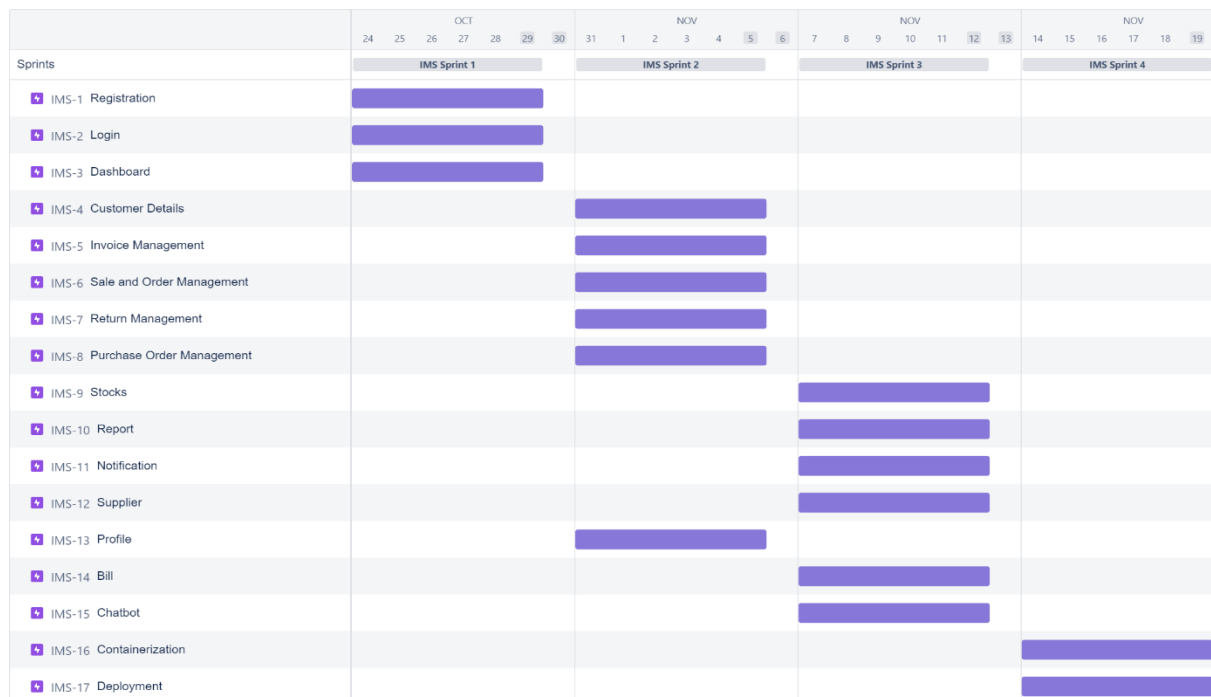
REPORTS FROM JIRA:

ROADMAP:

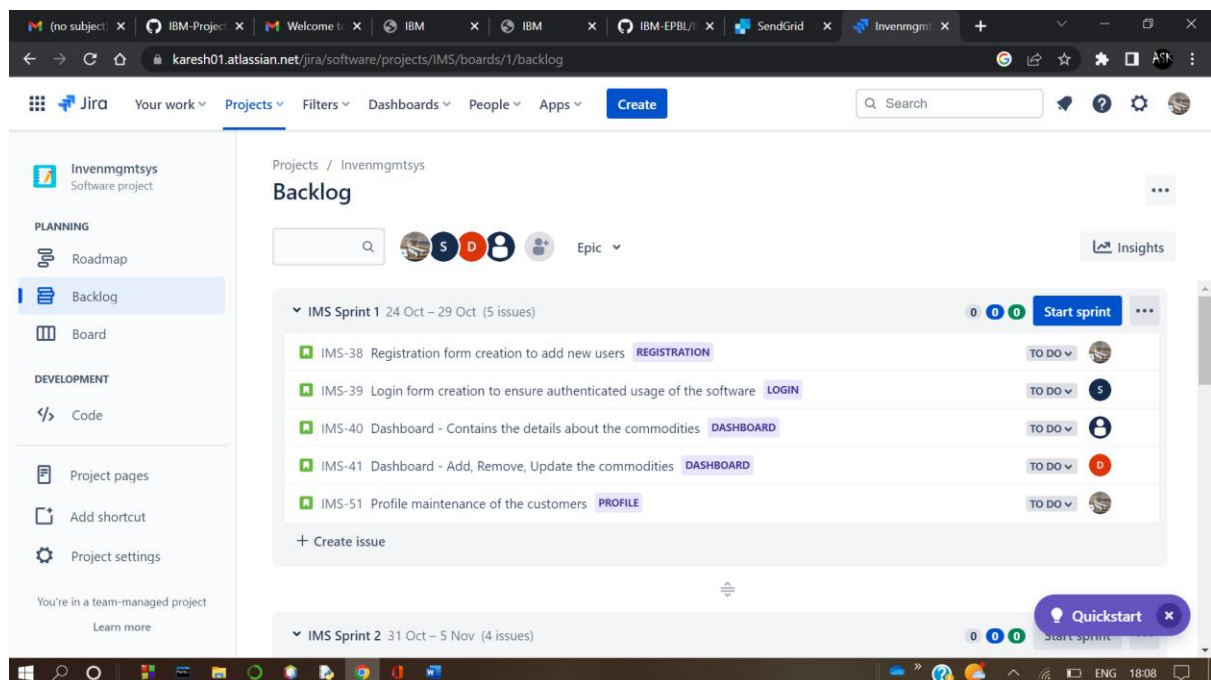


USAGE OF THE JIRA SOFTWARE:

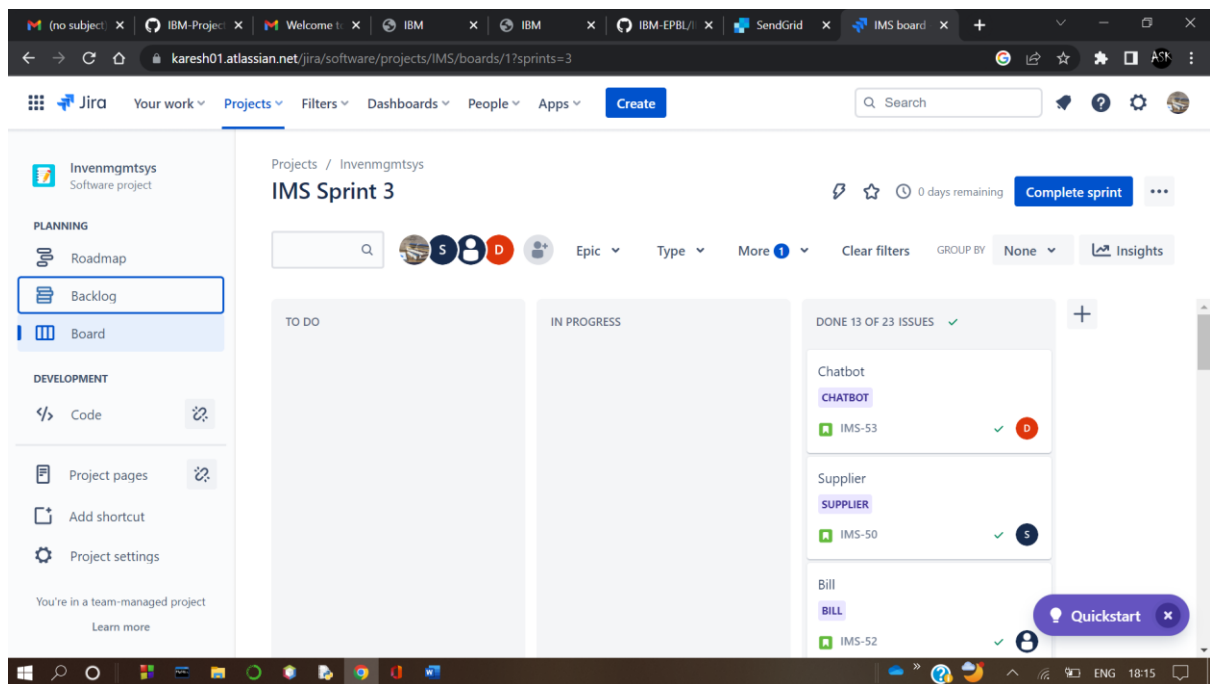
Roadmap for the project:



