

CONTENTS

1. INTRODUCTION

- 1.1Project Overview
- 1.2Purpose

2. LITERATURE SURVEY

- 2.1Existing problem
- 2.2References
- 2.3Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- 3.1Empathy Map Canvas
- 3.2Ideation & Brainstorming
- 3.3Proposed Solution
- 3.4Problem Solution fit

4. REQUIREMENT ANALYSIS

- 4.1Functional requirement
- 4.2Non-Functional requirements

5. PROJECT DESIGN

- 5.1Data Flow Diagrams
- 5.2Solution & Technical Architecture
- 5.3User Stories

6. PROJECT PLANNING & SCHEDULING

- 6.1Sprint Planning & Estimation
- 6.2Sprint Delivery Schedule
- 6.3Reports from JIRA

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

- 7.1Feature 1
- 7.2Feature 2

8. TESTING

- 8.1Test Cases
- 8.2User Acceptance Testing

9. RESULTS

- 9.1Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

- Source Code, GitHub
- Project Demo Link

1.INTRODUCTION

Inventory is the supply of raw materials, partially finished goods called work-in-progress and finished goods, an organization maintains to meet its operational needs. It represents a sizeable investment and a potential source of waste that needs to be carefully controlled. Inventory is defined as a stock of goods that is maintained by a business in anticipation of some future demand. The quantity to which inventory must fall to signal that an order must be placed to replenish an item.

Using an extension of a standard inventory-dependent demand model provides a convenient characterization of products that require early replenishment. The optimal cycle time is largely governed by the conventional trade-off between ordering and holding costs, whereas the reorder point relates to a promotions-oriented cost-benefit perspective. The optimal policy yields significantly higher profits than cost-based inventory policies, underscoring the importance of profit-driven inventory management. To work towards perfect order metrics, there must be aggressive inventory management, restructuring supply chain operations, and updating standards to the perfect standard. When updating the metrics, this would include the cases shipped vs. the orders on-time delivery, data synchronization, damages and unusable products, days in supply, the ordering time cycle, and shelf level of service.

1.1Project Overview

The Inventory Management System is an application designed to allow the supermarket staff to create, maintain and view the contents and value of its inventory of items in a categorized way. It also aims to analyze the position of the supermarket in the market and help it know what items to order in what quantity by producing graphs depicting the sale of different items on the different basis such as monthly, yearly, brand type etc.

The Inventory System is to facilitate our customers tracking their products as and when they are transported from the vendor to the warehouse and from the warehouse to the retail location to the customers. It is necessary to keep our resources safe and protected. In order to implement security in the application it would be done by implementing encryption, keeping a secure session base password, implementing two-level authentications, observing system logs and security faults, analyzing network flow using Wireshark, implementing Wireshark, preventing the

application validation from unnecessary inputs, session management, session hijacking, hacking, cross-site scripting and implementing code to prevent from SQL injection and many more.

1.2Purpose

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

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

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

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

2.LITURATURE SURVERY

Finger S and Dixon (1990) says that formal design research seems to have begun in the 1960's, with so-called "first generation" models used to attempt to find generic optimization routines that could be applied to any type of problem.

The formal design research seems to have begun in the 1960's, with so-called "first generation" models used to attempt to find generic optimization routines

that could be applied to any type of problem. The architectural models tended to include cognitive processes, while engineering models attempted to define stages in the design process.

Desmet and Hekkert (2007) says that surprisingly little is reported on the pragmatic influence of project stakeholders on industrial designers' selection of product materials and manufacturing processes.

This paper reports on a descriptive scoping study that revealed these influences as critical in making effective selection decisions. Using interview and case study methods, the study elicited the professional practices of industrial designers.

Kim and Kang (2008), identifies the critical factors of cross-functional cooperation for design teams in new product development.

Davis (2008) identifies the pressures on knowledge generation exerted by the shift from a mechanical, object centered paradigm for design practice to one characterized by systems that: evolve and behave organically; transfer control from designers to users or participants; emphasize the importance of community.

Heskett(2009) examines the influence of major economic theories in shaping views of what constitutes value as created by design system.

Its focus on markets and prices as set by market forces are believed to solve all problems if left free from government interference. The implosion of this system and its emphasis on unrestricted individualism is a crisis of theory as well as practice.

Vendanand Sakthidhasan (2010) addresses the application of lean manufacturing concepts to the continuous production sector with a focus on the motor manufacturing industry.

The goal of this research is to investigate how lean manufacturing tools can be adapted from the discrete to the continuous manufacturing environment. The

application of lean manufacturing concepts to the continuous production sector with a focus on the motor manufacturing industry.

Ahmad and Khaldoun (2011) research aims at presenting a realistic approach for resolving the multiple rate of return (MROR) problem.

The key advantage of the proposed approach is that it reflects real life opportunities and its decisions are consistent with worth methods as well as with other approaches.

Pastore and Martin (2012) study was to examine students' perceptions of designing and developing mobile based instructions by interviewing and surveying of graduate students.

Norman E (2012) discusses, while existing factors identified in the literature were found to be present in the context of today's design program, the critical perspective of this study recontextualized these factors, along with the identification of new or underrepresented factors.

Gray (2013) explains the experiences of six first-year design students were examined as they evolved in their conceptions of design.

Halaweh M (2013) aims to define and conceptualize the characteristics of ET. These characteristics are uncertainty, network effect, unseen social and ethical concerns, cost, limitation to particular countries, and a lack of investigation and research.

Leber (2014) reports the results of a survey on the use of innovation management techniques with the potential to improve effectiveness of new product development, and customer satisfaction.

Dou (2014) paper is committed to design a logistics industry development policy model based on system dynamic to simulate the policy measures which promote region economic and logistics efficiency.

2.1 EXISTING PROBLEM

- Lack of Inventory Visibility. ...
- Inefficient Inventory Management Process or Software
- Tracking Obsolete Material. ...
- Identifying Incorrectly Located Materials. ...
- Keeping up with Overstocks. ...
- Managing Inventory Waste & Defects. ...
- Lack of Centralized Inventory Hub. ...
- Changing Demand.

2.2 REFERENCE

1. Afentakis, P., Gavish, B., Karmarkar, U.: Computationally efficient optimal solutions to the lot-sizing problem in multistage assembly systems. *Management Science* 30, 222–239 (1984)
2. Aggarwal, A., Park, J.: Improved algorithms for economic lotsize problems. *Operations Research* 41, 549–571 (1993)
3. Aggarwal, P.K., Moinzadeh, K.: Order expedition in multiechelon production/distribution systems. *IIE Transactions* 26(2), 86–96 (1994)
4. Aggarwal, S.: A review of current inventory theory and its applications. *International Journal of Production Research* 12, 443–472 (1974)
5. Agrawal, V., Cohen, M.A., Zheng, Y.S.: Service parts logistics: A benchmark analysis. *IIE Transactions, Special Issue on Supply Chain Coordination and Integration* 29(8), 627–639 (1997)
6. Tripp et al., R.: A decision support system for assessing and controlling the effectiveness of multi-echelon logistics actions. *Interfaces* 21(4), 11–25 (1991)
7. Albright, S.C.: An approximation to the stationary distribution of a multi-echelon repairable item inventory system. *Naval Research Logistics* 36, 179–195 (1989)
8. Albright, S.C., Soni, A.: Markovian multi-echelon repairable inventory system. *Naval Research Logistics Quarterly* 35, 49–61 (1988)
9. Alfredsson, P., Verrijdt, J.: Modeling emergency supply flexibility in a two-echelon inventory system. *Management Science* 45(10), 1416–1431 (1999)

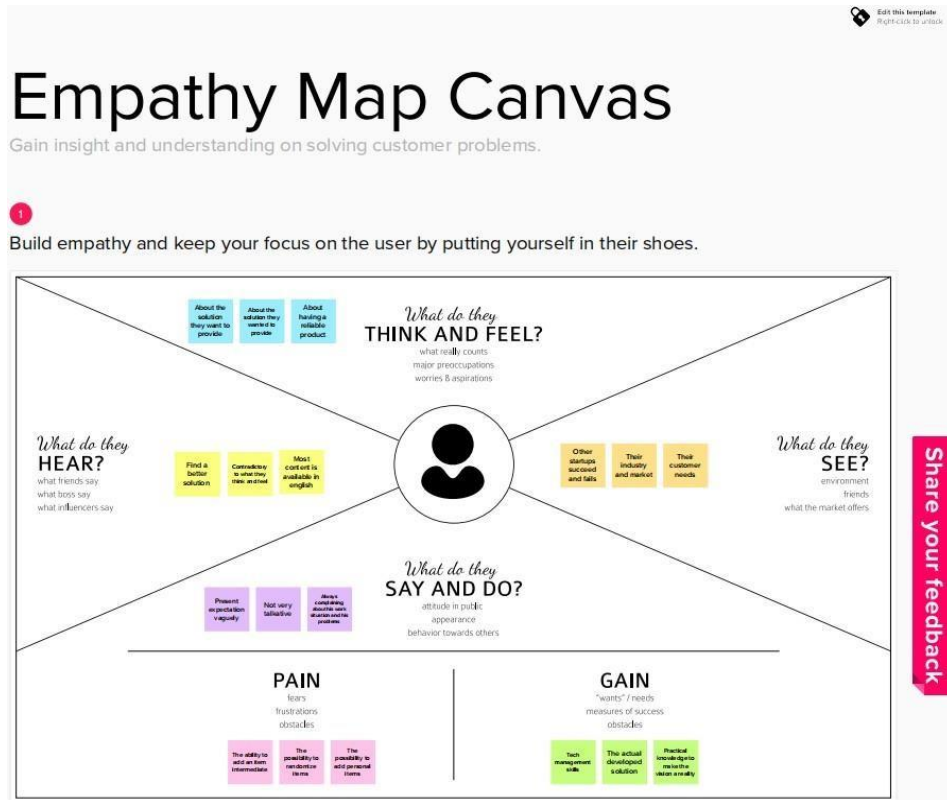
10. Allen, S.G.: Redistribution of total stock over several user locations. Naval Research Logistics Quarterly 5, 51–59 (1958)

2.3 PROBLEM STATEMENT DEFINITION

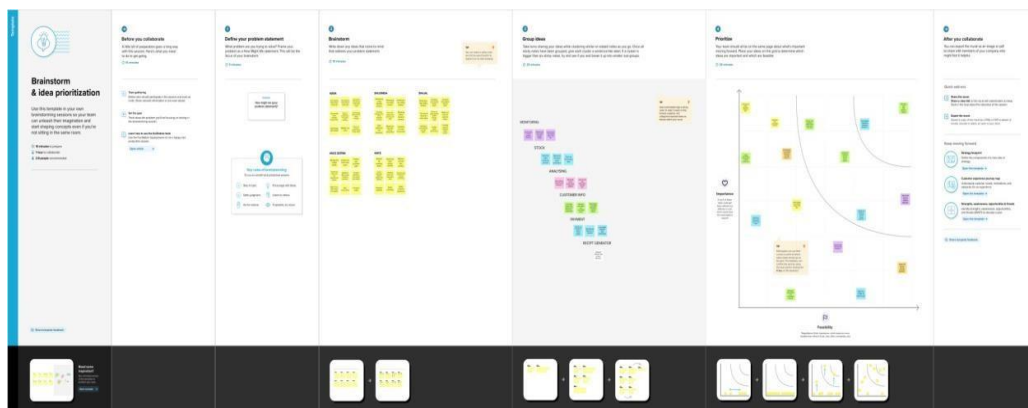
Problem Statement	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Retailer	Organize stock, manage purpose and sale detail	Lot of time and space consuming	I need to organize the stock, purchase detail, sale details and maintain the inventory model	Better relief and happy
PS-2	Retailer	Monitor stock levels & avoid out of stock situations	Is time consuming, redundant and vulnerable to errors	Using manual inventory tracking procedures across different software and spreadsheets	Frustrated
PS-3	Warehouse supervisor	Track items in real time as they are received, stored, picked, packed and shipped	Advises on inventory replenishment	Guides inventory receiving and put away	Tired & stressed

3.IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Retail store stock inventory analytics is implemented to analyse the historical sales data of a Brazilian top retailer. By deeply understanding the dataset, identifying pattern, relationships and connection using IBM cognos analytics and building visualizations of stocks inventory to create meaningful dashboards. The final dynamic dashboard helps retailers by providing detailed product listing, easy categorization, inventory reports, satisfying customer needs and meet variation in product demand.
2	Idea / Solution description	This project is aimed at developing a desktopbased application named Inventory Management System for managing the inventory system of any organization. The Inventory Management System (IMS) refers to the system and processes to manage the stockof organization with the

		<p>involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update the inventory based on the sales details, generate sales and inventory report daily or weekly based. This project is categorized individual aspects for the sales and inventory management system. In this system we are solving different problem affecting to direct sales management and purchase management. Inventory Management System is important to ensure quality control in businesses that handle transactions resolving around consumer goods. Without proper inventory control, a large retail store may run out of stock on an important item. A good inventory management system will alert the wholesaler when it is time to record. Inventory Management System is also on important means of automatically tracking large shipment. An</p>
--	--	---

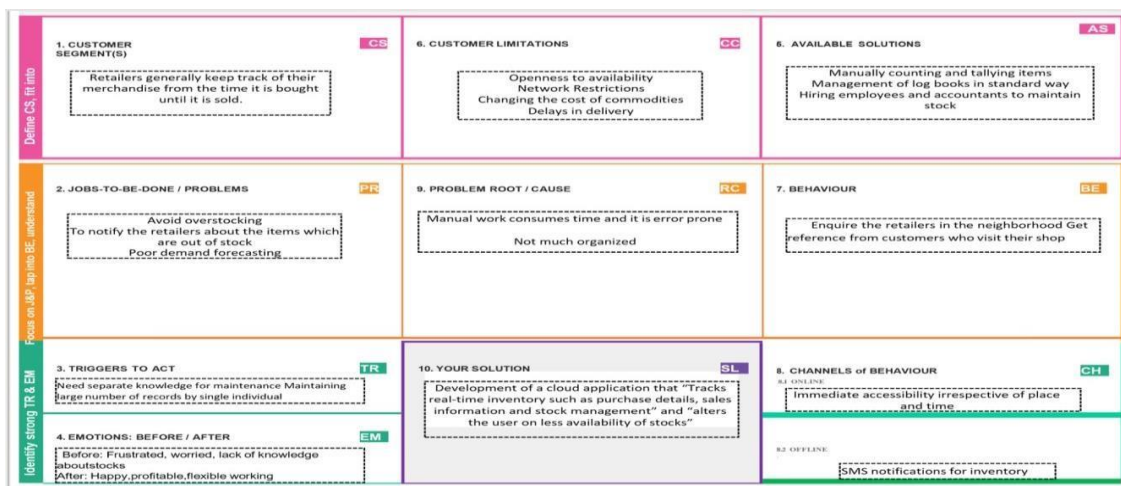
		<p>automated Inventory Management System helps to minimize the</p>
--	--	--

		errors while recording the stock.
3	Novelty / Uniqueness	<p>With solid inventory management, you know what's in stock and order only the amount of inventory you need to meet demand. Inventory management helps track what's in stock and what's on backorder, so you don't oversell products. Stock costs money until it sells. Carrying costs include storage handling and transportation fees, insurance and employee salaries.</p> <p>Inventory is also at risk of theft, loss from natural disasters or obsolescence. Inventory management also provides insights about which products sell and in what volume. Use that knowledge as leverage to negotiate better prices and terms with suppliers. Good inventory management solutions save time that could be spent on other activities. A better understanding of both availability and demand leads to higher inventory.</p>
4	Social Impact / Customer Satisfaction	Customer satisfaction entirely depends on the effective supply chain management which is not

		<p>an easy task. In past companies used to hold large inventories to avoid shortage of inventories and to increase the customer satisfaction however it has been observed that this satisfaction is subjective to person to person, though effective inventory management is the only way to increase customer satisfaction. This inventory caused manufacturers to stockpile large amounts of raw materials, work in process, and finished goods. The extra finished goods would be to protect them from going out of stock.</p> <p>The study of customer satisfaction has shown that there could be a disproportional relationship between cause and effect, or between a factor and its consequence on the organization.</p>
--	--	---

5	Business Model (Revenue Model)	Business technology is changing and becoming more efficient every day. Using today's instant communications and being able to track changes and metrics continually gives business owners the tools
		they need to make their business scalable. Businesses that embrace technology and use automated systems wherever they can are the ones that are in the best position to survive any kind of disaster or economic downturn.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login with username Login with Email Login with password
FR-4	Dashboard	View product availability, name of the product, stock keep unit, brand, retail price, product category, lot number, expiration date, vendor information, wholesale cost, etc.,
FR-5	Identification of the stock location	Provide number label for - shelf, rack and boxes
FR-6	Periodical stock checking	Automate the tracking of stock count
FR-7	Purchase management and Forecasting	Order review and placement, avoid risk stock, review product, priorities purchases based on an item's profitability, popularity, and lead time.
FR-8	Returns Management System	Examine for flaws or damage and return to the vendor if necessary. Add it to inventory counts if it is sellable.
FR-9	Markdown and promotion	Display product savings, Keep enough inventory on hand to satisfy demand.
FR-10	Calculating the death stock	Return the stock to the vendor for credits

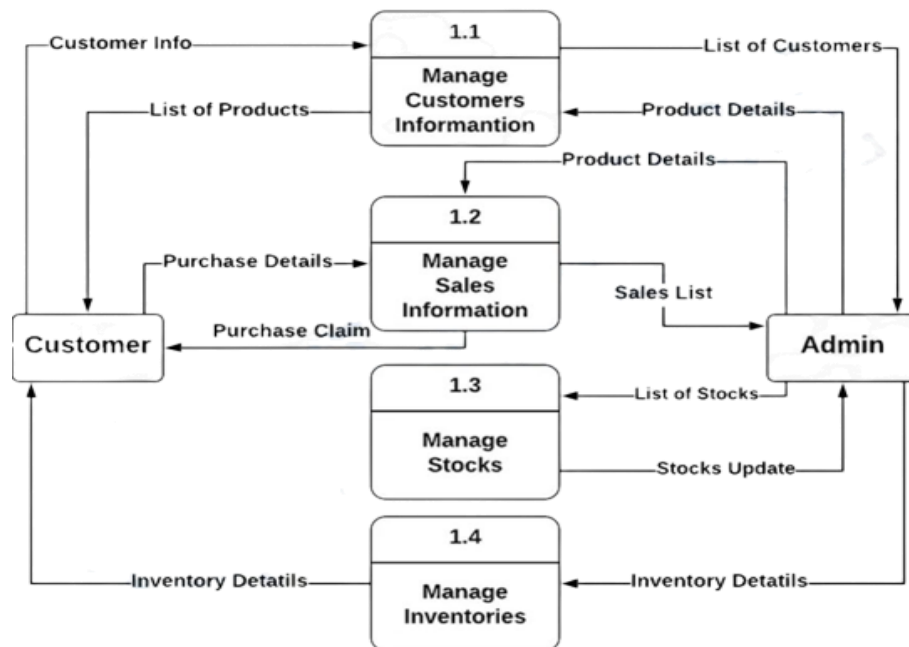
4.2 NON- FUNCTIONAL REQUIREEMENT

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The success of deploying an inventory monitoring system in your company depends on usability. The system must be user-friendly and straightforward in the manner it presents all pertinent data and linkages, and its menus must include buttons that are simple to grasp. The software is not worthwhile if training takes hours for your workforce. Remember to pick a solution that makes inventory management simple. This variant is compatible with desktop browsers.
NFR-2	Security	It is the method for making sure that kept goods are safe and under the best management control. It is crucial for effective warehouse management because a warehouse's productivity and safety determines how well a firm performs. In this case, only authorized people with their username and password can access the system.
NFR-3	Reliability	The user must constantly receive accurate inventory status from the system. By routinely comparing the real levels to the levels shown in the system, any errors are fixed.

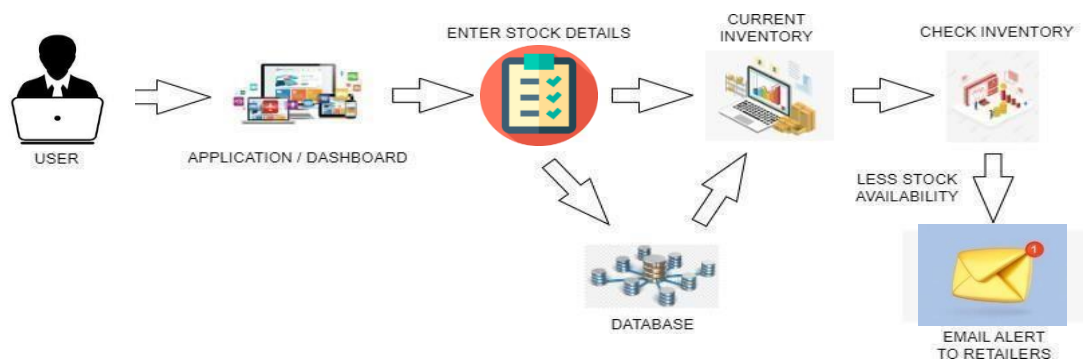
NFR-4	Performance	Every time a user requests a process, the system must successfully complete the tasks like updating the stocks in the database, adding new stocks, and deleting it. Every time the system is turned on, all of its features must be accessible to the customer. The system's calculations must adhere to the standards established by the customer and shouldn't change until the customer specifically requests a change.
NFR-5	Availability	Only the organization's admin will have access to the software, and it is he who will record information of the product as well as the customer details. A single item's inventory availability determines whether it is accessible for customer orders. The admin can also add, remove or update the stock and stock details respectively.
NFR-6	Scalability	The business will become considerably more scalable with the use of an automated inventory management system for inventory tracking, enabling it to capitalize on rising sales and maintain steady growth.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

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	I can register & access the dashboard with	Low	Sprint - 2

			through Facebook	Facebook Login		
		USN-4	As a user, I can register for the application through Gmail	I can register and access the dashboard with Gmail login	Medium	Sprint -1

	Login	USN-5	As a user, I can log into the application by entering email & password	I can login to the application with verified Email and password	High	Sprint -1
	Dashboard	USN-6	As a user, I can view the available product list and the inventory data.	I can view the stock availability status	High	Sprint -2
		USN-7	As a user, I can view the order and track the shipping status	I can view the order and shipping status after making a purchase	Medium	Sprint -3
Customer (Web user)	Registration	USN-8	As a user, I can register for the application by entering	I can access my account / dashboard	High	Sprint -1

			my email, password, and confirming my password.			
--	--	--	---	--	--	--

		USN-9	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-10	I can register & access the dashboard with Facebook Login	I can register & access the dashboard with Facebook Login	Low	Sprint -2
		USN-11	As a user, I can register for the application through Gmail	I can register and access the dashboard with Gmail login	Medium	Sprint -1
	Login	USN-12	As a user, I can log into the application by entering email & password	I can login to the application with verified Email and password	High	Sprint -1
	Dashboard	USN-13	As a user, I can view the	I can view the stock	High	Sprint -2
			available product list and the inventory data.	availability status		

		USN-14	As a user, I can view the order and track the shipping status	I can view the order and shipping status after making a purchase	Medium	Sprint-3
Customer Care Executive	Chat bot	USN-15	As a customer care I can view the complaints on chat bot and assist the users	I can view the complaints on the chat bot and assist the user with queries.	Medium	
Administrator	Alerts	USN-16	As an administrator, I would handle user registrations and maintenance of accounts	I can take care of registrations and maintenance of accounts	High	Sprint-3
		USN-17	As an administrator, I can refill the stock on receiving the alerts	I can refill stock if there's an alert	High	Sprint-3

6. Project planning and scheduling

6.1 sprint planning and estimation

Setting up the application environment	M-01	Setting up the resources needed in the local machine
Integrating send grid service	M-02	To send emails from the application, we need to integrate the SendGrid Service.
Deployment of the app in IBM Cloud	M-03	Containerize a Flask application by using Docker and deploy it to the IBM Cloud Kubernetes Service
Ideation Phase	M-04	Collecting information by referring to previous research on a topic and Prepare Literature Survey on the selected Project and Information Gathering, empathy map and ideation
Project Design Phase - I	M-05	Prepare the proposed solution, the problemsolution fit, and the Solution Architecture.
		Create a customer journey, functional
Project Design Phase - II	M-06	requirements, a data flow diagram, and a technology architecture

Project Planning Phase	M-07	Make a list of milestones, an activity list, and a sprint delivery plan.
Project Development Phase	M-08	Develop and submit Sprint 1, Sprint 2, Sprint 3 and Sprint 4



Activity Number	Activity	Sub Activity	Assigned To	Status
1.	Setting up Application Environment	<ul style="list-style-type: none"> ● Create Flask Project ● Create IBM Cloud Account ● Install IBM Cloud CLI ● Docker CLI Installation ● Create An Account In Sendgrid 	ASISA A, ANCE DEFRIN P, SHAJAL S, SHIJONIDA J V, ANCE C	Review
2.	Implementing Web Application	<ul style="list-style-type: none"> ● Create UI To Interact With Application 	ASISA A, ANCE DEFRIN P, SHAJAL S, SHIJONIDA J V, ANCE C	In-Progress
3.	Integrating SendGrid Service	<ul style="list-style-type: none"> ● SendGrid Integration With Python Code 	ASISA A, ANCE DEFRIN P, SHAJAL S, SHIJONIDA J V, ANCE C	In-Progress

4.	Deployment of App in IBM Cloud	<ul style="list-style-type: none"> ● Containerize The App ● Upload Image to IBM Container Registry ● Deploy in Kubernetes 	ASISA A, ANCE DEFRIN P, SHAJAL S, SHIJONIDA J V, ANCE C	In-Progress
----	--------------------------------	--	---	-------------

5.	Ideation Phase	<ul style="list-style-type: none"> • Literature Survey On The Selected Project & Information Gathering • Prepare Empathy Map • Ideation 	ASISA A, ANCE DEFRIN P, SHAJAL S, SHIJONIDA J V, ANCE C	Review
6.	Project Design Phase – I	<ul style="list-style-type: none"> • Proposed Solution • Problem Solution Fit • Solution Architecture 	ASISA A, ANCE DEFRIN P, SHAJAL S, SHIJONIDA J V, ANCE C	Review
7.	Project Planning Phase	<ul style="list-style-type: none"> • Prepare Milestone & Activity List • Sprint Delivery Plan 	ASISA A, ANCE DEFRIN P, SHAJAL S, SHIJONIDA J V, ANCE C	In-Progress
8.	Project Development Phase	<ul style="list-style-type: none"> • Delivery Of Sprint-1 • Delivery Of Sprint-2 • Delivery Of Sprint-3 • Delivery Of Sprint-4 	ASISA A, ANCE DEFRIN P, SHAJAL S, SHIJONIDA J V, ANCE C	In-Progress

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	2	High	5

			email, password, and confirming my password.			
Sprint-1		USN-2	As a user, I can register for the application through E-mail	1	Medium	5
Sprint-1	Confirmation	USN-3	As a user, I will receive confirmation email once I have registered for the application	2	Medium	5

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	2	High	5
Sprint-2	Dashboard	USN-5	As a user, I can view the products which are available	4	High	5
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	5
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	5

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	5
Sprint-4	Contact Administrator	USN-9	I can be able to report any difficulties I experience as a report	5	Medium	5

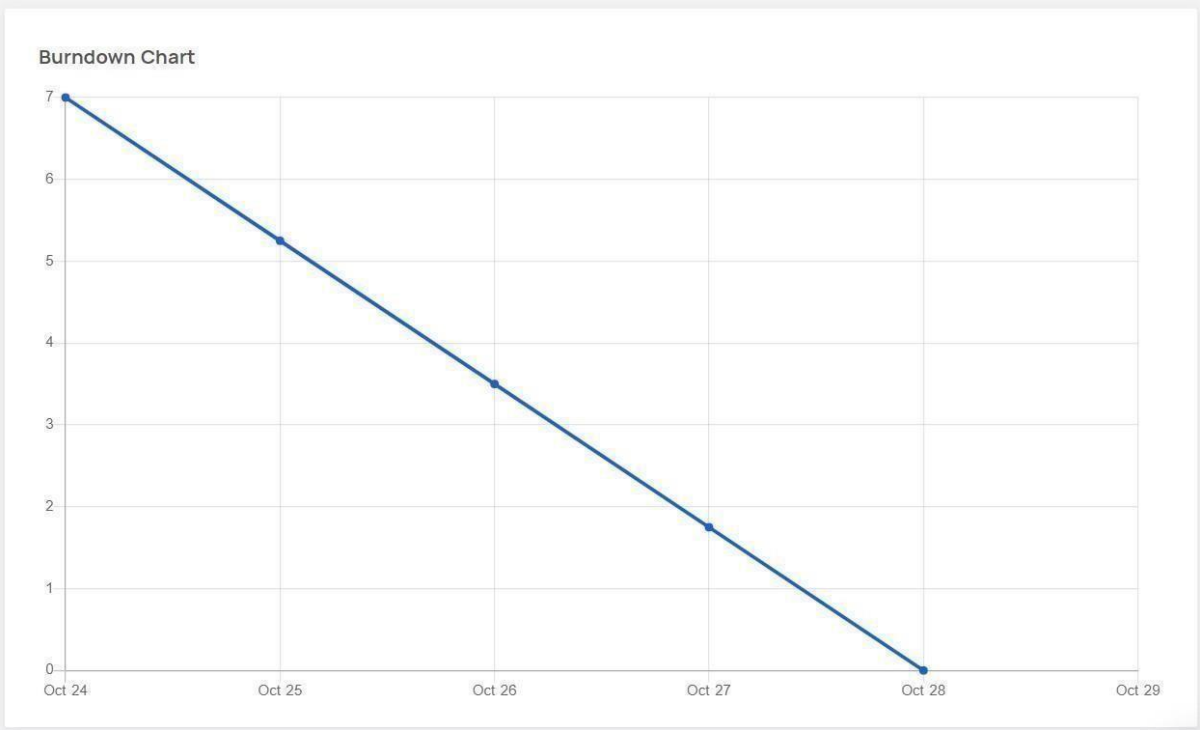
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	7	6 Days	24 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-2	9	6 Days	31 Oct 2022	08 Nov 2022	9	08 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