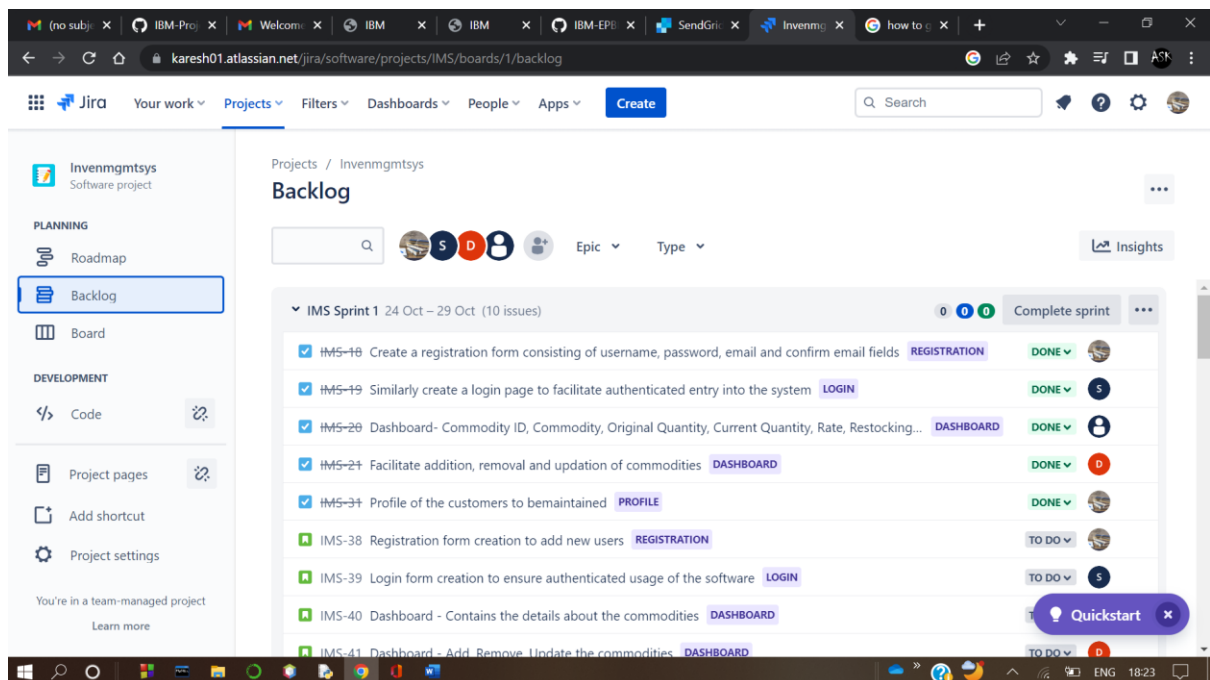
Creating the issue (Backlog) for the project:



Creating the board for the sprints of the project:



Completion of the assigned issue:



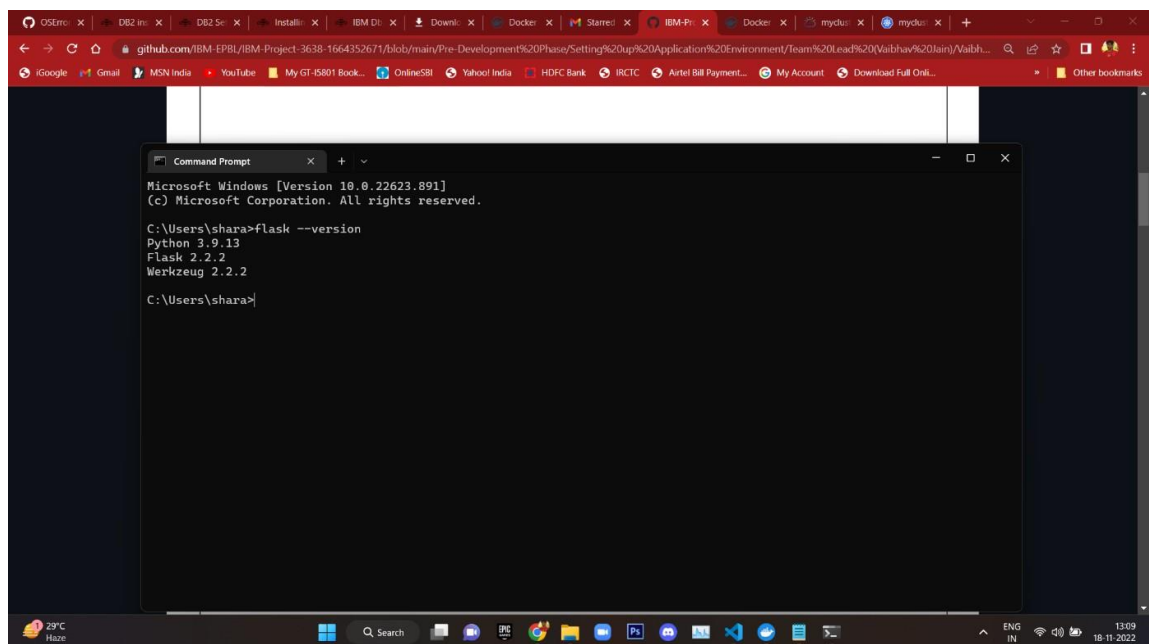
7. CODING AND SOLUTION

SETTING UP THE APPLICATION ENVIRONMENT:

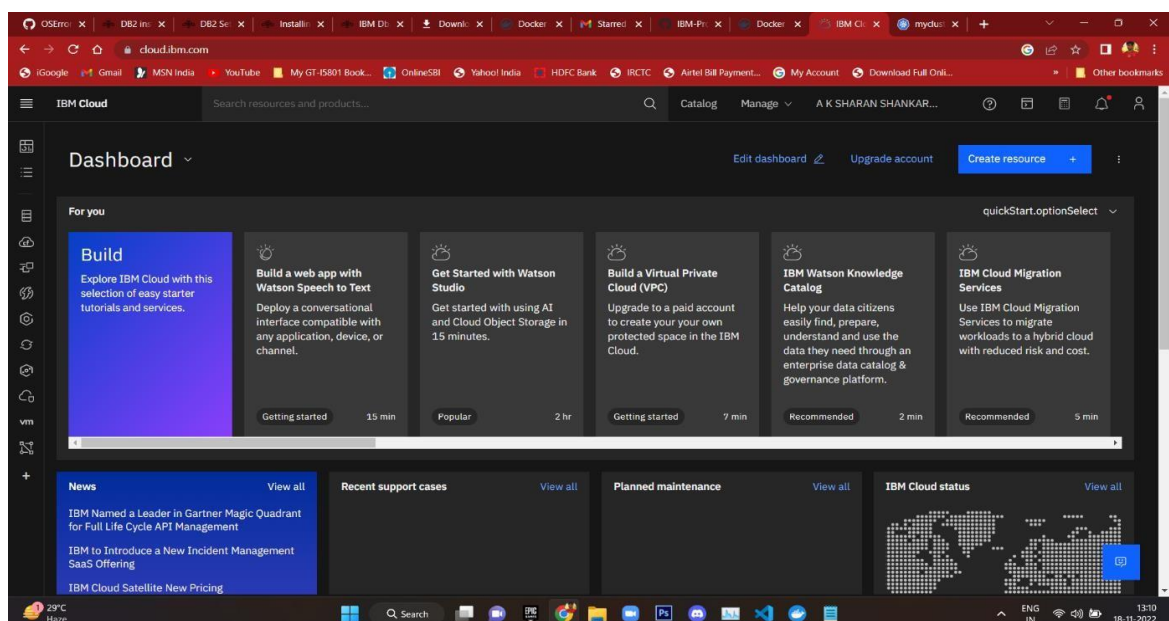
TASK

- Create Flask Project
- Create IBM Cloud Account
- Install IBM Cloud CLI
- Docker CLI Installation
- Create An Account in SendGrid