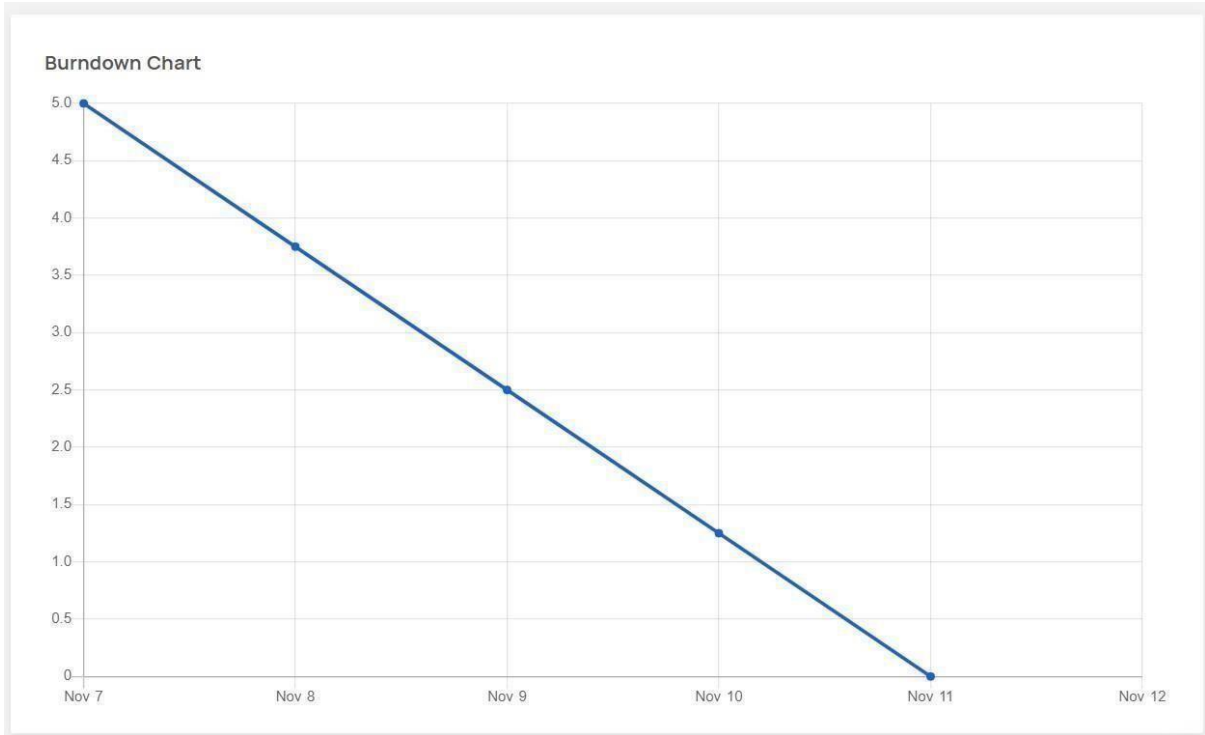
Velocity:

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

Burndown Chart:



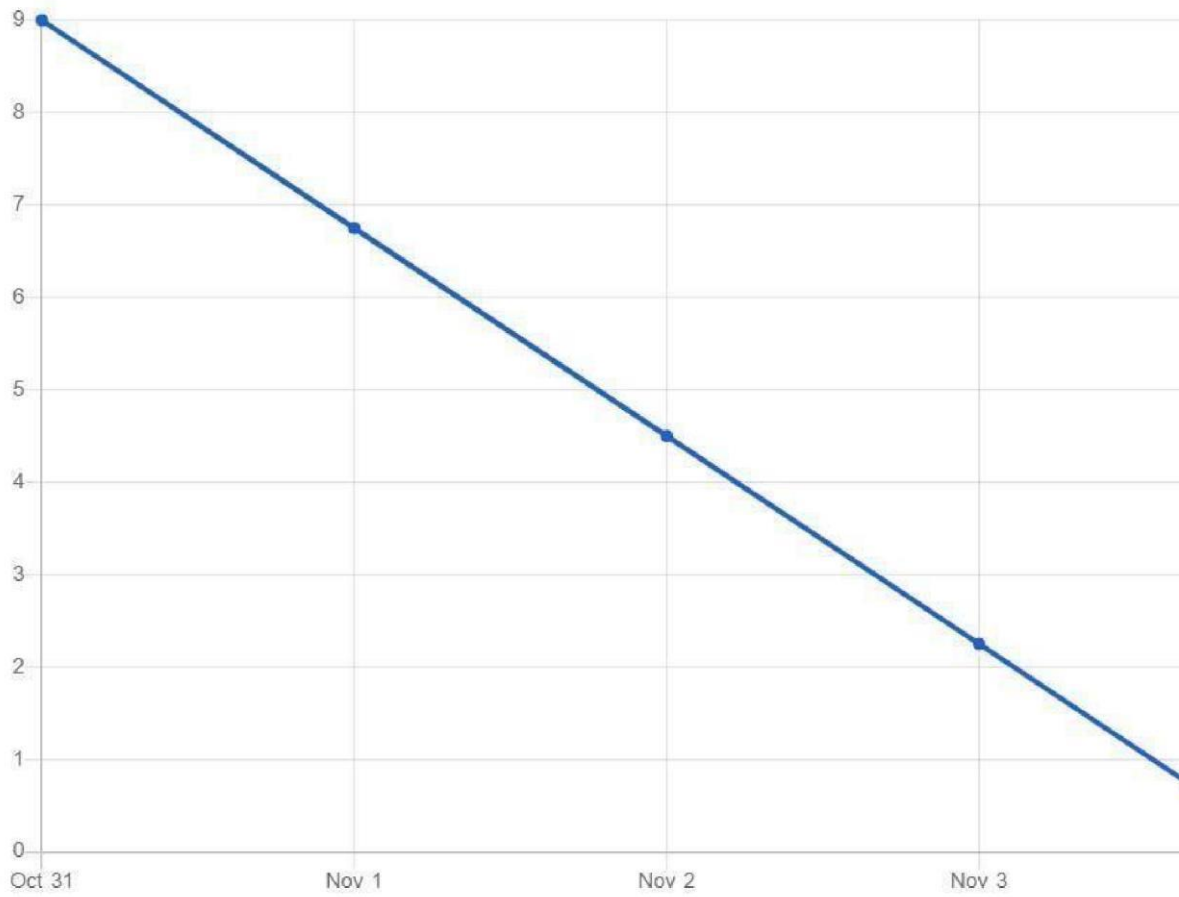
SPRINT 1



SPRINT 2

SPRINT 3

Burndown Chart



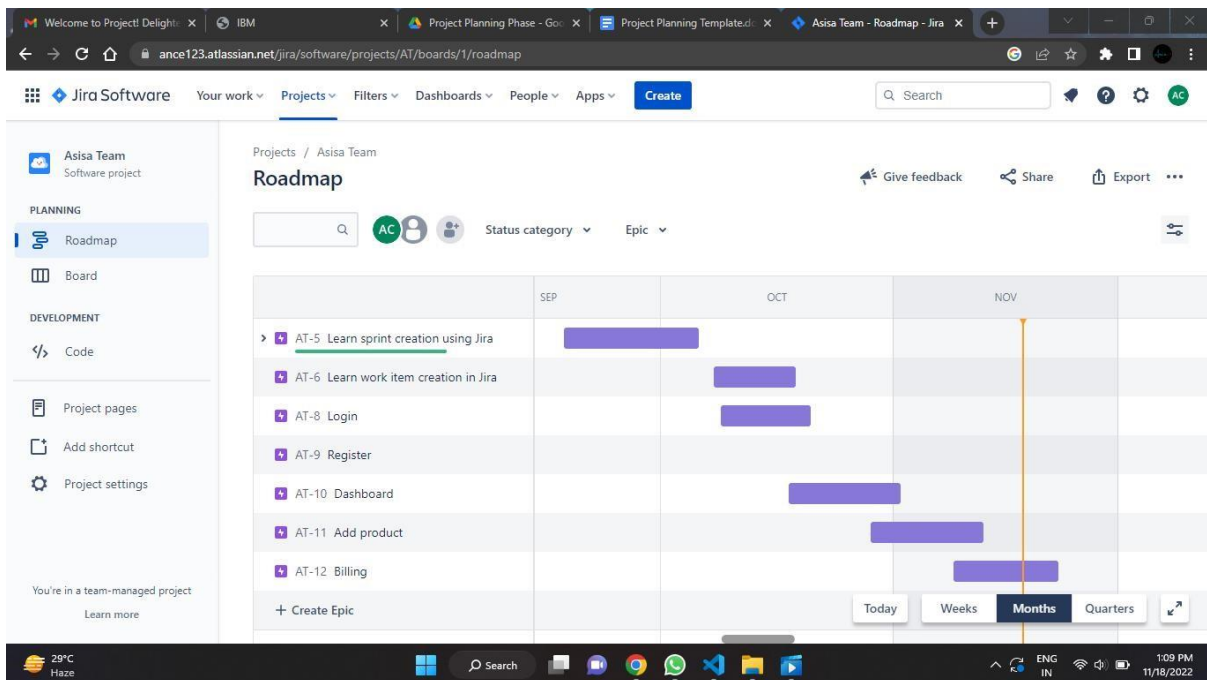
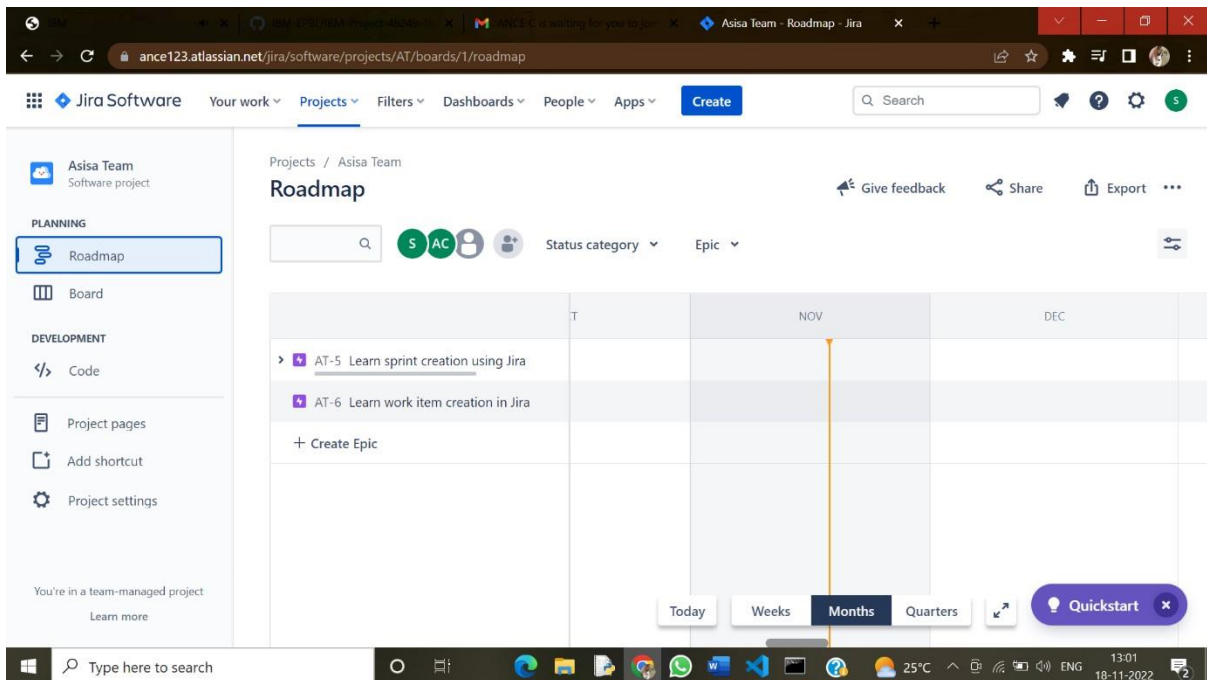
Burndown Chart



Sprint - 4

6.3 Reports from JIRA

The screenshot displays the Jira Software 'AT board' (Agile board) for the 'Asisa Team' project. The interface is viewed through a web browser with the URL `ance123.atlassian.net/jira/software/projects/AT/boards/1`. The top navigation bar includes 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'People', 'Apps', and a 'Create' button. A search bar is also present. The left sidebar shows the project structure with 'Asisa Team' as the 'Software project'. Under 'PLANNING', 'Roadmap' and 'Board' are listed, with 'Board' selected. Under 'DEVELOPMENT', 'Code', 'Project pages', 'Add shortcut', and 'Project settings' are visible. The main content area is titled 'AT board' and shows a Kanban board with three columns: 'TO DO 1 ISSUE', 'IN PROGRESS', and 'DONE'. The 'TO DO' column contains a card for 'Time frame of sprint' with a link to 'LEARN SPRINT CREATION USING J...' and a sub-task 'AT-7'. A 'Create issue' button is at the bottom of this column. The 'IN PROGRESS' and 'DONE' columns are currently empty. A 'GROUP BY' dropdown is set to 'None'. At the bottom right, there is a 'Quickstart' button. The Windows taskbar at the very bottom shows the search bar, task view, and various application icons, along with system information like '25°C' and '13:01 18-11-2022'.



7. CODING AND SOLUTIONING

7.1 Feature 1

App.py

```
from flask import Flask, render_template, request, redirect, url_for, flash
app = Flask(__name__)
import ibm_db
from flask_login import login_user, current_user, logout_user,
```

```
login_required,LoginManager,UserMixin import
datetime
from sendgrid import SendGridAPIClient
```

```
from sendgrid.helpers.mail import Mail
```

```
dsn_hostname = "9938aec0-8105-433e-8bf9-
0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud" # e.g.:
"54a2f15b-5c0f-46df-8954-
7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud" dsn_uid
= "krh91100"      # e.g. "abc12345"
dsn_pwd = "gSPyVtDpdim5wKGL"      # e.g. "7dBZ3wWt9XN6$o0J"
dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "bludb"      # e.g. "BLUDB" dsn_port
= "32459"      # e.g. "32733" dsn_protocol =
"TCPIP"      # i.e. "TCPIP" dsn_security = "SSL"
#i.e. "SSL"
```

```
dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname,
dsn_port, dsn_protocol, dsn_uid, dsn_pwd,dsn_security)
```

```
print(dsn)
```

```
try:
```

```
    conn = ibm_db.connect(dsn, "", "")    print ("Connected to database: ",
dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)
```

```
except:
```

```
    print ("Unable to connect: ", ibm_db.conn_errormsg() )
```

```
SECRET_KEY = 'Vibinprakash'
```

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

```
def home():    if request.method ==  
'POST':
```

```
    name=request.form.get('user')  
    password=request.form.get('password')  
    print(name,password)  
    sql = "SELECT * FROM users WHERE user_name =? AND password=?"  
    stmt = ibm_db.prepare(conn, sql)    ibm_db.bind_param(stmt,1,name)  
    ibm_db.bind_param(stmt,2,password)  
    ibm_db.execute(stmt)    account  
= ibm_db.fetch_assoc(stmt)  
    print (account)  
if account:
```

```
    return redirect(url_for('dashboard'))
```

```
else:
```

```
    msg='invalid user name and password'  
    return render_template('loginpage.html',msg=msg)
```

```
else:
```

```
    return render_template('loginpage.html')
```

```
@app.route("/register",methods=['GET', 'POST'])
```

```
def register():    if request.method == 'POST':
```

```
    name=request.form.get('full')  
    user_name=request.form.get('user')  
    email=request.form.get('email')  
    phone=request.form.get('phone')  
    password=request.form.get('password')  
    confirm=request.form.get('confirm')  
    print(confirm)    if  
password==confirm:
```

```

        sql ="SELECT id FROM users ORDER BY ID DESC limit 1"
        stm=ibm_db.exec_immediate(conn,sql)
while ibm_db.fetch_row(stm) != False:
        count=ibm_db.result(stm,0)
        print(count)

```

```

        insert=f"insert into users values ({int(count)+1}, '{name}',
' {user_name}', '{email}', '{password}', '{phone}')"
table=ibm_db.exec_immediate(conn,insert)
        return redirect(url_for('home'))

```

```

else:
        msg='invalid user name and password'
return render_template('register.html',msg=msg)
else:
        return render_template('register.html')

```

```

@app.route("/searchproduct",methods=['GET', 'POST'])
def searchproduct():    if request.method == 'POST':
Product=request.form.get('search')
        sql = "SELECT * FROM products WHERE product =?"
        stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,Product)
ibm_db.execute(stmt)    account =
ibm_db.fetch_assoc(stmt)
        print    (account)    return
render_template('searchproduct.html',product=account)    else:
        return render_template('searchproduct.html')

```

```

@app.route("/viewbill",methods=['GET', 'POST']) def
viewbill():
        sql ="SELECT * FROM billing"

        stmt = ibm_db.exec_immediate(conn, sql)

        bill=[]
        amount=0