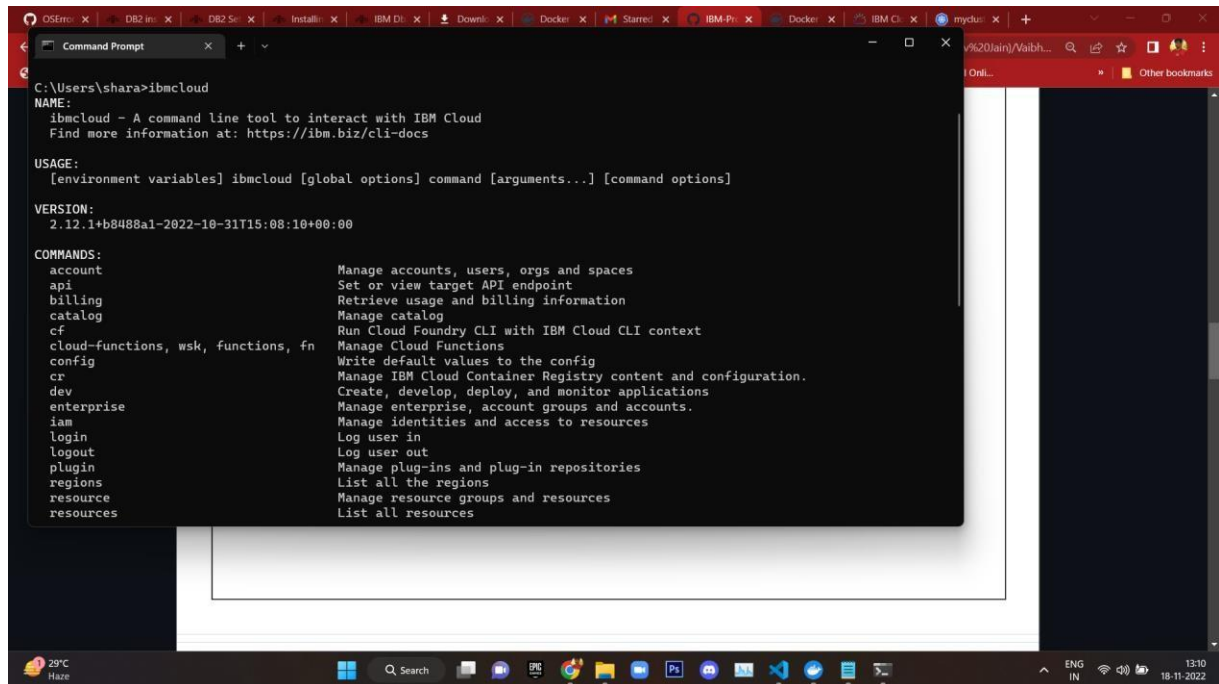
Create Flask Project



Create IBM Cloud Account



Install IBM Cloud CLI



```
C:\Users\shara>ibmcloud

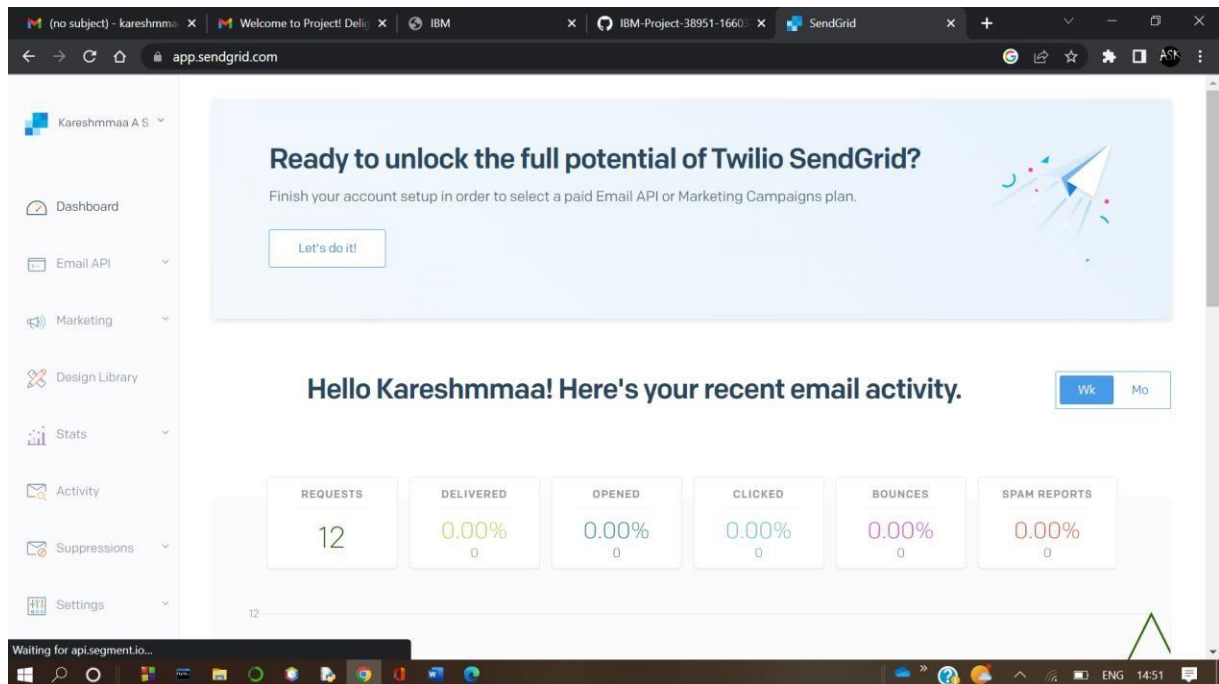
NAME:
  ibmcloud - A command line tool to interact with IBM Cloud
  Find more information at: https://ibm.biz/cli-docs

USAGE:
  [environment variables] ibmcloud [global options] command [arguments...] [command options]

VERSION:
  2.12.1+b8488a1-2022-10-31T15:08:10+00:00

COMMANDS:
  account      Manage accounts, users, orgs and spaces
  api          Set or view target API endpoint
  billing      Retrieve usage and billing information
  catalog      Manage catalog
  cf           Run Cloud Foundry CLI with IBM Cloud CLI context
  cloud-functions, wsk, functions, fn  Manage Cloud Functions
  config       Write default values to the config
  cr          Manage IBM Cloud Container Registry content and configuration.
  dev         Create, develop, deploy, and monitor applications
  enterprise  Manage enterprise, account groups and accounts.
  iam         Manage identities and access to resources
  login       Log user in
  logout      Log user out
  plugin      Manage plug-ins and plug-in repositories
  regions     List all the regions
  resource    Manage resource groups and resources
  resources   List all resources
```

Create An Account in SendGrid



Docker CLI Installation

```
Command Prompt
58c71ea40fb0: Waiting
2b0fb280b60d: Waiting
unauthorized: The login credentials are not valid, or your IBM Cloud account is not active.

C:\Users\shara>docker tag dockertest:latest jp2.icr.io/test1/dt_repo:1

C:\Users\shara>ibmcloud cr login --client docker
Logging 'docker' in to 'jp2.icr.io'...
Logged in to 'jp2.icr.io'.

OK

C:\Users\shara>
C:\Users\shara>docker push jp2.icr.io/test1/dt_repo:1
The push refers to repository [jp2.icr.io/test1/dt_repo]
30b0919f1d1: Pushed
01c46d8867f1: Pushed
d025656ee1: Pushed
f77b44634820: Pushed
5f354b8b5dc0: Pushed
f61107386c17: Pushed
db49993833a0: Pushed
58c71ea40fb0: Pushed
2b0fb280b60d: Pushed
1: digest: sha256:9b08f7f7584f684ce86c2dc6749fd331fb3585a2b37d5fbbef05a382049a1a4b size: 2204

C:\Users\shara>ibmcloud cr image-list
Listing images...

Repository      Tag    Digest          Namespace    Created      Size    Security status
jp2.icr.io/test1/dt_repo  1      9b08f7f7584f   test1        18 hours ago  38 MB   Scanning...

OK

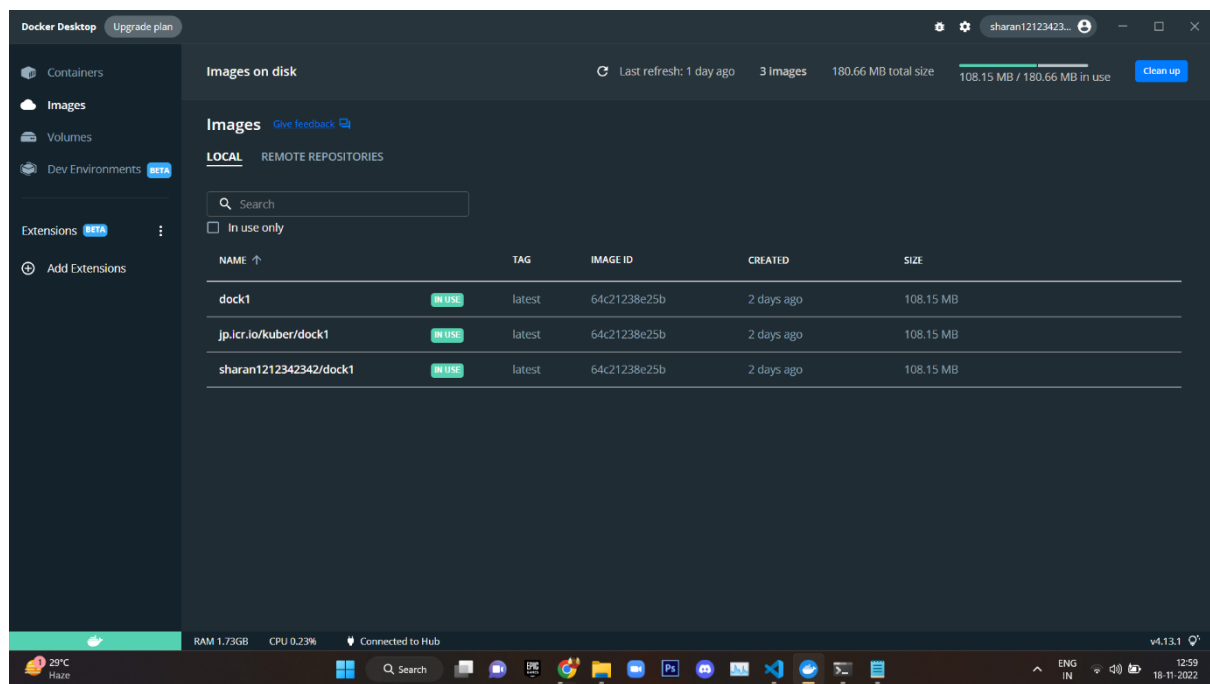
C:\Users\shara>
```