```

```

        while ibm_db.fetch_row(stmt) != False:
            dic=dict()
            dic['invoice']=ibm_db.result(stmt, 1)
            dic['product']=ibm_db.result(stmt, 3)
            dic['price']=ibm_db.result(stmt, 4)
            price=ibm_db.result(stmt, 4)
            dic['quantity']=ibm_db.result(stmt,5)
            quantity=ibm_db.result(stmt,5)
            dic['total']=int(price)*int(quantity)
            total=int(price)*int(quantity)      amount
            +=total
            bill.append(dic)

    print(bill)
    print(amount)
    return render_template('viewbill.html',datas=bill)

```

```

@app.route("/minimum",methods=['GET', 'POST'])
def minimum():    sql ="SELECT * FROM
products"    # stmt = ibm_db.prepare(conn, sql)
                # ibm_db.bind_param(stmt,1,name)    #
ibm_db.bind_param(stmt,2,password)
stmt = ibm_db.exec_immediate(conn, sql)

```

```

    datas=[]    while
ibm_db.fetch_row(stmt) != False:
    dic=dict()
    dic['product']=ibm_db.result(stmt, 0)
    dic['stock']=ibm_db.result(stmt, 1)
    dic['price']=ibm_db.result(stmt, 2)
    dic['alert']=ibm_db.result(stmt, 3)
    datas.append(dic)    print(datas)
        # ibm_db.execute(stmt)
        # account = ibm_db.fetchall(stmt)
        # print (account)
    return render_template('minimum.html',datas=datas)

```

```

@app.route("/dashboard",methods=['GET', 'POST']) def
dashboard():
    sql ="SELECT * FROM products"
    stmt = ibm_db.exec_immediate(conn, sql)

```

```

    datas=[]    low=0    count=0    while
ibm_db.fetch_row(stmt) != False:
dic=dict()
dic['product']=ibm_db.result(stmt, 0)
dic['stock']=ibm_db.result(stmt, 1)
stock=ibm_db.result(stmt, 1)
dic['price']=ibm_db.result(stmt, 2)
dic['alert']=ibm_db.result(stmt, 3)
alert=ibm_db.result(stmt, 3)    if
int(stock)< int(alert):        low += 1
    datas.append(dic)
    count+=1
print(datas)
sql ="SELECT  * FROM billing"

```

```

stmt = ibm_db.exec_immediate(conn, sql)

```

```

bill=[]
amount=0    bill_count=0    while
ibm_db.fetch_row(stmt) != False:
dic=dict()
dic['invoice']=ibm_db.result(stmt, 1)
dic['product']=ibm_db.result(stmt, 3)
dic['price']=ibm_db.result(stmt, 4)
price=ibm_db.result(stmt, 4)
dic['quantity']=ibm_db.result(stmt,5)
quantity=ibm_db.result(stmt,5)
dic['total']=int(price)*int(quantity)
total=int(price)*int(quantity)    amount
+=total    bill_count+=1
    bill.append(dic)

```

```

    print(bill)
print(amount)    return
render_template('dashboard.html',datas=datas,low=low,amount=amount,count=
count,bill_count=bill_count)

```

```

@app.route("/billing",methods=['GET', 'POST'])
def billing():
date=datetime.datetime.today().date()    if

```



```

request.method == 'POST':
invoice=request.form.get('invoice')
date=request.form.get('date')
product=request.form.get('product')
quantity=request.form.get('quantity')
price=request.form.get('price')
    insert=f"insert into billing values ( {invoice[-1]}, '{invoice}', '{date}',
'{'product}',          {int(quantity)},          {int(price)})"
table=ibm_db.exec_immediate(conn,insert)          sql =
"SELECT * FROM products WHERE product =?"          stmt
=          ibm_db.prepare(conn,          sql)
ibm_db.bind_param(stmt,1,product)
ibm_db.execute(stmt)          account =
ibm_db.fetch_assoc(stmt)          try:
    count=account['STOCK']
alert=account['Alert']
update_count=int(count)-int(quantity)          if
int(update_count) < int(alert):
print('email code')

    #write sendgrid email code here

message = Mail(

    from_email='shajalraj333@gmail.com',
to_emails='shijonida2@gmail.com',          subject='Sending
with Twilio SendGrid is Fun',

    html_content='<strong>and easy to do anywhere, even with
Python</strong>')          try:          sg
=SendGridAPIClient('SG.ugbhjeYMQkKqLEAXNHd3ig.ovmpPK1LNdf_oXy
bocXoEx qsjavVbSEefk0NvjqCEJs')
    response = sg.send(message)
print(response.status_code)
print(response.body)
print(response.headers)          except
Exception as e:
    print(e)

```

```

        update=f"UPDATE products SET  stock = {update_count} WHERE
product = '{product}'"
        table=ibm_db.exec_immediate(conn,update)
        return redirect(url_for('dashboard'))
except:

```

```

        return redirect(url_for('dashboard'))

```

```

else:
return
render
_template
ate('bil
ling.ht
ml',dat
e=date
,invoic
e=invo
ice_no
())

```

```

@app.route("/addproduct",methods=['GET', 'POST'])
def addproduct():    if request.method == 'POST':
Product=request.form.get('Product')
        Stock=request.form.get('Stock')
        Price=request.form.get('Price')
Alert=request.form.get('Alert')
print(Product,Stock,Price,Alert)
        insert=f"insert into products values ( '{Product}', {int(Stock)},
{int(Price)}, {int(Alert)})"
        table=ibm_db.exec_immediate(conn,insert)

        return redirect(url_for('dashboard'))
else:
        return render_template('addproduct.html')

```

```

def invoice_no():
    sql ="SELECT  id FROM billing ORDER BY ID DESC limit 1"
    stm=ibm_db.exec_immediate(conn,sql)    while

```

```

ibm_db.fetch_row(stm) != False:      count=ibm_db.result(stm,0)
if count:
    return f'bill00{int(count)+1}'
else:    return 'bill001'

```

```
app.run(debug=True)
```

TEMPLATES:

Reg.html

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <title> Inventory Managment System - Register </title>
    <link rel="stylesheet" href="{ { url_for('static',filename='reg.css') } }">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div class="container">
      <div class="mx-4 mt-2 text-danger" style="color: red;">{{ msg }}</div>
      <div class="title">Registration</div>
      <div class="content">
        <form method="POST" action="">
          <div class="user-details">
            <div class="input-box">
              <span class="details">Full Name</span>
              <input type="text" placeholder="Enter your name" name="full"
required>
            </div>
            <div class="input-box">
              <span class="details">Username</span>
              <input type="text" placeholder="Enter your username" name="user"
required>
            </div>
            <div class="input-box">
              <span class="details">Email</span>
              <input type="text" placeholder="Enter your email" name="email"
required>
            </div>

```

```

        <div class="input-box">
            <span class="details">Phone Number</span>
            <input type="text" placeholder="Enter your number" name="phone"
required >
        </div>
        <div class="input-box">
            <span class="details">Password</span>
            <input type="text" placeholder="Enter your password"
name="password" required>
        </div>
        <div class="input-box">
            <span class="details">Confirm Password</span>
            <input type="text" placeholder="Confirm your password"
name="confirm" required>
        </div>
        </div>

        <div class="button">
            <input type="submit" value="Register">
        </div>
        <p>have an account? <a href="{{ url_for('home') }}">login</a></p>
</form>
</div>
</div>

</body>
</html>

```

Login.html

```

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

<head>
    <meta charset="UTF-8">
    <title> Inventory Managment System - Login</title>
    <link rel="stylesheet" href="{{ url_for('static',filename='reg.css') }}">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

```

```
<body>
  <div class="container">
    <div class="title">Login</div>
    <div class="content">
      <div class="mx-4 mt-2 text-danger" style="color: red;">{{ msg
}}</div>
      <form method="POST" action="">

        <div class="login">

          <div class="input-box">
            <span class="detailslog">Username</span>
            <input type="text" placeholder="Enter your username"
name="user" required>
          </div>

          <br>

          <div class="input-box">
            <span class="detailslog">Password</span>
            <input type="text" placeholder="Enter your password"
name="password" required>
          </div>

          <br>

          <div class="button1">
            <input type="submit" value="Login">
          </div>
        </div>

        <p>Don't have an account? <a href="{{ url_for('register')
}}">Register here</a></p>

      </form>
    </div>
  </div>
</body>
```

</html>

Addprod.html

<!DOCTYPE html>

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

<head>

<meta charset="UTF-8">

<title> Inventory Managment System - Add Product </title>

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

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

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/css/bootstrap.min.css" rel="stylesheet" />

<link rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.3/css/all.min.css" />

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

</head>

<body>

<div class="d-flex" id="wrapper">

<!-- Sidebar -->

<div class="bg-white" id="sidebar-wrapper">

<div class="sidebar-heading text-center py-4 primary-text fs-4 fw-bold text-uppercase border-bottom"><i

class="fas fa-shopping-bag"></i> Fruit Shop</div>

<div class="list-group list-group-flush my-3">

<div>

<i

class="fas fa-tachometer-alt me-2"></i> Dashboard

<a href="{{ url_for('addproduct') }}"

class="list-group-item list-group-item-action bg-transparent secondtext fw-bold active"><i class="fas fa-project-diagram me-2"></i> Add Product

<i

```

        class="fas fa-chart-line me-2"></i>Search Product</a>
<a href="{ { url_for('minimum') } }" class="list-group-item list-group-item-
action bg-transparent second-text fw-bold"><i
        class="fas fa-paperclip me-2"></i>Minimun Quantity</a>
<a href="{ { url_for('billing') } }"
        class="list-group-item list-group-item-action bg-transparent
secondtext fw-bold"><i
        class="fas fa-gift me-2"></i>Generate Bill</a>
<a href="{ { url_for('viewbill') } }"
        class="list-group-item list-group-item-action bg-transparent
secondtext fw-bold"><i
        class="fas fa-gift me-2"></i>View Bill</a>
</div>
<div>
        <a href="{ { url_for('home') } }"
        class="list-group-item list-group-item-action bg-transparent
textdanger fw-bold "><i
        class="fas fa-power-off me-2"></i>Logout</a>
</div>
</div>
</div>
<!-- /#sidebar-wrapper -->

<!-- Page Content -->
<div id="page-content-wrapper">
        <nav class="navbar navbar-expand-lg navbar-light bg-transparent py-4
px4">
                <div class="d-flex align-items-center">
                        <i class="fas fa-align-left primary-text fs-4 me-3"
id="menutoggle"></i>
                        <h2 class="fs-2 m-0"></h2>
                </div>

                <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent"
ariacontrols="navbarSupportedContent"
                        aria-expanded="false" aria-label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                </button>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">

```

```

    </div>
</nav>

<div class="container ">

    <div class="content">

        <form method="POST" action="">
            <div class="title my-2">Add Product</div>
            <div class="user-details text-left">
                <div class="input-box">
                    <span class="details">Product Name</span>
                    <input type="text" placeholder="Product Name" name="Product"
required>
                </div>
                <div class="input-box">
                    <span class="details">Stock Quantity</span>
                    <input type="number" placeholder="Stock Quantity" name="Stock"
required>
                </div>
                <div class="input-box">
                    <span class="details">Product
Price<small>&nbsp;/<small></span>
                    <input type="number" placeholder="Product Price" name="Price"
required>
                </div>
                <div class="input-box">
                    <span class="details">Stock Alert<small>&nbsp;min.
Qty</small></span>
                    <input type="number" placeholder="Stock Alert" name="Alert"
required>
                </div>
            </div>
            <div class="button">
                <input type="submit" value="Add Product">
            </div>
        </form>

```



```

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

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/js/bootstrap.bundle.
min.js"></script>
<script>
    var el = document.getElementById("wrapper");
    var toggleButton = document.getElementById("menu-toggle");

    toggleButton.onclick = function () {
        el.classList.toggle("toggled");
    };
</script>
</body>

</html>

```

Billing.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title> Inventory Managment System - Billing </title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- custom css file link -->
    <link rel="stylesheet" href="{ { url_for('static',filename='bill.css') } }">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/css/bootstrap.min.
css" rel="stylesheet" />
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.3/css/all.min.css
" />

```

```

<link rel="stylesheet" href="{{ url_for('static',filename='dash.css') }}" />
</head>
<body>

<div class="d-flex" id="wrapper">
  <!-- Sidebar -->
  <div class="bg-white" id="sidebar-wrapper">
    <div class="sidebar-heading text-center py-4 primary-text fs-4 fw-bold
text-uppercase border-bottom"><i
      class="fas fa-shopping-bag"></i> &nbsp; Fruit Shop</div>
    <div class="list-group list-group-flush my-3">
      <div>
        <a href="{{ url_for('dashboard') }}" class="list-group-item
listgroup-item-action bg-transparent second-text "><i
          class="fas fa-tachometer-alt me-2"></i>Dashboard</a>
        <a href="{{ url_for('addproduct') }}"
          class="list-group-item list-group-item-action bg-transparent
second-text fw-bold"><i
          class="fas fa-project-diagram me-2"></i>Add Product</a>
        <a href="{{ url_for('searchproduct') }}" class="list-group-item listgroup-item-
action bg-transparent second-text fw-bold"><i
          class="fas fa-chart-line me-2"></i>Search Product</a>
        <a href="{{ url_for('minimum') }}" class="list-group-item listgroup-item-
action bg-transparent second-text fw-bold"><i
          class="fas fa-paperclip me-2"></i>Minimun Quantity</a>
        <a href="{{ url_for('billing') }}"
          class="list-group-item list-group-item-action bg-transparent
second-text fw-bold active"><i
          class="fas fa-gift me-2"></i>Generate Bill</a>
        <a href="{{ url_for('viewbill') }}"
          class="list-group-item list-group-item-action bg-transparent
second-text fw-bold"><i
          class="fas fa-gift me-2"></i>View Bill</a>
      </div>
      <div>
        <a href="{{ url_for('register') }}"
          class="list-group-item list-group-item-action bg-transparent
text-danger fw-bold "><i

```

```

        class="fas fa-power-off me-2"></i>Logout</a>
</div>
    </div>
</div>
<!-- /#sidebar-wrapper -->

<!-- Page Content -->
<div id="page-content-wrapper">
    <nav class="navbar navbar-expand-lg navbar-light bg-transparent py-4
px-4">
        <div class="d-flex align-items-center">
            <i class="fas fa-align-left primary-text fs-4 me-3"
id="menutoggle"></i>

        </div>

        <button class="navbar-toggler" type="button" data-
bstoggle="collapse"
            data-bs-target="#navbarSupportedContent"
ariacontrols="navbarSupportedContent"
            aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse"
id="navbarSupportedContent">

            </div>
        </nav>

        <div class="container">
            <div class="mx-4 mt-2 text-danger" style="color: red;">{{ msg
}}</div>
            <form method="POST" action="">

                <div class="row">

                    <div class="col">

                        <h3 class="title">Billing</h3>

```

```

<div class="row gx-3">
  <div class="col inputBox">
    <span>Invoice no :</span>
    <input type="text" name="invoice" value="{{ invoice }}"
readonly>
  </div>
  <div class="col inputBox">
    <span>Bill Date:</span>
    <input type="text" name="date" value="{{ date }}"
readonly>
  </div>
</div>
<div class="inputBox">
  <span>Product Name :</span>
  <input type="text" name="product" placeholder="Enter
Product" required>
</div>
<div class="inputBox">
  <span>Product Quantity:</span>
  <input type="number" name="quantity" placeholder="Enter
Quantity" required>
</div>
<div class="inputBox">
  <span>Price/Quantity:</span>
  <input type="number" name="price" placeholder="Enter
Price" required>
</div>
</div>
</div>
<div class="buttonrow">
  <input type="submit" value="Save" class="submit-btn">
  <input type="reset" value="Cancel" class="submit-btn1 ">
</div>