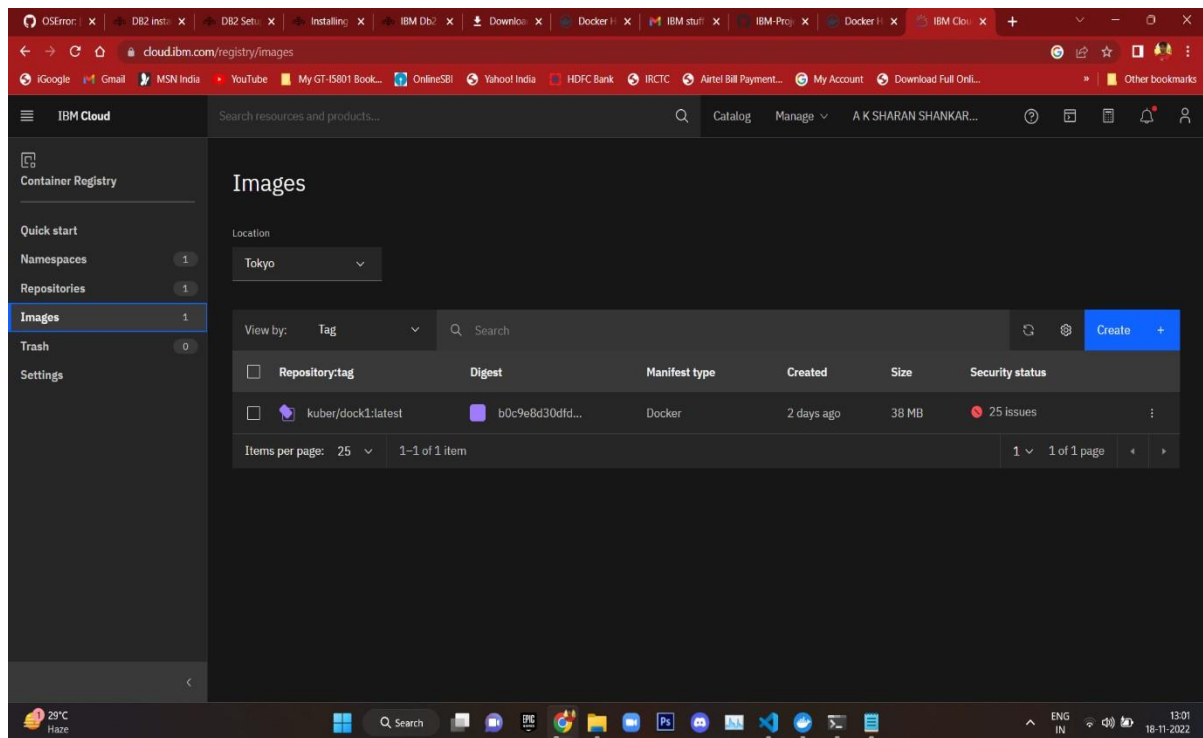
DEPLOYMENT OF APPLICATION IN IBM CLOUD:

TASK:

- Containerise the app
- Upload image to IBM container registry
- Deployment in Kubernetes cluster

Containerise the app:





Upload image to IBM Container Registry:

```

Command Prompt
58c71ea40fb0: Waiting
2b0fb280b6d: Waiting
unauthorized: The login credentials are not valid, or your IBM Cloud account is not active.

C:\Users\shara>docker tag dockertest:latest jp2.icr.io/test1/dt_repo:1

C:\Users\shara>ibmcloud cr login --client docker
Logging 'docker' in to 'jp2.icr.io'...
Logged in to 'jp2.icr.io'.

OK

C:\Users\shara>
C:\Users\shara>docker push jp2.icr.io/test1/dt_repo:1
The push refers to repository [jp2.icr.io/test1/dt_repo]
30b0919f1dc1: Pushed
81c46d8867f1: Pushed
d025656eee1: Pushed
f77b44634820: Pushed
5f354b8b5dc0: Pushed
f61107386c17: Pushed
db49993833a0: Pushed
58c71ea40fb0: Pushed
2b0fb280b6d: Pushed
1: digest: sha256:9b08f7f7584f684ce86c2dc6749fd331fb3585a2b37d5fbbef05a382049a1a4b size: 2204

C:\Users\shara>ibmcloud cr image-list
Listing images...

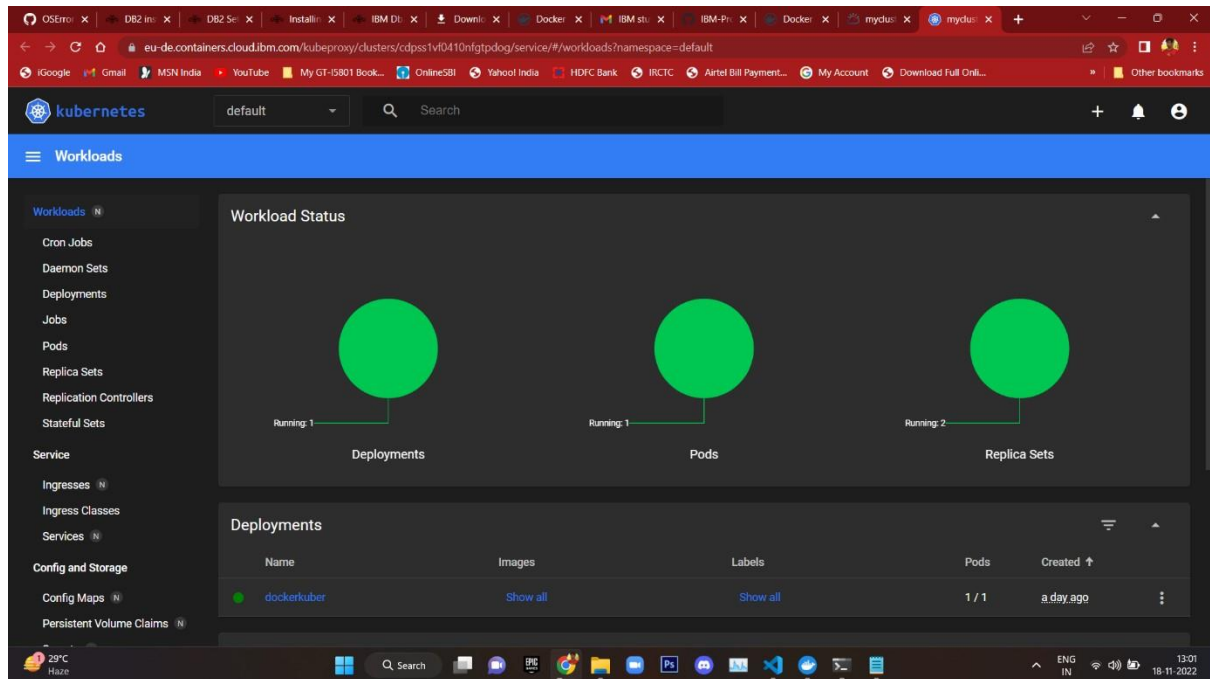
Repository      Tag  Digest          Namespace  Created      Size  Security status
jp2.icr.io/test1/dt_repo  1    9b08f7f7584f    test1      18 hours ago  38 MB  Scanning...

OK

C:\Users\shara>

```

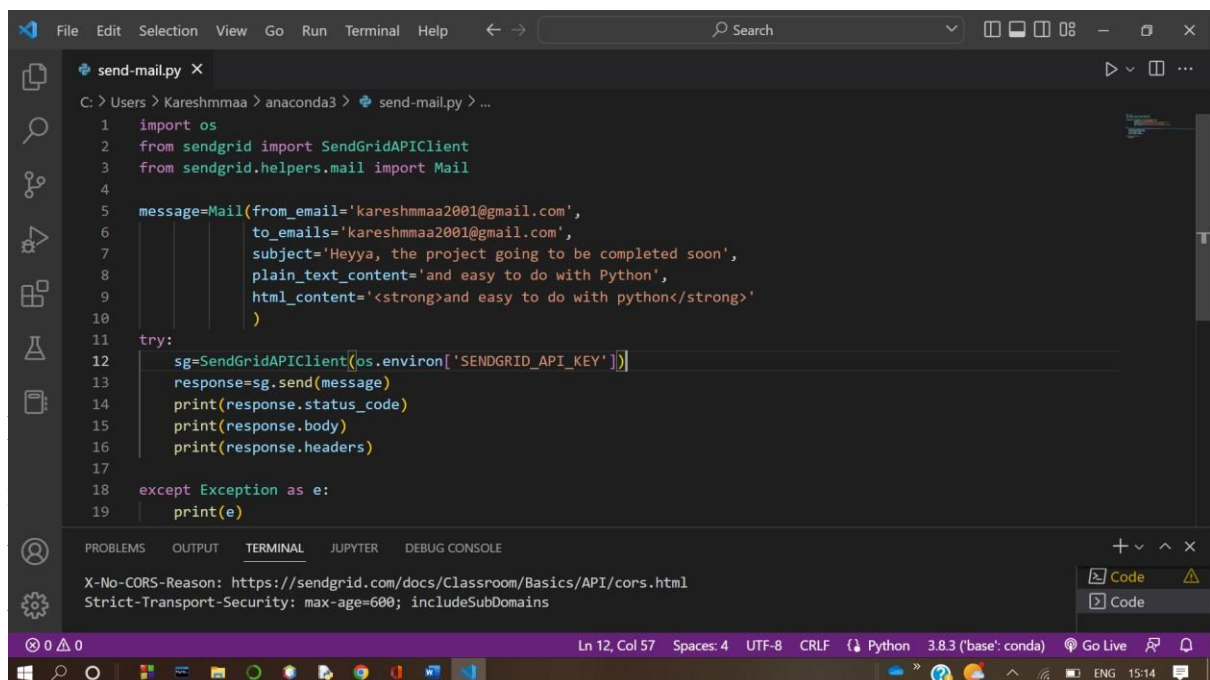
Deploy in Kubernetes Cluster:

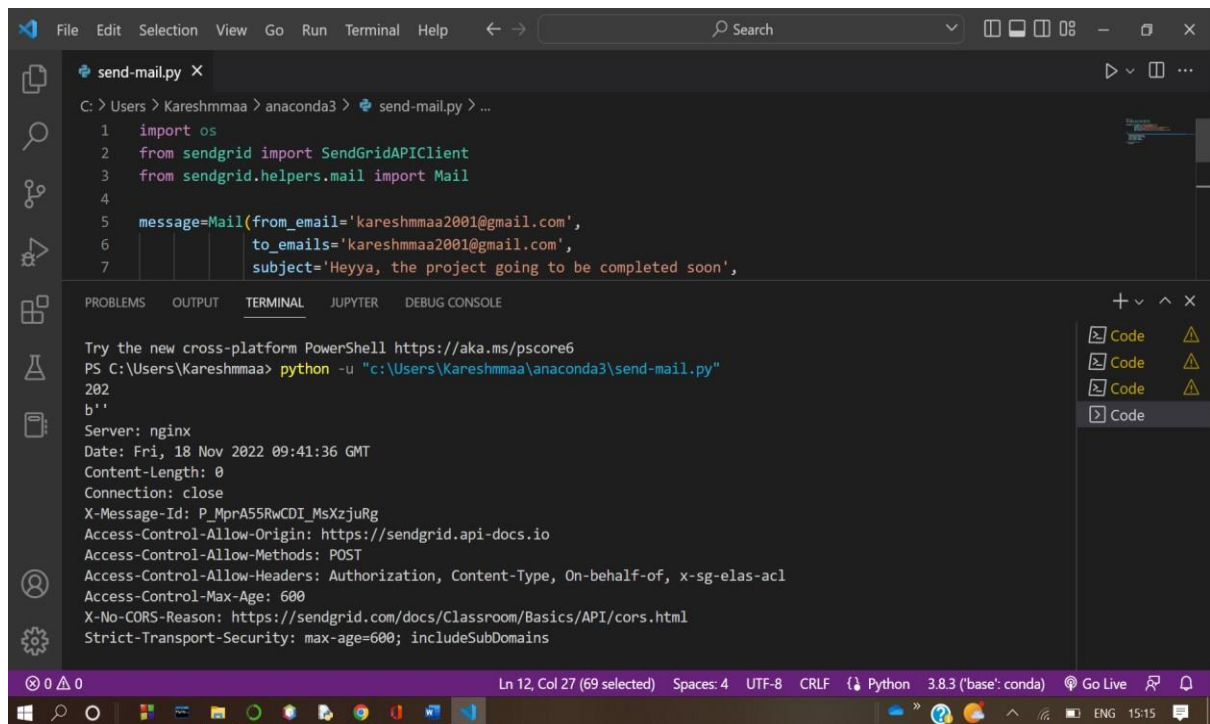


INTEGRATING SENDGRID SERVICE:

Here, SENDGRID_API_KEY must be substituted with unique API Key that is generated when you create an API Key in the SendGrid account.

SendGrid integration with Python code:





```
File Edit Selection View Go Run Terminal Help
C: > Users > Kareshmmaa > anaconda3 > send-mail.py > ...
1 import os
2 from sendgrid import SendGridAPIClient
3 from sendgrid.helpers.mail import Mail
4
5 message=Mail(from_email='kareshmmaa2001@gmail.com',
6               to_emails='kareshmmaa2001@gmail.com',
7               subject='Heyya, the project going to be completed soon',
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657

```

Create UI to interact with application

Registration Page

Login Signup

Full Name

Username

Email Address

Password

Confirm Password

Mobile Number

Signup

Login Page

Login Form

Login Signup

User Name

Password

Login

Not a member? [Signup now](#)

Display items in the Dashboard

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/login'. The page has a blue header with the text 'Hello, jane! (logout)'. Below the header is a table with the following columns: Commodity ID, Commodity, Original Quantity, Current Quantity, Rate, Restocking Date, and Last Modification Date. At the bottom of the page, there are three buttons: 'Add', 'Remove', and 'Update'. The Windows taskbar at the bottom shows the date and time as 14:18 on 18-11-2022.

Commodity ID	Commodity	Original Quantity	Current Quantity	Rate	Restocking Date	Last Modification Date
--------------	-----------	-------------------	------------------	------	-----------------	------------------------

This screenshot shows the same web dashboard as the previous one, but with one data row added to the table. The row contains the following values: Commodity ID 'CX001', Commodity 'Cucumber', Original Quantity '12', Current Quantity '12', Rate '1', Restocking Date '11/17/2022, 7:03:50 PM', and Last Modification Date '-'. The 'Add', 'Remove', and 'Update' buttons remain at the bottom of the page.

Commodity ID	Commodity	Original Quantity	Current Quantity	Rate	Restocking Date	Last Modification Date
CX001	Cucumber	12	12	1	11/17/2022, 7:03:50 PM	-

Adding items

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/login'. The page has a blue header with the text 'Hello, jane! (logout)'. Below the header is a table with the following columns: Commodity ID, Commodity, Original Quantity, Current Quantity, Rate, Restocking Date, and Last Modification Date. At the bottom of the page, there is a form with the following fields: Commodity ID (IPRO), Commodity (Airpods), Original Quantity (5), Current Quantity (100000), and a Save button. To the right of the form are two buttons: Remove and Update. The Windows taskbar at the bottom shows the date and time as 14:19 on 18-11-2022.

Commodity ID	Commodity	Original Quantity	Current Quantity	Rate	Restocking Date	Last Modification Date
--------------	-----------	-------------------	------------------	------	-----------------	------------------------

Add

IPRO Airpods 5 100000 **Save** **Remove** **Update**

The screenshot shows the same web browser window as the previous one, but now the table has one row of data. The 'Add' button is no longer visible, and the 'Remove' and 'Update' buttons are still present. The Windows taskbar at the bottom shows the date and time as 14:19 on 18-11-2022.

Commodity ID	Commodity	Original Quantity	Current Quantity	Rate	Restocking Date	Last Modification Date
IPRO	Airpods	5	5	100000	11/18/2022, 2:19:10 PM	-

Add **Remove** **Update**

Removing Items

The screenshot shows a web application interface. At the top, there's a navigation bar with a "Hello, jane! (logout)" message. Below it is a table with the following data:

Commodity ID	Commodity	Original Quantity	Current Quantity	Rate	Restocking Date	Last Modification Date
CX001	Cucumber	12	12	1	11/17/2022, 7:03:50 PM	-

Below the table, there are three buttons: "Add", "Remove", and "Update". The "Remove" button is highlighted, and a small modal is open with the text "Remove CX001 Remove".

Create IBM DB2 and connect with Node.js

The screenshot shows the IBM Db2 on Cloud console. The "Tables" tab is selected, and the "USERS" table is highlighted. The table definition is shown on the right:

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	30	0
USERNAME	VARCHAR	N	15	0
EMAIL	VARCHAR	N	35	0

The console also shows a "New table" button and a "View data" button. The bottom status bar indicates the time is 22:07 on 18-11-2022.

8. TESTING

The following briefly explain the test coverage and open issues of the Inventory Management system for Retailers project at the time of the release to User Acceptance Testing (UAT).

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	8	7	1	2	18
Duplicate	2	0	2	0	4
External	2	3	1	2	8
Fixed	12	1	5	17	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	16	13	23	76

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	6	0	0	6
Client Application	55	0	0	55
Security	4	0	0	4
Outsource Shipping	3	0	0	3
Exception Reporting	8	0	0	8
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. ADVANTAGES AND DISADVANTAGES

Real-time inventory tracking makes it easier to manage your inventory and guarantees that you have enough stock on hand to complete orders. By avoiding overstocking, you can cut back on the expense and labour-intensive manual processes involved in maintaining it.

The advantages are:

- The advantages of using effective inventory management methods and software include increased agility and cost optimization.
- Reduces the need for huge working capital, increases cash flow, and gives you the money you need to pay employees, produce new products, or engage in any other business activity.
- To save time on inventory forecasting, you can automate your inventory management process.
- You may move data with ease and keep tabs on business activity by using a central data warehouse.
- Complete and correct customer orders in a timely manner.

The disadvantages are:

- The potential for a system crash is among the biggest issues with any computerised system.
- A valuable tool to use when looking for possible financial information or personal information of owners, vendors, or customers is an inventory system connected to point-of-sale devices and accounting.
- It is simple to omit time-consuming physical inventory audits when everything is automated which is necessary too.