```

```

        </form>

    </div>
</div>
</div>
<!-- /#page-content-wrapper -->

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/js/bootstrap.bundle.
min.js"></script>
<script>
    var el = document.getElementById("wrapper");
    var toggleButton = document.getElementById("menu-toggle");

    toggleButton.onclick = function () {
        el.classList.toggle("toggled");
    };
</script>
</body>
</html>

```

Dashboard.html

```

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

<head>
    <meta charset="UTF-8" />
    <title> Inventory Managment System - Dashboard </title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/css/bootstrap.min.css"
rel="stylesheet" />
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.3/css/all.min.css"
"/>
    <link rel="stylesheet" href="{ { url_for('static',filename='dash.css') } }" />

```

```

</head>

<body>
  <div class="d-flex" id="wrapper">
    <!-- Sidebar -->
    <div class="bg-white" id="sidebar-wrapper">
      <div class="sidebar-heading text-center py-4 primary-text fs-4 fw-bold text-uppercase border-bottom"><i
        class="fas fa-shopping-bag"></i> &nbsp; Fruit Shop</div>
      <div class="list-group list-group-flush my-3">
        <div>
          <a href="{ { url_for('dashboard') } }" class="list-group-item list-group-item-action bg-transparent second-text active"><i
            class="fas fa-tachometer-alt me-2"></i>Dashboard</a>
          <a href="{ { url_for('addproduct') } }"
            class="list-group-item list-group-item-action bg-transparent secondtext fw-bold"><i
              class="fas fa-project-diagram me-2"></i>Add Product</a>
          <a href="{ { url_for('searchproduct') } }" class="list-group-item listgroup-item-action bg-transparent second-text fw-bold"><i
            class="fas fa-chart-line me-2"></i>Search Product</a>
          <a href="{ { url_for('minimum') } }" class="list-group-item list-group-item-action bg-transparent second-text fw-bold"><i
            class="fas fa-paperclip me-2"></i>Minimun Quantity</a>
          <a href="{ { url_for('billing') } }"
            class="list-group-item list-group-item-action bg-transparent secondtext fw-bold"><i
              class="fas fa-gift me-2"></i>Generate Bill</a>
          <a href="{ { url_for('viewbill') } }"
            class="list-group-item list-group-item-action bg-transparent secondtext fw-bold"><i
              class="fas fa-gift me-2"></i>View Bill</a>
        </div>
        <div>
          <a href="{ { url_for('register') } }"
            class="list-group-item list-group-item-action bg-transparent textdanger fw-bold "><i
              class="fas fa-power-off me-2"></i>Logout</a>
        </div>
      </div>
    </div>
  </div>

```

```

<!-- /#sidebar-wrapper -->

<!-- Page Content -->
<div id="page-content-wrapper">
    <nav class="navbar navbar-expand-lg navbar-light bg-transparent py-4
px-4">
        <div class="d-flex align-items-center">
            <i class="fas fa-align-left primary-text fs-4 me-3"
id="menutoggle"></i>
            <h2 class="fs-2 m-0">Dashboard</h2>
        </div>

        <button class="navbar-toggler" type="button" data-
bstoggle="collapse"
            data-bs-target="#navbarSupportedContent"
ariacontrols="navbarSupportedContent"
            aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle second-text fw-bold"
href="#" id="navbarDropdown"
                    role="button" data-bs-toggle="dropdown"
ariaexpanded="false">
                        <i class="fas fa-user me-2"></i>Nicky Roy
                    </a>
                    <ul class="dropdown-menu"
arialabelledby="navbarDropdown">
                        <li><a class="dropdown-item" href="{ { url_for('home')
}}">Logout</a></li>
                    </ul>
                </li>
            </ul>
        </div>
    </nav>

    <div class="container-fluid px-4">

```

```

<div class="mx-4 mt-2 text-danger">{{ msg }}</div>
<div class="row g-3 my-2">
  <div class="col-md-3">
    <div class="p-3 bg-white shadow-sm d-flex justify-
contentaround align-items-center rounded">
      <div>
        <h3 class="fs-2">{{ count }}</h3>
        <p class="fs-5">Products</p>
      </div>

      <i class="fas fa-apple-alt fs-1 primary-text border rounded-full
secondary-bg p-3"></i>
    </div>
  </div>

  <div class="col-md-3">
    <div class="p-3 bg-white shadow-sm d-flex justify-
contentaround align-items-center rounded">
      <div>
        <h3 class="fs-2">{{ amount }}</h3>
        <p class="fs-5">Sales Amount</p>
      </div>

      <i class="fas fa-hand-holding-usd fs-1 primary-text border
rounded-full secondary-bg p-3"></i>
    </div>
  </div>

  <div class="col-md-3">
    <div class="p-3 bg-white shadow-sm d-flex justify-
contentaround align-items-center rounded">
      <div>
        <h3 class="fs-2">{{ low }}</h3>
        <p class="fs-5">Minimum Quantity </p>
      </div>
      <i class="fab fa-microblog fs-1 primary-text border
roundedfull secondary-bg p-3"></i>
    </div>
  </div>

  <div class="col-md-3">

```



```
<div class="p-3 bg-white shadow-sm d-flex justify-content-around align-items-center rounded">
```

```
<div>
```

```
<h3 class="fs-2">{{ bill_count }}</h3>
```

```
<p class="fs-5">Bills</p>
```

```
</div>
```

```
<i class="fas fa-file-invoice fs-1 primary-text border roundedfull secondary-bg p-3"></i>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="row my-5 card">
```

```
<h3 class="fs-4 m-3 text-center">Products</h3>
```

```
<div class="col">
```

```
<table class="table bg-white rounded shadow-sm table-hover text-center ">
```

```
<thead>
```

```
<tr>
```

```
<th scope="col">Product</th>
```

```
<th scope="col">Quantity<small>(kg)</small></th>
```

```
<th scope="col">Price<small>/(kg)</small></th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
{% for i in datas %}
```

```
<tr>
```

```
<td>{{ i['product'] }}</td>
```

```
<td>{{ i['stock'] }}</td>
```

```
<td>{{ i['price'] }}</td>
```

```
</tr>
```

```
{% endfor %}
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```

```

        </div>
    </div>
</div>
<!-- /#page-content-wrapper -->

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/js/bootstrap.bundle.
min.js"></script>
<script>
    var el = document.getElementById("wrapper");
    var toggleButton = document.getElementById("menu-toggle");

    toggleButton.onclick = function () {
el.classList.toggle("toggled");
    };
</script>
</body>

</html>

```

Minimum.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> Inventory Managment System - Minimum Quantity </title>

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/css/bootstrap.min.
css" rel="stylesheet" />
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.3/css/all.min.css
" />
    <link rel="stylesheet" href="{ { url_for('static',filename='dash.css') } }" />
</head>
<body>

```

```

<div class="d-flex" id="wrapper">
  <!-- Sidebar -->
  <div class="bg-white" id="sidebar-wrapper">
    <div class="sidebar-heading text-center py-4 primary-text fs-4 fw-bold
text-uppercase border-bottom"><i
      class="fas fa-shopping-bag"></i> &nbsp; Fruit Shop</div>
    <div class="list-group list-group-flush my-3">
      <div>
        <a href="{{ url_for('dashboard') }}" class="list-group-item
listgroup-item-action bg-transparent second-text "><i
          class="fas fa-tachometer-alt me-2"></i>Dashboard</a>
        <a href="{{ url_for('addproduct') }}"
          class="list-group-item list-group-item-action bg-transparent
second-text fw-bold"><i
          class="fas fa-project-diagram me-2"></i>Add Product</a>
        <a href="{{ url_for('searchproduct') }}" class="list-group-item listgroup-item-
action bg-transparent second-text fw-bold"><i
          class="fas fa-chart-line me-2"></i>Search Product</a>
        <a href="{{ url_for('minimum') }}" class="list-group-item listgroup-item-
action bg-transparent second-text fw-bold active"><i
          class="fas fa-paperclip me-2"></i>Minimun Quantity</a>
        <a href="{{ url_for('billing') }}"
          class="list-group-item list-group-item-action bg-transparent
second-text fw-bold"><i
          class="fas fa-gift me-2"></i>Generate Bill</a>
        <a href="{{ url_for('viewbill') }}"
          class="list-group-item list-group-item-action bg-transparent
second-text fw-bold"><i
          class="fas fa-gift me-2"></i>View Bill</a>
      </div>
      <div>
        <a href="{{ url_for('home') }}"
          class="list-group-item list-group-item-action bg-transparent
text-danger fw-bold "><i
          class="fas fa-power-off me-2"></i>Logout</a>
      </div>
    </div>
  </div>
  <!-- /#sidebar-wrapper -->

  <!-- Page Content -->

```

```

<div id="page-content-wrapper">
  <nav class="navbar navbar-expand-lg navbar-light bg-transparent py-4
px-4">
    <div class="d-flex align-items-center">
      <i class="fas fa-align-left primary-text fs-4 me-3"
id="menutoggle"></i>

    </div>

    <button class="navbar-toggler" type="button" data-
bstoggle="collapse"
      data-bs-target="#navbarSupportedContent"
      ariacontrols="navbarSupportedContent"
      aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse"
id="navbarSupportedContent">

      </div>
    </nav>

    <div class="container">
      <div class="card p-3 row my-5">
        <h3 class="fs-4 mb-3 text-center">Minimum Quantity List</h3>
        <div class="col">
          <table class="table bg-white rounded shadow-sm table-hover
table-bordered text-center">
            <thead>
              <tr>

                <th scope="col">Product</th>
                <th scope="col">Min. Quantity</th>
                <th scope="col">Present Quantity</th>

              </tr>
            </thead>
            <tbody>
              { % for i in datas % }
              <tr>

```

```

        <td>{{ i['product'] }}</td>
        <td>{{ i['alert'] }}</td>
        <td>{{ i['stock'] }}</td>

    </tr>

    {% endfor %}

</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<!-- /#page-content-wrapper -->

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/js/bootstrap.bundle.
min.js"></script>
<script>
    var el = document.getElementById("wrapper");
    var toggleButton = document.getElementById("menu-toggle");

    toggleButton.onclick = function () {
el.classList.toggle("toggled");
    };
</script>

</body>
</html>

```

Searchprod.html

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title> Inventory Managment System - Search Product </title>
```

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

```
  <script src="https://kit.fontawesome.com/b99e675b6e.js"></script>
```

```
  <link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/css/bootstrap.min.
css" rel="stylesheet" />
```

```
  <link rel="stylesheet"
```

```
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.3/css/all.min.css
" />
```

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

```
</head>
```

```
<body>
```

```
  <div class="d-flex" id="wrapper">
```

```
    <!-- Sidebar -->
```

```
    <div class="bg-white" id="sidebar-wrapper">
```

```
      <div class="sidebar-heading text-center py-4 primary-text fs-4 fw-bold
text-uppercase border-bottom"><i
```

```
        class="fas fa-shopping-bag"></i> &nbsp; Fruit Shop</div>
```

```
      <div class="list-group list-group-flush my-3">
```

```
        <div>
```

```
          <a href="{ { url_for('dashboard') } }" class="list-group-item
listgroup-item-action bg-transparent second-text "><i
```

```
            class="fas fa-tachometer-alt me-2"></i>Dashboard</a>
```

```
          <a href="{ { url_for('addproduct') } }"
```

```
            class="list-group-item list-group-item-action bg-transparent
second-text fw-bold"><i
```

```
              class="fas fa-project-diagram me-2"></i>Add Product</a>
```

```
<a href="{ { url_for('searchproduct') } }" class="list-group-item listgroup-item-
action bg-transparent second-text fw-bold active"><i
```

```
  class="fas fa-chart-line me-2"></i>Search Product</a>
```

```
<a
```

```

href="{{ url_for('minimum') }}" class="list-group-item listgroup-item-action
bg-transparent second-text fw-bold"><i
    class="fas fa-paperclip me-2"></i>Minimun Quantity</a>
<a href="{{ url_for('billing') }}"
    class="list-group-item list-group-item-action bg-transparent
second-text fw-bold"><i
    class="fas fa-gift me-2"></i>Generate Bill</a>
<a href="{{ url_for('viewbill') }}"
    class="list-group-item list-group-item-action bg-transparent
second-text fw-bold"><i
    class="fas fa-gift me-2"></i>View Bill</a>
</div>
<div>
<a href="{{ url_for('home') }}"
    class="list-group-item list-group-item-action bg-transparent
text-danger fw-bold "><i
    class="fas fa-power-off me-2"></i>Logout</a>
</div>
</div>
<!-- /#sidebar-wrapper -->

<!-- Page Content -->
<div id="page-content-wrapper">
    <nav class="navbar navbar-expand-lg navbar-light bg-transparent py-4
px-4">
        <div class="d-flex align-items-center">
            <i class="fas fa-align-left primary-text fs-4 me-3"
id="menutoggle"></i>
            <h2 class="fs-2 m-0"></h2>
        </div>

        <button class="navbar-toggler" type="button" data-
bstoggle="collapse"
            data-bs-target="#navbarSupportedContent"
ariacontrols="navbarSupportedContent"
            aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse"

```

```
id="navbarSupportedContent">
```

```
</div>
```

```
</nav>
```

```
<div class="container mt-5">
```

```
<form method="POST" action="">
```

```
<div class="search_box w-50 m-auto">
```

```
<button class="search_btn" type="submit"><i class="fas  
fasearch"></i></button>
```

```
<input type="text" class="input_search" name="search"  
placeholder="search product?">
```

```
</div>
```

```
</form>
```

```
<div class="row my-5 card">
```

```
<h3 class="mt-3 mb-5 text-center">Product</h3>
```

```
<table class="table bg-white rounded shadow-sm table-hover  
textcenter" cellpadding="0" width="100%">
```

```
<thead>
```

```
<tr>
```

```
<th scope="col">Product</th>
```

```
<th scope="col">Quantity<small>  
&nbsp;(kg)</small></th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
{ % if product % }
```

```
<tr>
```

```
<td>{{ product['PRODUCT'] }}</td>
```

```
<td>{{ product['STOCK'] }}</td>
```

```
</tr>
```

```
{ % else % }
```

```
<tr>
```



```
        <td colspan='2'>no data</td>

    </tr>
    {% endif %}
</tbody>
</table>

</div>
</div>
</div>
</div>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/js/bootstrap.bundle.
min.js"></script>
```

```
<script>
    var el = document.getElementById("wrapper");
    var toggleButton = document.getElementById("menu-toggle");

    toggleButton.onclick = function () {
el.classList.toggle("toggled");
    };
</script>
```

```
</body>
```

```
</html>
```

Viewbill.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title> Inventory Managment System - View Bill </title>

  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/css/bootstrap.min.
css" rel="stylesheet" />
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.3/css/all.min.css
" />
  <link rel="stylesheet" href="{{ url_for('static',filename='dash.css') }}" />
</head>
<body >

```

```

<div class="d-flex" id="wrapper">
  <!-- Sidebar -->
  <div class="bg-white" id="sidebar-wrapper">
    <div class="sidebar-heading text-center py-4 primary-text fs-4 fw-bold
text-uppercase border-bottom"><i
      class="fas fa-shopping-bag"></i> &nbsp; Fruit Shop</div>
    <div class="list-group list-group-flush my-3">
      <div>
        <a href="{{ url_for('dashboard') }}" class="list-group-item list-
groupitem-action bg-transparent second-text "><i
          class="fas fa-tachometer-alt me-2"></i>Dashboard</a>
        <a href="{{ url_for('addproduct') }}"
          class="list-group-item list-group-item-action bg-transparent
secondtext fw-bold"><i
          class="fas fa-project-diagram me-
2"></i>Add Product</a>
        <a href="{{ url_for('searchproduct')
        }}"class="list-group-item listgroup-item-action bg-transparent second-text
fw-bold"><i
          class="fas fa-chart-line me-2"></i>Search Product</a>
        <a href="{{ url_for('minimum') }}" class="list-group-item list-groupitem-
action bg-transparent second-text fw-bold"><i

```

```

        class="fas fa-paperclip me-2"></i>Minimun Quantity</a>
    <a href="{{ url_for('billing') }}"
        class="list-group-item list-group-item-action bg-transparent
secondtext fw-bold"><i>
        class="fas fa-gift me-2"></i>Generate Bill</a>
    <a href="{{ url_for('viewbill') }}"
        class="list-group-item list-group-item-action bg-transparent
secondtext fw-bold active"><i>
        class="fas fa-gift me-2"></i>View Bill</a>
    </div>
    <div>
        <a href="{{ url_for('home') }}"
            class="list-group-item list-group-item-action bg-transparent
textdanger fw-bold "><i>
                class="fas fa-power-off me-2"></i>Logout</a>
    </div>
    </div>
</div>
<!-- /#sidebar-wrapper -->

<!-- Page Content -->
<div id="page-content-wrapper">
    <nav class="navbar navbar-expand-lg navbar-light bg-transparent py-4
px-4">
        <div class="d-flex align-items-center">
            <i class="fas fa-align-left primary-text fs-4 me-3"
id="menutoggle"></i>

            </div>

            <button class="navbar-toggler" type="button" data-
bstoggle="collapse"
                data-bs-target="#navbarSupportedContent"
ariacontrols="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse"
id="navbarSupportedContent">

```

```
</div>
</nav>
```

```
<div class="container">
  <div class="card p-3 row my-5">
    <h3 class="fs-4 mb-3 text-center"> Bills </h3>
    <div class="col">
      <table class="table bg-white rounded shadow-sm table-hover
tablebordered text-center">
        <thead>
          <tr>

            <th scope="col">invoice No</th>
            <th scope="col">Product</th>
            <th scope="col">Quantity<small> &nbsp; (kg)</small></th>
            <th scope="col">Price &nbsp;/ (kg)</small></th>
            <th scope="col">Total Price</th>

          </tr>
        </thead>
        <tbody>
          { % for i in datas % }
          <tr>

            <td>{ { i['invoice'] } }</td>
            <td>{ { i['product'] } }</td>
            <td>{ { i['price'] } }</td>
            <td>{ { i['quantity'] } }</td>
            <td>{ { i['total'] } }</td>

          </tr>
          { % endfor % }

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

```
    </div>
  </div>
<!-- /#page-content-wrapper -->
```

```
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/js/bootstrap.bundle.
min.js"></script>
```

```
  <script>
    var el = document.getElementById("wrapper");
    var toggleButton = document.getElementById("menu-toggle");

    toggleButton.onclick = function () {
      el.classList.toggle("toggled");
    };
  </script>
```

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

STATIC:
Add.css

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;
600;700&display=swap');
*{ margin: 0; padding: 0;
box-sizing: border-box; font-
family: 'Poppins',sans-serif; }

.container{ display: flex;
justify-content: center; align-
items: center; padding:25px;
min-height:calc( 100vh - 88px) ;
line-height: 2rem;
}
```

```
.container .title{ font-size: 25px; font-weight: 500; position: relative; text-align:center; }
```

```
.container form{ padding:20px; width:700px; background: #fff; box-shadow: 0 5px 10px rgba(0,0,0,.1); }
```

```
form .input-box span.details{ display: block; font-weight: 500; margin-bottom: 5px;
```

```
}
```

```
.user-details .input-box input{ height: 45px; width: 100%; outline: none; font-size: 16px; border-radius: 5px; padding-left: 15px; border: 1px solid #ccc; border-bottom-width: 2px; transition: all 0.3s ease;
```

```
}
```

```
.user-details .input-box input:focus, .user-details .input-box input:valid{ border-color: #f2f546; }
```

```
form .category{ display: flex; width: 80%; margin: 14px 0 ; justify-content: space-between; } form .category label{ display: flex; align-items: center; cursor: pointer; } form .category label .dot{ height: 18px; width: 18px; border-radius: 50%; margin-
```

```

right: 10px; background:
#d9d9d9; border: 5px solid
transparent; transition: all
0.3s ease;
}
#dot-1:checked ~ .category label .one, #dot-
2:checked ~ .category label .two, #dot-3:checked ~
.category label .three{ background: #d9e94c;
border-color: #d9d9d9;
} form
input[type="radio"]{
display: none;
} form .button{
height: 55px;
margin: 35px 0;
} form .button
input{ height:
100%; width:
50%; border-
radius: 5px;
border: none;
color: #fff; font-
size: 18px; font-
weight: 500; letter-
spacing: 1px;
cursor: pointer;
transition: all 0.3s
ease;
background: linear-gradient(135deg, #71b7e6, #9b59b6); margin-
left: 25%;
}

form .button input:hover{
/* transform: scale(0.99); */
background: linear-gradient(-135deg, #71b7e6, #fff23d);
}
@media(max-width: 584px){
.container{ max-
width: 100%;

```

```

} form .user-details .input-
box{  margin-bottom:
15px;  width: 100%;
}
form .category{
  width: 100%;
}
.content form .user-details{
max-height: 300px;  overflow-y:
scroll;
}
.user-details::-webkit-scrollbar{
width: 5px;
}
}
@media(max-width: 459px){
.container .content .category{
  flex-direction: column;
}
}

```

Bill.css

```

@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;
600;700&display=swap');

*{  font-family: 'Poppins', sans-
serif;  margin:0; padding:0;
box-sizing: border-box;  outline:
none; border:none;  text-
transform: capitalize;  transition:
all .2s linear;
}

.container{  display: flex;
justify-content: center;  align-
items: center;  padding:25px;
min-height:calc( 100vh - 88px) ;

}

```



```
.container form{  
padding:20px;  
width:700px; background:  
#fff;  
box-shadow: 0 5px 10px rgba(0,0,0,.1);  
}
```

```
.container form .row{  
display: flex; flex-wrap:  
wrap; gap:15px;  
}
```

```
.container form .row .col{  
flex:1 1 250px;  
}
```

```
.container form .row .col .title{  
font-size: 20px; color:#333;  
padding-bottom: 5px; text-  
transform: uppercase; text-  
align: center;  
}
```

```
.container form .row .col .inputBox{  
margin:15px 0;  
}
```

```
.container form .row .col .inputBox span{  
margin-bottom: 10px; display: block;  
}
```

```
.container form .row .col .inputBox input{  
width: 100%; border:1px solid #ccc;  
padding:10px 15px; font-size: 15px;  
text-transform: none;  
}
```

```
.container form .row .col .inputBox input:focus{  
border:1px solid #000;  
}
```

```
.container form .row .col .flex{  
  display: flex;  
  gap:15px;  
}
```

```
.container form .row .col .flex .inputBox{  margin-  
top: 5px;  
}
```

```
.container form .row .col .inputBox img{  
  height: 34px;  margin-  
top: 5px;  
  filter: drop-shadow(0 0 1px #000);  
}
```

```
.container form .submit-btn{  
width: 100%;  
padding:12px;  font-size:  
17px;  
  background: #27ae60;  
color:#fff;  
  margin-top: 5px;  
cursor: pointer;  border-  
radius: 10px;  
}
```

```
.container form .submit-btn1{  
width: 100%;  padding:12px;  
font-size: 17px;  
background: #ff3300;  
  color:#fff;  margin-  
top: 5px;  cursor:  
pointer;  border-radius:  
10px;  
}
```

```
.container form .submit-btn:hover{  
background: #2ecc71;  
}
```

```
.container form .submit-btn1:hover{  
background: #ff6741;  
}
```

```
.buttonrow { margin : 20px  
auto; display: flex; flex-  
direction: row; justify-content:  
space-between; gap:20px;  
}
```

```
input .danger{ background-  
color: red;  
color: red;  
}
```

Dashboard.css

```
@import  
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;  
600;700&display=swap');
```

```
:root {  
--main-bg-color: #0066ff;  
--main-text-color: #0066ff;  
--second-text-color: #bbbec5;  
--second-bg-color: #c1efde;  
}
```

```
.primary-text {  
color: var(--main-text-color);  
}
```

```
.second-text {  
color: var(--second-text-color);  
}
```

```
.primary-bg {  
background-color: var(--main-bg-color);  
}
```

```
.secondary-bg {
```

```
background-color: var(--second-bg-color);
}

.rounded-full {
border-radius: 100%;
}

#wrapper {
overflow-x: hidden;
background-image: linear-gradient(135deg, #71b7e6, #9b59b6);
}

#sidebar-wrapper { min-
height: 100vh; margin-left: -
15rem;
-webkit-transition: margin 0.25s ease-out;
-moz-transition: margin 0.25s ease-out; -o-
transition: margin 0.25s ease-out;
transition: margin 0.25s ease-out;
}

#sidebar-wrapper .sidebar-heading {
padding: 0.875rem 1.25rem; font-size:
1.2rem;
}

#sidebar-wrapper .list-group {
height: 89%; width: 15rem;
display: flex; justify-content:
space-between;
}

#page-content-wrapper {
min-width: 100vw;
}

#wrapper.toggled #sidebar-wrapper {
margin-left: 0;
}
```

```
#menu-toggle {  
  cursor: pointer;  
}
```

```
.list-group-item {  
  border: none;  
  padding: 20px 30px;  
}
```

```
.list-group-item.active {  
  background-color: transparent;  
  color: var(--main-text-color);  
  font-weight: bold;  border:  
  none;  
}
```

```
@media (min-width: 768px) {  
  #sidebar-wrapper {  margin-  
  left: 0;  
}
```

```
#page-content-wrapper {  
  min-width: 0;  width:  
  100%;  
}
```

```
#wrapper.toggled #sidebar-wrapper {  
  margin-left: -15rem;  
}  
}
```

Reg.css

```
@import  
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;  
600;700&display=swap');  
*{  margin: 0;  padding: 0;  
  box-sizing: border-box;  font-  
  family: 'Poppins',sans-serif;  
} body{  height:  
  100vh;  display: flex;
```

```
justify-content: center;
align-items: center;
padding: 10px;
  background: linear-gradient(135deg, #71b7e6, #9b59b6); }
.container{  max-width:
700px; width: 100%;
background-color: #fff;
padding: 25px 30px;
border-radius: 5px;
  box-shadow: 0 5px 10px rgba(0,0,0,0.15);
}
.container .title{  font-
size: 25px; font-weight:
500; position: relative;
text-align: center;
}
.container .title::before{
content: ""; position:
absolute; bottom: 0;
height: 3px; width:
30px; border-radius:
5px;
  background: linear-gradient(135deg, #71b7e6, #9b59b6); text-
align: center;
}
.content form .user-details{
  display: flex; flex-wrap:
wrap; justify-content: space-
between; margin: 20px 0 12px
0;
}

form .user-details .input-box{
margin-bottom: 15px; width:
calc(100% / 2 - 20px);
}

form .input-box span.details{
  display: block; font-
weight: 500;
}
```

```
form .input-box span.detailslog{
  display: block; font-
weight: 500; margin-
left:27%;
}
```

```
.login .input-box input{
height: 45px; width: 50%;
outline: none; font-size:
16px; border-radius: 5px;
padding-left: 15px;
border: 1px solid #ccc;
border-bottom-width: 2px;
transition: all 0.3s ease;
margin-left: 25%;
}
```

```
.user-details .input-box input{
height: 45px; width: 100%;
outline: none; font-size:
16px; border-radius: 5px;
padding-left: 15px; border:
1px solid #ccc; border-
bottom-width: 2px;
transition: all 0.3s ease;
}
```

```
.user-details .input-box input:focus, .user-details
.input-box input:valid{ border-color: #9b59b6;
}
```

```
form .gender-details .gender-title{
font-size: 20px; font-weight:
500;
} form .category{ display:
flex; width: 80%; margin:
14px 0 ; justify-content: space-
between;
} form .category label{
display: flex; align-items:
center; cursor: pointer; }
form .category label .dot{
height: 18px; width: 18px;
```

```
border-radius: 50%; margin-
right: 10px; background:
#d9d9d9; border: 5px solid
transparent; transition: all
0.3s ease;
}
#dot-1:checked ~ .category label .one, #dot-
2:checked ~ .category label .two, #dot-3:checked ~
.category label .three{ background: #9b59b6;
border-color: #d9d9d9;
}
form input[type="radio"]{
display: none;
} form .button{
height: 45px;
margin: 35px 0
} form .button input{
height: 100%; width:
100%; border-radius:
5px; border: none;
color: #fff; font-size:
18px; font-weight:
500; letter-spacing:
1px; cursor: pointer;
transition: all 0.3s ease;
background: linear-gradient(135deg, #71b7e6, #9b59b6);
}
```

```
form .button1{
height: 45px;
margin: 35px 0;

}
```

```
form .button1 input{
height: 100%; width:
50%; border-radius:
5px; border: none;
color: #fff; font-size:
18px; font-weight:
500; letter-spacing:
```



```
1px;  cursor: pointer;
transition: all 0.3s ease;
    background: linear-gradient(135deg, #71b7e6, #9b59b6);  margin-
left: 25%;
}
```

```
form .button1 input:hover{
/* transform: scale(0.99); */
    background: linear-gradient(-135deg, #71b7e6, #9b59b6);
}
```

```
form .button input:hover{
/* transform: scale(0.99); */
    background: linear-gradient(-135deg, #71b7e6, #9b59b6);
}
```

```
@media(max-width: 584px){
.container{  max-
width: 100%;
} form .user-details .input-
box{  margin-bottom:
15px;  width: 100%;
}
    form .category{
width: 100%;
}
    .content form .user-details{
max-height: 300px;  overflow-y:
scroll;
}
    .user-details::-webkit-scrollbar{
width: 5px;
}
}
    @media(max-width: 459px){
.container .content .category{  flex-
direction: column;
}
}
```

Search.css

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;
600;700&display=swap');
```

```
*{
margin: 0;
padding: 0;
box-sizing:
border-box;
    font-family: 'Montserrat', sans-serif;
}
```

```
body{
    background: #e0dbef;
}
```

```
.search_box{
background: #ffffff;
position: relative;
padding: 15px;  border-
radius: 50px;  display:
flex;
}
```

```
.search_box .search_btn{
width: 50px;  height:
50px;  border-radius:
50%;
    background: linear-gradient(135deg, #71b7e6, #9b59b6);
display: flex;  justify-content: center;  align-items:
center;  color: #fff;  margin-right: 15px;  cursor:
pointer;
}
```

```
.search_box .input_search{
outline: none;
    border: 0;
    background: linear-gradient(135deg, #71b7e6, #9b59b6);
```

```
border-radius: 50px;
padding: 15px 20px;
width: 500px;
height: 50px; color:
#fff;
}
```

```
::placeholder {
color: #fff;
}
```

```
::-webkit-input-placeholder {
color: #fff;
}
```

```
:-ms-input-placeholder {
color: #fff;
}
```

```
table{ width:
400px; } th{
text-align: center;
} table, th, td{ border:
1px solid #000;
}
```

```
.centre{ margin-
left: auto; margin-
right: auto; text-
align: center;
}
```

```
#sidebar-wrapper { min-
height: 100vh; margin-left: -
15rem;
-webkit-transition: margin 0.25s ease-out;
-moz-transition: margin 0.25s ease-out; -o-
transition: margin 0.25s ease-out;
transition: margin 0.25s ease-out;
```

```
}
```

```
#sidebar-wrapper .sidebar-heading {  
padding: 0.875rem 1.25rem;    font-  
size: 1.2rem;  
}
```