10. RESULTS AND CONCLUSION

A manual pen and paper system must be replaced with an inventory management system. Its primary objective is to regulate how the products are moved and stored, with the added advantages of improved security and quicker handling. A software programme called an inventory management system is essential for keeping track of a certain retailer's stock levels. Additionally, it can offer insightful data to sales analytics. In the end, it serves as a company's lifeline because it drives profitability by creating sales. A company's overall success may be significantly impacted by how it manages its inventory.

Thus, a successful “Inventory management System for Retailers” using Python, SendGrid and IBM Cloud Services (IBM DB2, IBM Container registry, IBM Kubernetes) has been developed.

11. FUTURESCOPE:

- Inventory Management System will be seen as a strategic asset by successful businesses rather than a burdensome expense or an unavoidable evil.
- Effective inventory management will depend on collaboration with supply chain partners and a holistic approach to supply chain management.
- Decisions about the deployment of inventory will be significantly impacted by the changing nature of globalization.
- The main drivers for modifying supply chain and inventory strategies will be an increased emphasis on supply chain security and worries about the quality of inventory itself.

12. APPENDIX:

Source Code:

main.py:

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

from database_handlers import *

from ibm_db_dbi import IntegrityError, ProgrammingError

from json import loads, dumps


print('entering flask')

app = Flask(__name__)

app.secret_key = 'temp string. tolerate this for now pls.'

FIELDS = ['cid', 'cname', 'oqty', 'cqty', 'rate', 'date']

@app.route('/')

def login_initial():

    """

    Handles the entryptoint to the website. Acts as a landing page.

    :return: Initial Login page

    """

    return render_template('login.html', err_msg="")

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

def login_check():

    """

    Handles the login process

    :return: Login page

    """
```



```
# capture the username and the password entered by the user and remove trailing and leading whitespaces
```

```
username = request.form['uname'].rstrip().rstrip()
```

```
entered_password = request.form['password']
```

```
# get the expected password corresponding to the username
```

```
expected_password = fetch_password(username)
```

```
if not expected_password:
```

```
    # if expected password is not found in the database, it means no such account is registered yet.
```

```
    return render_template('login.html', err_msg = 'Register an account first!')
```

```
elif entered_password != expected_password:
```

```
    # if the entered and expected password don't match, it means the password entered is wrong.
```

```
    return render_template('login.html', err_msg = 'Wrong username or password')
```

```
else:
```

```
    # if no errors occur, log the user in and create a session.
```

```
    session.update({'username': username})
```

```
    # fetch the inventory items corresponding to the user. It's stored in the form of a JSON as CLOB
```

```
    tableContents = fetch_table_contents(username)
```

```
    # the value returned above will be of the form list[tuple[str/int/float, str/int/float, ...]]
```

```
    if tableContents[0][0] is not None:
```

```
        # if the contents are actually present, convert the JSON to python Dict.
```

```
        tableContents = loads(tableContents[0][0])
```

```
        # print(tableContents['cid'][1])
```

```
    else:
```

```
        # if no content is present, pass an empty list to the template which will be evaluated as 'false' by the
```

```
        # jinja2 template engine.
```

```
        tableContents = []
```

```

        return render_template('dashboard.html', greeting=f'Hello, {username}!',

                               tableContents=tableContents)

    # unreachable code. It's there because I'm paranoid.

    return 'You weren\'t supposed to be here :/'

@app.route('/signup', methods=['POST'])
def signup():
    """
    Handles the signup process.

    :return: Signup Webpage or Dashboard
    """

    # capture the details passed from the form and remove trailing and leading whitespaces.

    fullName = request.form.get('fname').lstrip().rstrip()

    username = request.form.get('uname').lstrip().rstrip()

    email = request.form.get('email').lstrip().rstrip()

    password = request.form.get('password').lstrip().rstrip()

    mobile = request.form.get('mobile').lstrip().rstrip()

    try:

        # try to create account with the given details

        create_account(fullName, username, email, password, mobile)

    except IntegrityError:

        # integrity error means either NOT NULL or UNIQUE constraint has been violated.

        # the former being highly unlikely because of the validation checks in the front end.

        return render_template('login.html', err_msg = 'The account exists already. Please log in.')

    else:

        # if no errors occur, log the user in and create a session.

        # session.update({'username': username})

```

```
# redirect the user to the login procedure, code = 307 specifies to preserve the HTTP method used originally
```

```
# (POST in this case)
```

```
return redirect(url_for('login_check'), code=307)
```

```
# return render_template('dashboard.html', greeting = f'Welcome, {fullName.split()[0]}! ({username})')
```

```
# unreachable code
```

```
return 'You weren\'t supposed to be here. Please go back'
```

```
@app.route('/check-username', methods = ['POST'])
```

```
def check_username():
```

```
    """
```

```
    Checks if the username is already present in the database or not.
```

```
    :return: 'Exists' if present, else return the passed email itself.
```

```
    """
```

```
# get the passed email and remove leading and trailing whitespaces, if any.
```

```
passed_username = request.form.get('username').lstrip().rstrip()
```

```
# check if the email exists or not
```

```
exists = check_username_existence(passed_username)
```

```
if exists:
```

```
    return 'Exists', 403
```

```
return passed_username, 200
```

```
@app.route('/check-email', methods = ['POST'])
```

```
def check_email():
```

```
    """
```

```
    Checks if the email is already present in the database or not.
```

```
    :return: 'Exists' if present, else return the passed email itself.
```

```
    """
```

```

# get the passed email and remove leading and trailing whitespaces, if any.

passed_email = request.form.get('email').lstrip().rstrip()

# check if the email exists or not

exists = check_email_existence(passed_email)

if exists:

    return 'Exists', 403

return passed_email, 200

@app.route('/add-commodity', methods = ['POST'])

def add_commodity():

    # print(request.form)

    details = request.form.to_dict()

    tableContents = fetch_table_contents(session.get('username'))

    if tableContents[0][0] is None:

        tableContents = {field: [] for field in FIELDS}

    else:

        tableContents = loads(tableContents[0][0])

    if details.get('cid') in list(tableContents.values())[0]:

        return 'Such an item already exists.', 403

    for key, value in tableContents.items():

        value.append(details.get(key))

    details_json = dumps(tableContents)

    print(details_json)

    print(details)

    try:

        add_details_to_db(session.get('username'), details_json)

    except Exception as exception:

        print(exception)

```

```
return 'Something went wrong, try again', 400

else:

    return 'Details Added.', 200

    # return redirect(url_for('login_check'), code = 307)

return 'add req rcvd'

if __name__ == '__main__':

    app.static_folder = 'static'

    app.run()
```

db_handlers.py

```
import ibm_db_dbi as db

print('[CONNECTING]')

print('[CONNECTED]')

def create_account(fullName, username, email, password, mobileNumber):

    cursor = conn.cursor()

    SQL = f"INSERT INTO users VALUES(?, ?, ?, ?, ?, NULL)"

    cursor.execute(SQL, [fullName, username, email, password, mobileNumber])

    print('[INSERTED]')

cursor.close()

conn.commit()

def execute_generic_query(sql, *params):

    # return type is: [(1, 2, 3), (4, 5, 6), (7, 8, 9)]

    cursor = conn.cursor()

    if params:

        cursor.execute(sql, params)

    else:

        cursor.execute(sql)
```

```
conn.commit()

try:

    rows = cursor.fetchall()

except db.ProgrammingError:

    rows = None

else:

    if not rows:

        print('Nothing returned by the query')

    else:

        for row in rows:

            print(row)

finally:

    cursor.close()

return rows

def check_username_existence(value):

    SQL = f'SELECT password FROM users WHERE username LIKE \' {value} \'
```

```

    return True

def fetch_password(username):

    SQL = f'SELECT password FROM users WHERE username like \' {username} \'"

    password = execute_generic_query(SQL)

    print(password)

    if not password:

        return password

    return password[0][0]

def fetch_table_contents(username):

    SQL = f'SELECT contents FROM users WHERE username = \' {username} \'"

    tableContents = execute_generic_query(SQL)

    print('table contents are', tableContents)

    return tableContents

def add_details_to_db(username, details):

    SQL = f'UPDATE users SET contents = \' {details} \' WHERE username = \' {username} \'"

    execute_generic_query(SQL)

'''

def view_all_entries():

    execute_generic_query('SELECT * FROM users')

def view_names():

    execute_generic_query('SELECT name FROM users')

def view_usernames():

    execute_generic_query('SELECT username FROM users')

def view_passwords():

    execute_generic_query('SELECT password FROM users')

def view_emails():

    execute_generic_query('SELECT email FROM users')

```

```

def view_mobiles():

    execute_generic_query('SELECT mobile FROM users')

def view_table_layout():

    # You cannot describe a table for some reason.

    return

    # Useless junk below

    cursor = conn.cursor()

    cursor.execute("SELECT * FROM SYSIBM.COLUMNS WHERE TABLE_NAME = 'users'")

    rows = cursor.fetchall()

    for row in rows:

        print(row)

    cursor.close()

'''

def main():

    # view_all_entries()

    # view_table_layout()

    '''

    functions = [

        view_names,

        view_usernames,

        view_emails,

        view_passwords,

        view_mobiles

    ]

    for function in functions:

        print('*'*15)

        function()

```



```
print('[EXIT-MAIN]')

if __name__ == '__main__':

    main()
```

dashboard.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

    #nav {

        width: 100vw;

        border: 1px solid red;

        text-align: center;

    }

    #inventory {

        width: 100%;

        border: 1px solid black;

    }

    #inventory * {

        border: 1px solid black;

    }

    #add-container {

        display: none;

    }
```

```
#update-container {

    display: none;

}

</style>

<title>Dashboard</title>

</head>

<body>

    <nav id="nav">

        <span>{{ greeting }}</span>

    </nav>

    <div id="table-container">

        {% if tableContents %}

            <table id="inventory">

                <tr>

                    <th>Commodity ID</th>

                    <th>Commodity</th>

                    <th>Original Quantity</th>

                    <th>Current Quantity</th>

                    <th>Rate</th>

                    <th>Modification Date</th>

                </tr>

                {% for index in range(tableContents[(tableContents.keys()|list)[0]][length]) %}

                    <tr onclick="update_qty()">

                        {% for attribute in tableContents.keys() %}

                            <td>{{ tableContents[attribute][index] }}</td>

                        {% endfor %}

                    </tr>

                {% endfor %}

            </table>

        {% else %}

            <div>No inventory data found.</div>

        {% endif %}

    </div>

</body>

</html>
```

```

        </tr>

        { % endfor % }

    </table>

    { % else % }

    <p>No content</p>

    { % endif % }

</div>

<div id="modification-container">

    <button class="modification-btn" onclick="reveal_addition_wizard()">Add</button>

    <div id="add-container">

        <input type="text" placeholder="Commodity ID">

        <input type="text" placeholder="Commodity name">

        <input type="text" placeholder="Quantity">

        <input type="text" placeholder="Rate">

        <button onclick="verify_and_insert_to_table()" class="modification-btn">Save</button>

    </div>

    <button class="modification-btn">Remove</button>

    <button lass="modification-btn" onclick="reveal_updatation_wizard()">Update</button>

    <div id="update-container">

        <input type="text" placeholder="Commodity ID">

        <input type="text" placeholder="New Quantity">

        <button onclick="update_qty()" class="modification-btn">Save</button>

    </div>

</div>

</div>

<script>

    const cidFormat = /^[a-zA-Z0-9_]+$/;

    const cnameFormat = /^[a-zA-Z0-9_]+$/;

    const qtyFormat = /^[d]+$/;

```

```
const rateFormat = /^\\d+\\.?\\d*$/;

const addBtn = document.getElementsByClassName('modification-btn')[0];

const saveBtn = document.getElementsByClassName('modification-btn')[1];

const removeBtn = document.getElementsByClassName('modification-btn')[2];

const updateBtn = document.getElementsByClassName('modification-btn')[3];

const table = document.getElementById('inventory');

function reveal_updatation_wizard() {

    document.getElementById('update-container').style.display = 'block';

}

function hide_updatation_wizard() {

    document.getElementById('update-container').style.display = 'none';

}

function reveal_addition_wizard() {

    document.getElementById('add-container').style.display = 'block';

}

function hide_addition_wizard() {

    document.getElementById('add-container').style.display = 'none';

}

function verify_and_insert_to_table() {

    let allGood = true;

    let errString = 'Enter ';

    let cid = document.getElementById('add-container').children[0].value.trim();

    let cname = document.getElementById('add-container').children[1].value.trim();

    let qty = document.getElementById('add-container').children[2].value.trim();

    let rate = document.getElementById('add-container').children[3].value.trim();

    if(!cidFormat.test(cid)) { errString += 'Commodity ID, '; allGood = false; }

    if(!cnameFormat.test(cname)) { errString += 'Commodity name, '; allGood = false; }
```

```
if(!qtyFormat.test(qty)) { errString += 'Quantity, '; allGood = false; }

if(!rateFormat.test(rate)) { errString += 'Rate '; allGood = false; }

errString += 'correctly.';

if(!allGood) alert(errString);

else {

    let dateTime = new Date().toLocaleString();

    let array = [cid, cname, qty, qty, rate, dateTime];

    addBtn.disabled = true;

    removeBtn.disabled = true;

    saveBtn.disabled = true;

    updateBtn.disabled = true;

    xhr = new XMLHttpRequest();

    xhr.open("POST", "/add-commodity", true);

    xhr.onreadystatechange = () => {

        if (xhr.readyState === XMLHttpRequest.DONE) {

            alert(xhr.responseText);

            addBtn.disabled = false;

            removeBtn.disabled = false;

            saveBtn.disabled = false;

            updateBtn.disabled = false;

            if(xhr.status === 200) {

                let row = table.insertRow(-1);

                for(let cell = 0; cell < 6; ++cell) {

                    let newCell = row.insertCell(cell);

                    newCell.innerHTML = array[cell];

                }

            }

        }

    }

}
```

```

        hide_addition_wizard();

    }

}

xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');

xhr.send(`cid=${cid}&cname=${cname}&oqty=${qty}&cqty=${qty}&rate=${rate}&date=${dateTime}`);

}

}

function update_qty() {

    let allGood = true;

    let targetCommodity = document.getElementById('update-container').children[0].value.trim();

    let newQty = document.getElementById('update-container').children[1].value.trim();

    if(!cidFormat.test(targetCommodity) || !qtyFormat.test(newQty)) allGood = false;

    if(!allGood) alert('Detail(s) entered in wrong format.');
```

```

    else {

        }

    }

}

</script>

</body>

</html>

```

login.html:

```

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

    <meta charset="utf-8">

    <title>Login and Registration</title>

```

```
<link rel="stylesheet" href="{ { url_for('static', filename='login_style.css') } }">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
</head>
```

```
<body>
```

```
<div class="wrapper">
```

```
<div class="title-text">
```

```
<div class="title login">
```

```
    Login Form
```

```
</div>
```

```
<div class="title signup">
```

```
    Signup Form
```

```
</div>
```

```
</div>
```

```
<div class="form-container">
```

```
<div class="slide-controls">
```

```
<input type="radio" name="slide" id="login" checked>
```

```
<input type="radio" name="slide" id="signup">
```

```
<label for="login" class="slide login">Login</label>
```

```
<label for="signup" class="slide signup">Signup</label>
```

```
<div class="slider-tab"></div>
```

```
</div>
```

```
<div class="form-inner">
```

```
<form action="/login" class="login" method="post" name="login-form">
```

```
<div class="field">
```

```
<input type="text" placeholder="User Name" required name="uname">
```

```
</div>
```

```
<div class="field">
```

```
<input type="password" placeholder="Password" required name="password">
```

```
</div>
```

```
<div class="pass-link">
```

```
<a href="#">Forgot password?</a>
```

```
</div>
```

```
<div style="color: red; text-align: center;">
```

```
{ { err_msg } }
```

```
</div>
```

```
<div class="field btn">
```

```
<div class="btn-layer"></div>
```

```
<input type="submit" value="Login">
```

```
</div>
```

```
<div class="signup-link">
```

```
Not a member? <a href="">Signup now</a>
```

```
</div>
```

```
</form>
```

```
<form action="/signup" class="signup" method="post" name="signup-form">
```

```
<div class="field">
```

```
<input type="text" placeholder="Full Name" required name="fname"
onfocusout="check_name()">
```

```
</div>
```

```
<div class="field">
```

```
<input type="text" placeholder="Username" required name="uname"
onfocusout="check_username()">
```

```
</div>
```

```
<div class="field">
```

```
<input type="text" placeholder="Email Address" required name="email"
```



```
onfocusout="check_email()">

    </div>

    <div class="field">

        <input type="password" placeholder="Password" required name="password"

            onfocusout="check_password()">

    </div>

    <div class="field">

        <input type="password" placeholder="Confirm Password" required name="password-cnf"

            onfocusout="check_password()">

    </div>

    <div class="field">

        <input type="tel" placeholder="Mobile Number" required name="mobile"

            onfocusout="check_mobile()">

    </div>

    <div class="field btn">

        <div class="btn-layer"></div>

        <input type="submit" value="Signup" disabled name="signup-btn" style="cursor: not-
allowed;"

            onclick="check_all()">

    </div>

</form>

</div>

</div>

</div>

<script>

    var flag1 = false, flag2 = false, flag3 = false, flag4 = false, flag5 = false;

    const loginText = document.querySelector(".title-text .login");
```

```
const loginForm = document.querySelector("form.login");

const loginBtn = document.querySelector("label.login");

const signupBtn = document.querySelector("label.signup");

const signupLink = document.querySelector("form .signup-link a");

const nameField = document.forms['signup-form']['fname'];

const usernameField = document.forms['signup-form']['uname'];

const emailField = document.forms['signup-form']['email'];

const passwordField = document.forms['signup-form']['password'];

const password_cnfField = document.forms['signup-form']['password-cnf'];

const mobileField = document.forms['signup-form']['mobile'];

const signupBtnActual = document.forms['signup-form']['signup-btn'];

signupBtn.onclick = () => {

    loginForm.style.marginLeft = "-50%";

    loginText.style.marginLeft = "-50%";

});

loginBtn.onclick = () => {

    loginForm.style.marginLeft = "0%";

    loginText.style.marginLeft = "0%";

});

signupLink.onclick = () => {

    signupBtn.click();

    return false;

});

function check_all() {

    console.log('check_all');

    check_name();

    check_username();
```

```
check_password();

check_email();

check_mobile();

if (check_flags()) {

    enable_signup_btn();

}

else disable_signup_btn();

}

function check_flags() {

    return flag1 && flag2 && flag3 && flag4 && flag5;

}

function enable_signup_btn() {

    signupBtnActual.disabled = false;

    signupBtnActual.style.cursor = 'pointer';

}

function disable_signup_btn() {

    signupBtnActual.disabled = true;

    signupBtnActual.style.cursor = 'not-allowed';

}

function check_username() {

    if (usernameField.value === "") return;

    var xhr = new XMLHttpRequest();

    xhr.open("POST", "/check-username", true);

    let usernameformat = /^[a-z][a-z0-9_\.]+$/;

    let username = usernameField.value.trim();

    let valid = usernameformat.test(username);

    if (valid) {
```

```
xhr.onreadystatechange = () => {

    if (xhr.readyState === XMLHttpRequest.DONE && xhr.status === 403) {

        usernameField.value = 'Username Already Exists!';

        usernameField.style.backgroundColor = '#FFC6B9';

        valid = false;

    }

}

xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');

xhr.send("username=" + username);

}

if (!valid)

    usernameField.style.backgroundColor = '#FFC6B9';

else

    usernameField.style.backgroundColor = '#C5FFB9';

flag2 = valid;

if (check_flags()) enable_signup_btn();

else disable_signup_btn(); }

function check_email() {

    if (emailField.value === "") return;

    var xhr = new XMLHttpRequest();

    let mailformat = /^[w+([\\.-]?w+)*@[w+([\\.-]?w+)*([\\w]{2,3})+}$/;

    let email = emailField.value.trim();

    let valid = mailformat.test(email);

    if (valid) {

        xhr.open("POST", "/check-email", true);

        xhr.onreadystatechange = () => {

            if (xhr.readyState === XMLHttpRequest.DONE && xhr.status === 403) {
```

```
emailField.value = 'Email Already Exists!';
```

```
    valid = false;
```

```
    emailField.style.backgroundColor = '#FFC6B9';
```

```
    }
```

```
  }
```

```
  xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
```

```
  xhr.send("email=" + email);
```

```
}
```

```
if (!valid)
```

```
    emailField.style.backgroundColor = '#FFC6B9';
```

```
else
```

```
    emailField.style.backgroundColor = '#C5FFB9';
```

```
flag3 = valid;
```

```
if (check_flags()) enable_signup_btn();
```

```
else disable_signup_btn();
```

```
}
```

```
function check_name() {
```

```
    let name = nameField.value.trim();
```

```
    if (name === "") return;
```

```
    let    nameformat    =    /^(?:(?!(^0-9_!i?÷?¿/\+=@#$$%^&*(){}|~<>;:[\]",\-.\\s])){1,}([",\-.\\.)]{0,1}){2,}((?!(^0-9_!i?÷?¿/\+=@#$$%^&*(){}|~<>;:[\]",\-.\\s])){1,})*((([+]){0,1}(((?!(^0-9_!i?÷?¿¿/\+=@#$$%^&*(){}|~<>;:[\]",\-.\\s])){1,})("[\`-,\\.]) {0,1}){2,}(((?!(^0-9_!i?÷?¿¿/\+=@#$$%^&*(){}|~<>;:[\]",\-.\\s])){2,})?)*)$/;
```

```
    let valid = nameformat.test(name);
```

```
    if (valid)
```

```
        nameField.style.backgroundColor = '#C5FFB9';
```

```
    else
```

```
        nameField.style.backgroundColor = '#FFC6B9';
```

```
flag1 = valid;

    if (check_flags()) enable_signup_btn();

    else disable_signup_btn();

}

function check_mobile() {

    let mobile = mobileField.value.trim();

    if (mobile === "") return;

    let mobileformat = /^d{10}$/;

    let valid = mobileformat.test(mobile);

    if (valid)

        mobileField.style.backgroundColor = '#C5FFB9';

    else

        mobileField.style.backgroundColor = '#FFC6B9';

    flag5 = valid;

    if (check_flags()) enable_signup_btn();

    else disable_signup_btn();

}

function check_password() {

    if (passwordField.value === "" && password_cnfField.value === "") return;

    let valid = (passwordField.value === password_cnfField.value);

    if (!valid) {

        password_cnfField.style.backgroundColor = '#FFC6B9';

        passwordField.style.backgroundColor = '#FFC6B9';

    }

    else {

        password_cnfField.style.backgroundColor = '#C5FFB9';

        passwordField.style.backgroundColor = '#C5FFB9';

    }

}
```

```
}

    flag4 = valid;

    if (check_flags()) enable_signup_btn();

    else disable_signup_btn();

}

</script>

</body>

</html>
```

login_style.css:

```
@import url('https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display=swap');

*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    font-family: 'Poppins', sans-serif;

}

html,body{

    display: grid;

    height: 100%;

    width: 100%;

    place-items: center;

    background: -webkit-linear-gradient(top, #eacda3, #fff);

}

::selection{

    background: #d6ae7b;

    color: #fff;

}
```

```
.wrapper{

  overflow: hidden;

  max-width: 390px;

  background: #fff;

  padding: 30px;

  border-radius: 5px;

  box-shadow: 0px 15px 20px rgba(0,0,0,0.1);

}

.wrapper .title-text{

  display: flex;

  width: 200%;

}

.wrapper .title{

  width: 50%;

  font-size: 35px;

  font-weight: 600;

  text-align: center;

  transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);

}

.wrapper .slide-controls{

  position: relative;

  display: flex;

  height: 50px;

  width: 100%;

  overflow: hidden;

  margin: 30px 0 10px 0;

  justify-content: space-between;
```



```
border: 1px solid lightgrey;

border-radius: 5px;

}

.slide-controls .slide{

height: 100%;

width: 100%;

color: #fff;

font-size: 18px;

font-weight: 500;

text-align: center;

line-height: 48px;

cursor: pointer;

z-index: 1;

transition: all 0.6s ease;

}

.slide-controls label.signup{

color: #000;

}

.slide-controls .slider-tab{

position: absolute;

height: 100%;

width: 50%;

left: 0;

z-index: 0;

border-radius: 5px;

background: -webkit-linear-gradient(left, #eacda3, #d6ae7b);

transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);
```

```
}

input[type="radio"]{

    display: none;

}

#signup:checked ~ .slider-tab{

    left: 50%;

}

#signup:checked ~ label.signup{

    color: #fff;

    cursor: default;

    user-select: none;

}

#signup:checked ~ label.login{

    color: #000;

}

#login:checked ~ label.signup{

    color: #000;

}

#login:checked ~ label.login{

    cursor: default;

    user-select: none;

}

.wrapper .form-container{

    width: 100%;

    overflow: hidden;

}

.form-container .form-inner{

    display: flex;
```

```
width: 200%;

}

.form-container .form-inner form{

width: 50%;

transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);

}

.form-inner form .field{

height: 50px;

width: 100%;

margin-top: 20px;

}

.form-inner form .field input{

height: 100%;

width: 100%;

outline: none;

padding-left: 15px;

border-radius: 5px;

border: 1px solid lightgrey;

border-bottom-width: 2px;

font-size: 17px;

transition: all 0.3s ease;

}

.form-inner form .field input:focus{

border-color: #eacda3;

/* box-shadow: inset 0 0 3px #fb6aae; */

}

.form-inner form .field input::placeholder{
```

```
color: #999;

transition: all 0.3s ease;

}

form .field input:focus::placeholder{

color: #b3b3b3;

}

.form-inner form .pass-link{

margin-top: 5px;

}

.form-inner form .signup-link{

text-align: center;

margin-top: 30px;

}

.form-inner form .pass-link a,

.form-inner form .signup-link a{

color: #eacda3;

text-decoration: none;

}

.form-inner form .pass-link a:hover,

.form-inner form .signup-link a:hover{

text-decoration: underline;}

form .btn{

height: 50px;

width: 100%;

border-radius: 5px;

position: relative;

overflow: hidden;

}
```

```
form .btn .btn-layer{

    height: 100%;

    width: 300%;

    position: absolute;

    left: -100%;

    background: -webkit-linear-gradient(right, #d6ae7b, #eacda3, #d6ae7b, #eacda3);

    border-radius: 5px;

    transition: all 0.4s ease;;}

form .btn:hover .btn-layer{

    left: 0;}

form .btn input[type="submit"]{

    height: 100%;

    width: 100%;

    z-index: 1;

    position: relative;

    background: none;

    border: none;

    color: #fff;

    padding-left: 0;

    border-radius: 5px;

    font-size: 20px;

    font-weight: 500;

    cursor: pointer;}
```

GitHub and Demo Link:

<https://github.com/IBM-EPBL/IBM-Project-38951-1660387276>