```
#sidebar-wrapper .list-group {  
width: 15rem;  
}
```

```
#page-content-wrapper {  
min-width: 100vw;  
}
```

```
#wrapper.toggled #sidebar-wrapper {  
margin-left: 0;  
}
```

```
#menu-toggle {  
cursor: pointer;  
}
```

```
.list-group-item {  
border: none; padding:  
20px 30px;  
}
```

```
.list-group-item.active {  
background-color: transparent;  
color: var(--main-text-color);  
font-weight: bold; border:  
none;  
}
```

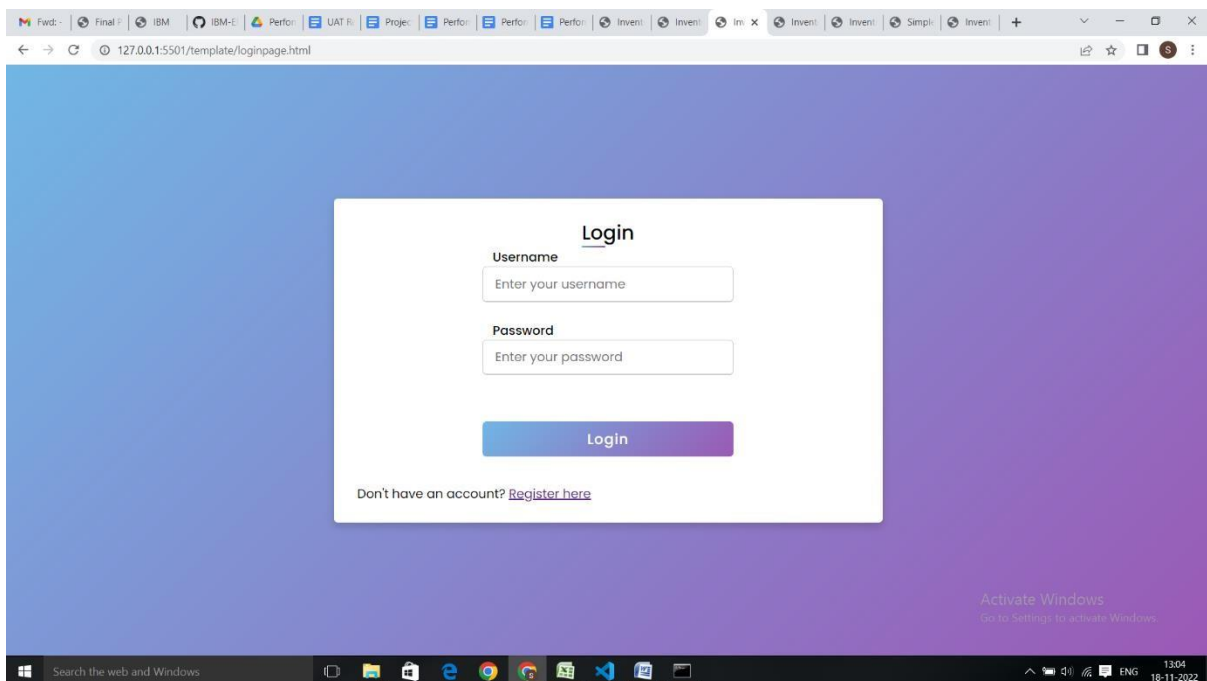
```
@media (min-width: 768px) {  
#sidebar-wrapper { margin-  
left: 0;  
}
```

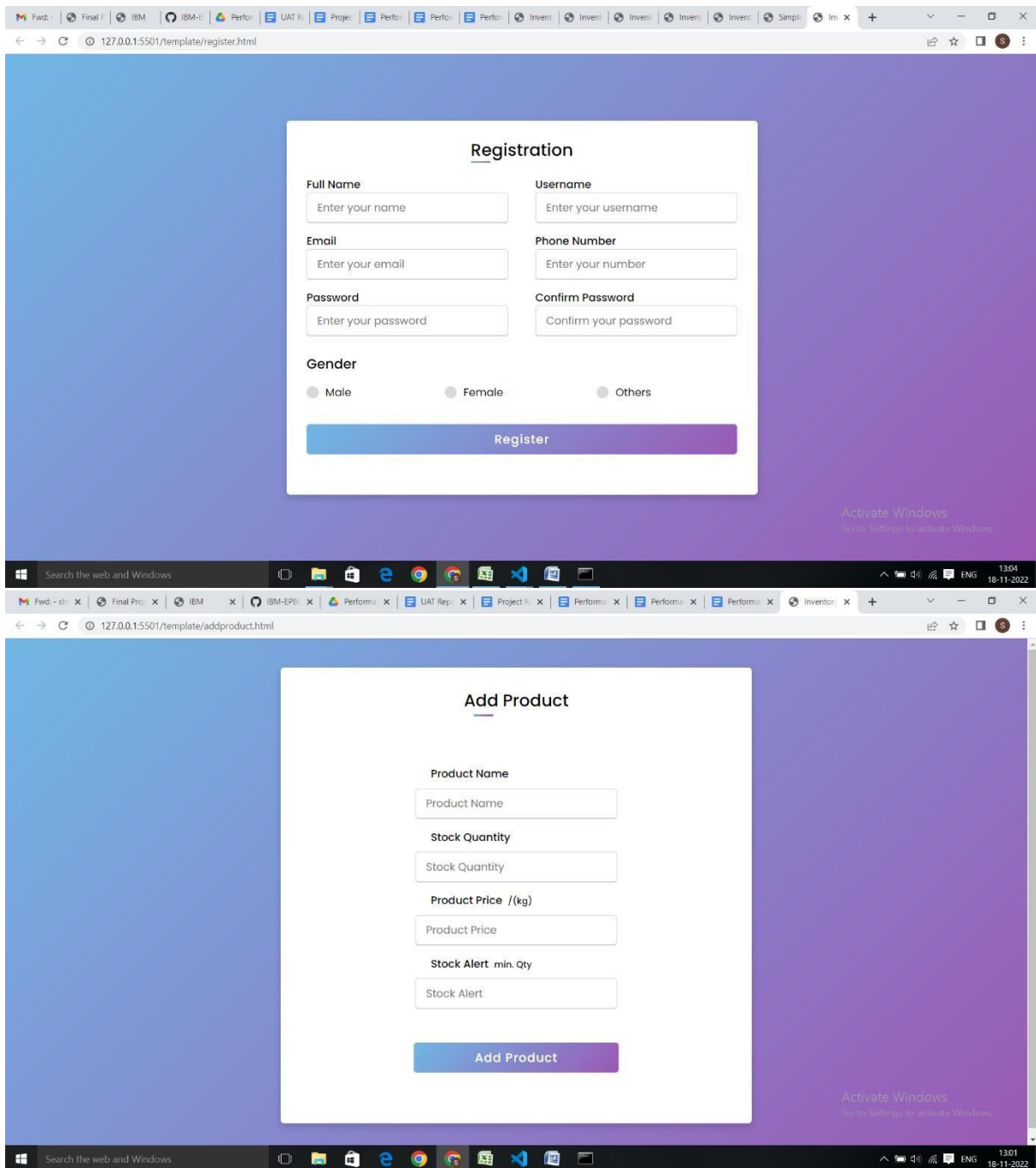
```
#page-content-wrapper {    min-  
width: 0;  
    width: 100%;  
}
```

```
#wrapper.toggled #sidebar-wrapper {    margin-  
left: -15rem;  
    }  
}
```

7.2 Feature 2

OUTPUT SCREENSHOTS:





127.0.0.1:5501/template/billing.html

BILLING

Invoice No :

Enter No

Bill Date:

Enter Date

Product Name :

Enter Product

Product Quantity:

Enter Quantity

Price:

Enter Price

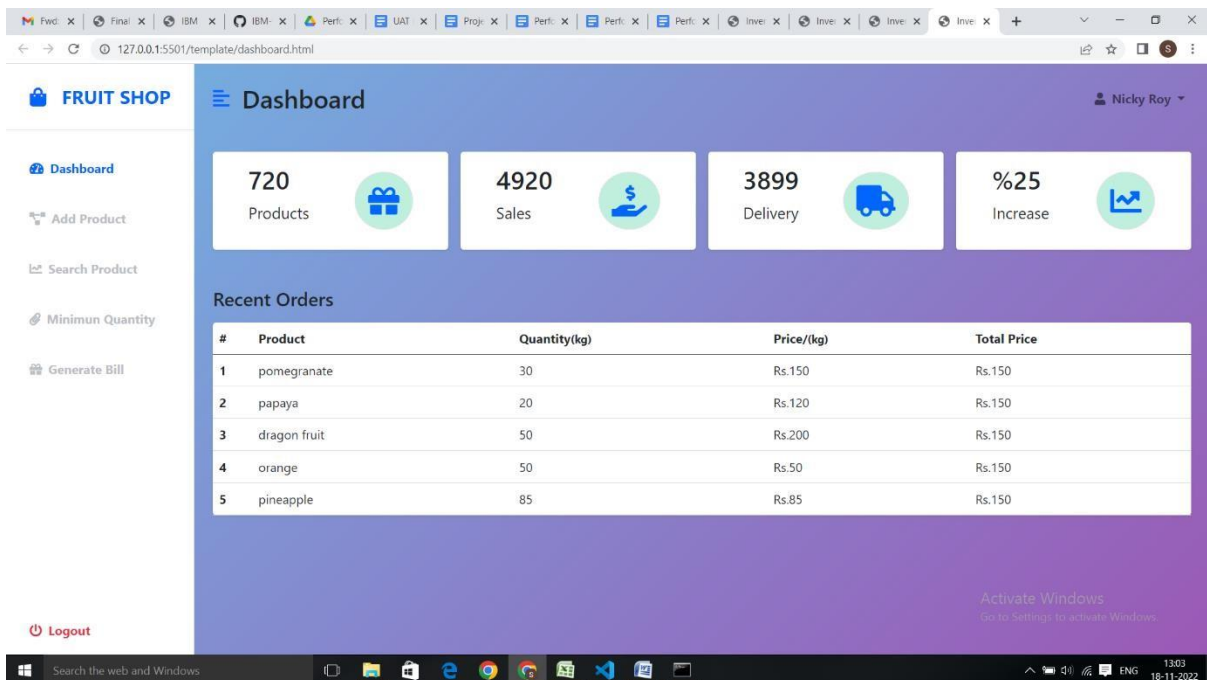
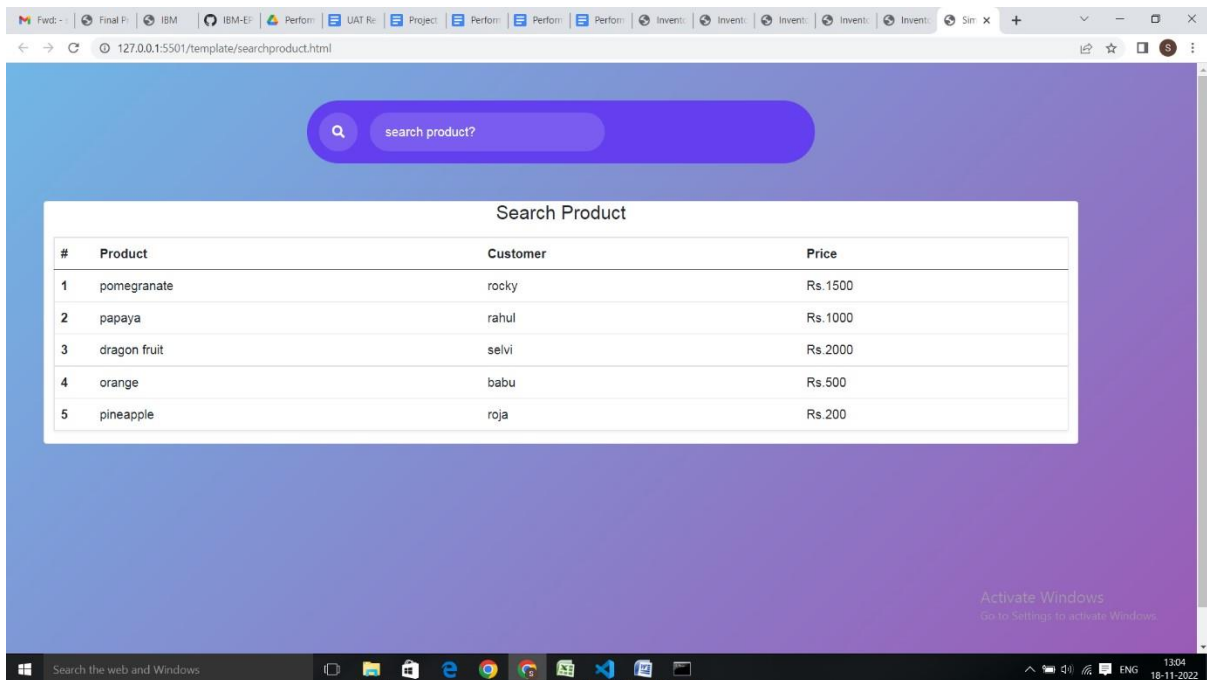
Save

Cancel

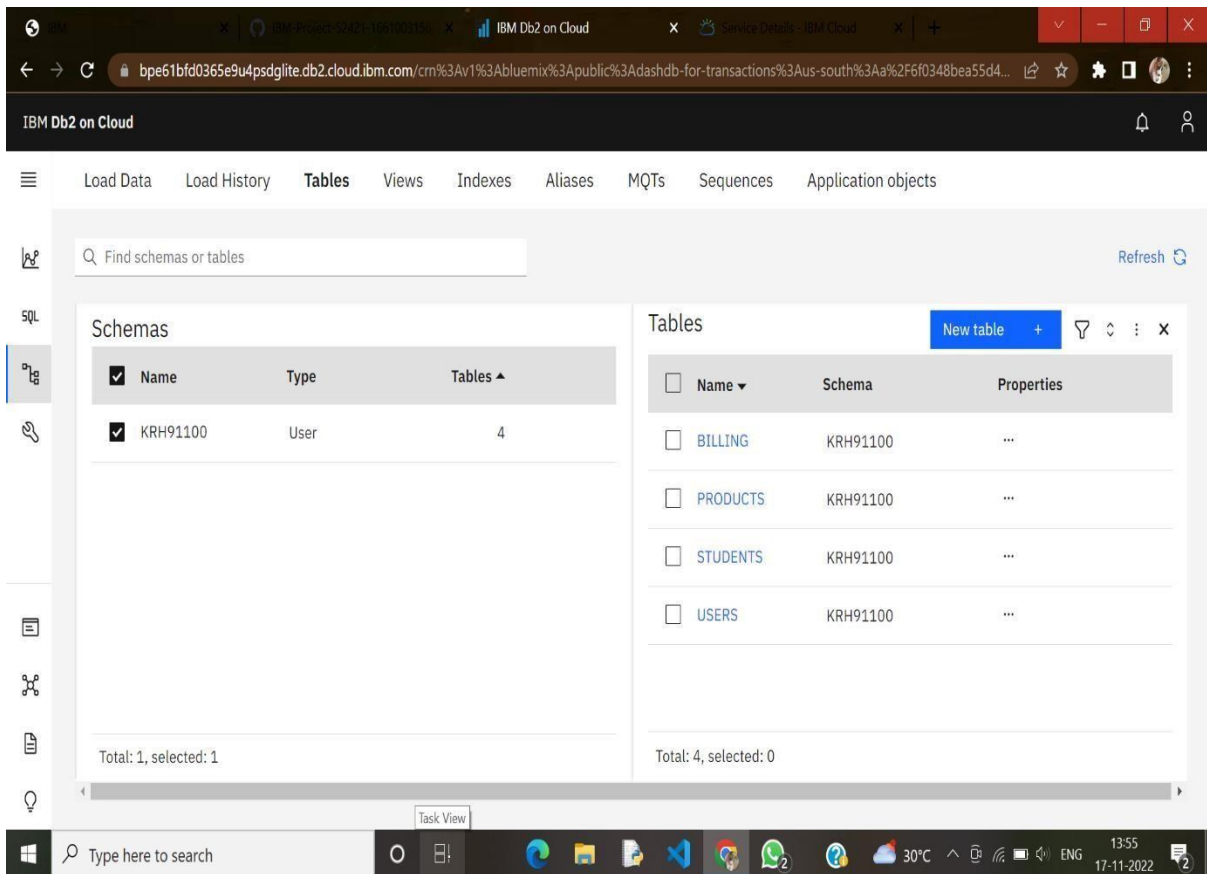
127.0.0.1:5501/template/minimum.html

Minimum Quantity List

#	Product	Min. Quantity	Present Quantity
1	pomegranate	5	5
2	papaya	5	2
3	dragon fruit	5	5
4	orange	5	4
5	pineapple	5	1



7.3 Database schema



Source code:

```
dsn_hostname = "9938aec0-8105-433e-8bf9-
0fbb7e483086.c1ogj3sd0tgu0lqde00.databases.appdomain.cloud" # e.g.:
"54a2f15b-5c0f-46df-8954-
7e38e612c2bd.c1ogj3sd0tgu0lqde00.databases.appdomain.cloud"
dsn_uid = "krh91100"      # e.g. "abc12345" dsn_pwd =
"gSPyVtDpdim5wKGL"      # e.g. "7dBZ3wWt9XN6$o0J"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "bludb"      # e.g. "BLUDB" dsn_port
= "32459"                  # e.g. "32733" dsn_protocol =
"TCPIP"                   # i.e. "TCPIP" dsn_security = "SSL"
#i.e. "SSL"

dsn = (
```

```

"DRIVER={0};"
"DATABASE={1};"
"HOSTNAME={2};"
"PORT={3};"
"PROTOCOL={4};"
"UID={5};"
"PWD={6};"

"SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname,
dsn_port, dsn_protocol, dsn_uid, dsn_pwd,dsn_security)

print(dsn)

try
:
    conn = ibm_db.connect(dsn, "", "")    print ("Connected to database: ",
dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)

except:    print ("Unable to connect: ", ibm_db.conn_errormsg()
)

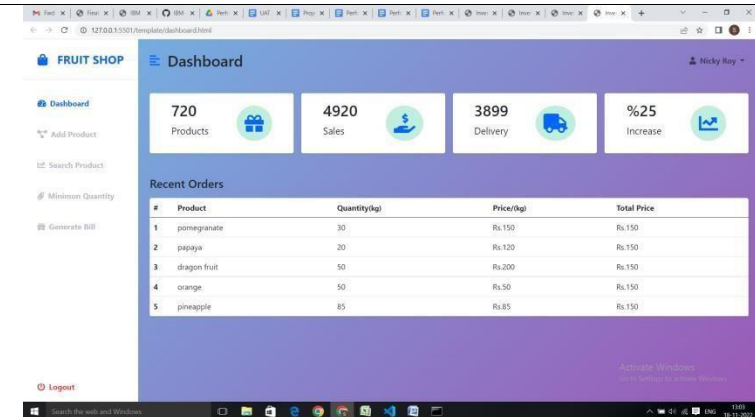
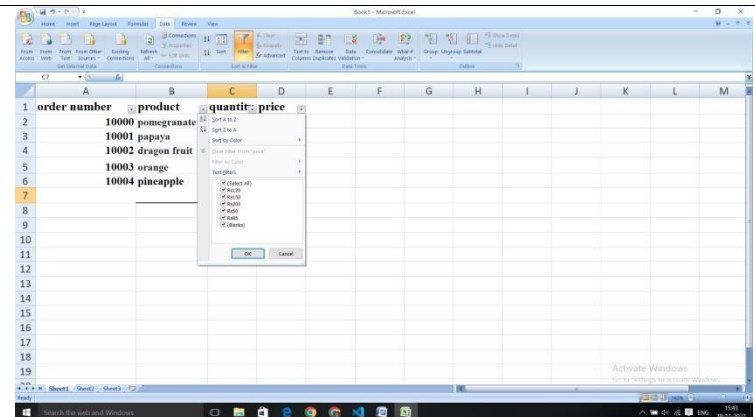
select="select * from users"
stm=ibm_db.exec_immediate(conn,select)
while ibm_db.fetch_row(stm) != False:

    print(ibm_db.result(stm,1))

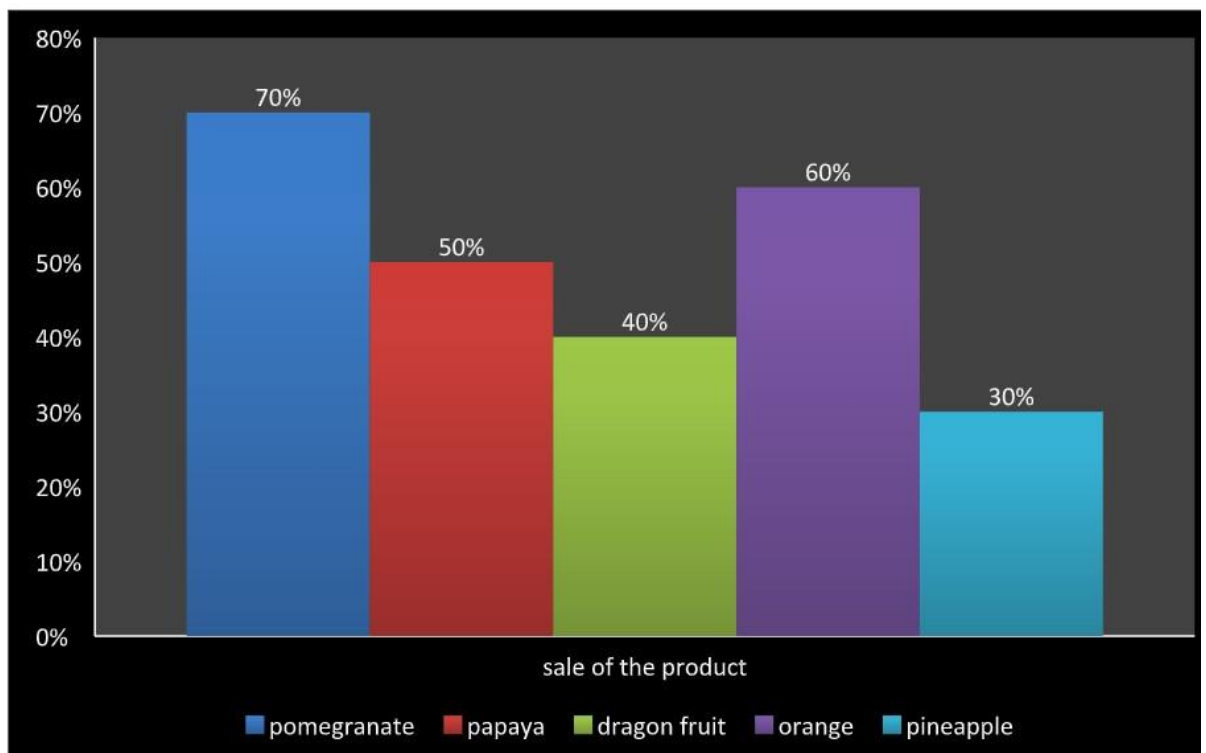
```

8.TESTING

8.1 Test cases

S.No.	Parameter	Screenshot / Values
1.	Dashboard design	
2.	Data Responsiveness	<ul style="list-style-type: none">• Responds quickly to unpredictable demand.• Have a lower utilization rate.• Utilise an excess buffer capacity.• Invest in lead time reduction.
3.	Amount Data to Rendered (DB2 Metrics)	8
4.	Utilization of Data Filters	
5.	Effective User Story	<ul style="list-style-type: none">• Dashboard is used to accessing the other pages.• Add product page is to add the product they need.• Minimum quantity page is to view the present quantity and minimum quantity of the product they need.• Search product page is used to search the product quickly.

6.	Descriptive Reports	Item number	Description	Unit of stock	Unit price	Total price
		10001	pomegranate	20	Rs100	Rs2000
		10002	papaya	0	Rs80	0
		10003	dragon fruit	25	Rs200	Rs2500
		10004	orange	40	Rs90	Rs3600
		10005	pineapple	0	Rs70	0



8.2 User Acceptance Testing

Step	Procedures	Expected Result	Result
1	Insert admin, username, and password	Save the insert data into database	Success
2	Insert correct username, password for login	Verify the admin	Success
3	Click 'Register,' 'Login' button	Application redirect admin to Login page after register and Main page after login	Success
4	Repeat step 2 and 3 for login using false username, password	Application display error message	Success
5	Update Admin Account	New update data saved into database	Success
6	Log Out Account	Log out redirected to Login page	Success
Precondition		No credentials are currently login	
Post-condition		New and updated Admin name, username, and password saved in	

Acceptance Checklist - Word

FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW DEVELOPER Ivan Walsh K

[Project Name] [Document Name - Version Number]

Acceptance Checklist Template

[Use this checklist to provide guidance to testers when identifying and planning software acceptance activities. This checklist captures acceptance management activities to be performed throughout the development life-cycle. Its objectives are to verify the software meets user's requirements, is fully operational, obtain the owner's acceptance, and transfer responsibility to the owner.]

Acceptance	Reference	Yes/No
User Training has been conducted.	Conduct User Training	Yes/No
System and Data Conversion has been performed as per Conversion Plan.	Installation Activities	Yes/No
Installation site were inspected in accordance with Installation Plan.	Installation Activities	Yes/No
Installation has been coordinated with System Owner, Operations, Support <u>etc.</u>	Installation Activities	Yes/No
Modifications to the physical installation environment are completed.	Installation Activities	Yes/No
Hardware has been inventoried and tested.	Installation Activities	Yes/No
Data-loads and Data Conversion have been installed and executed.	Installation Activities	Yes/No
Software has been installed & tested on the hardware platform	Installation Activities	Yes/No
Problems and corrective action are documented.	Conduct Installation Tests	Yes/No
Equipment and software retested after repairs, replacements, or modifications.	Conduct Installation Tests	Yes/No
User Training has been conducted.	Conduct User Training	Yes/No
System and Data Conversion has been performed as per Conversion Plan.	Installation Activities	Yes/No

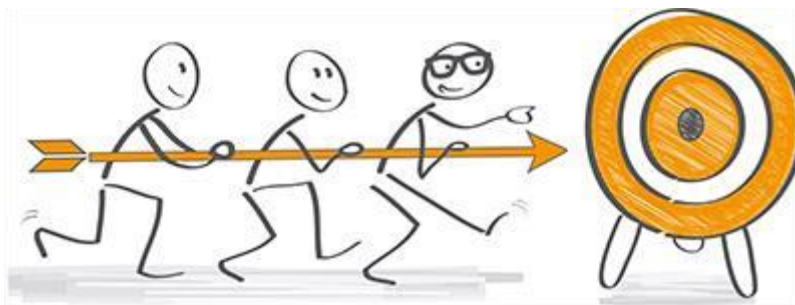
© [Name of Company] 1 | Page

PAGE 1 OF 3 550 WORDS ENGLISH (UNITED STATES) 70%

9.RESULT

9.1 Performance metrics

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



- Weeks on Hand. ...
- Inventory Turnover Rate. ...
- Days on Hand. ...
- Stock to Sales Ratio. ...
- Sell-through Rate. ...
- Backorder Rate. ...
- Accuracy of Forecast Demand. ...
- Rate of Return.

10.ADVANTAGE AND DISADVANTAGE

Advantage:

- To maintain the right amount of stocks
- To a more organized warehouse
- It saves time and money
- Improves efficiency and productivity
- A well-structured inventory management system leads to improved customer retention:
- It leads It helps Avoid lawsuits and regulatory fines

- Schedule maintenance
- Reduction in holding costs
- Flexibility **Disadvantage:**
- Bureaucracy
- Impersonal touch
- Production problem
- Increased space is need to hold the inventory
- Complexity

11.conclusion

In conclusion As you can see the importance of inventory management is very serious, it is one of the most important aspects of any business. The aspect of this part of the business is whether or not you can satisfy the demand of your customers if you aren't sure if you have all the materials available to make the final product Without having the proper inventory management they would not be able to supply their customers with their ordered ambulance. And this product is what their entire business is based on, so it is of great importance When they are choosing from the different types of programs or automated systems to help with keeping records accurate, needs to keep in mind that the customer is not concerned with which materials are needed to complete the finished product, but the product is operating as promised based on the contract. In addition, the plans for the maintenance of having proper inventory levels need to be in place and also adjusted when the company grows and as the business dictates implements the new suggestions they will be on the right track to having a well established business

12. FUTURE SCOPE

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

13. APPENDIX

Source Code

```

from flask import Flask,render_template,request,redirect, url_for,flash
app = Flask(__name__) import ibm_db
from flask_login import login_user, current_user, logout_user,
login_required,LoginManager,UserMixin import
datetime
from sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import Mail


dsn_hostname = "9938aec0-8105-433e-8bf9-
0fbb7e483086.clogj3sd0tgu0lqde00.databases.appdomain.cloud" # e.g.:
"54a2f15b-5c0f-46df-8954-
7e38e612c2bd.clogj3sd0tgu0lqde00.databases.appdomain.cloud" dsn_uid
= "krh91100"      # e.g. "abc12345"
dsn_pwd = "gSPyVtDpdim5wKGL"    # e.g. "7dBZ3wWt9XN6$o0J"


dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "bludb"           # e.g. "BLUDB"
dsn_port = "32459"               # e.g. "32733"
dsn_protocol = "TCPIP"           # i.e. "TCPIP"
dsn_security = "SSL"             #i.e. "SSL"


dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname,
dsn_port, dsn_protocol, dsn_uid, dsn_pwd,dsn_security)

print(dsn)

try:

```



```
conn = ibm_db.connect(dsn, "", "")    print ("Connected to database: ",
dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)
```

```
except:    print ("Unable to connect: ",
ibm_db.conn_errormsg() )
```

```
SECRET_KEY = 'Vibinprakash'
```

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

```
    name=request.form.get('user')
password=request.form.get('password')
print(name,password)
    sql = "SELECT * FROM users WHERE user_name =? AND password=?"
stmt = ibm_db.prepare(conn, sql)    ibm_db.bind_param(stmt,1,name)
ibm_db.bind_param(stmt,2,password)    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print (account)
if account:
```

```
    return redirect(url_for('dashboard'))
```

```
else:
```

```
    msg='invalid user name and password'
    return render_template('loginpage.html',msg=msg)
```

```
else:
```

```
    return render_template('loginpage.html')
```

```
@app.route("/register",methods=['GET', 'POST']) def
register():
```

```
    if request.method == 'POST':
name=request.form.get('full')
user_name=request.form.get('user')
email=request.form.get('email')
phone=request.form.get('phone')
```

```

password=request.form.get('password')
confirm=request.form.get('confirm')
    print(confirm)        if
password==confirm:

    sql ="SELECT id FROM users ORDER BY ID DESC limit 1"
    stm=ibm_db.exec_immediate(conn,sql)
while ibm_db.fetch_row(stm) != False:
count=ibm_db.result(stm,0)
    print(count)

    insert=f"insert into users values ({int(count)+1}, '{name}',
'{user_name}', '{email}', '{password}', '{phone}')"
table=ibm_db.exec_immediate(conn,insert)
    return redirect(url_for('home'))

else:
    msg='invalid user name and password'
return render_template('register.html',msg=msg)

else:
    return render_template('register.html')

@app.route("/searchproduct",methods=['GET', 'POST']) def
searchproduct():
    if request.method == 'POST':
Product=request.form.get('search')
        sql = "SELECT * FROM products WHERE product =?"
        stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,Product)
ibm_db.execute(stmt)        account =
ibm_db.fetch_assoc(stmt)
        print (account)        return
render_template('searchproduct.html',product=account)    else:
        return render_template('searchproduct.html')

@app.route("/viewbill",methods=['GET', 'POST'])
def viewbill():

```

```

sql ="SELECT * FROM billing"

stmt = ibm_db.exec_immediate(conn, sql)

bill=[]
amount=0

while ibm_db.fetch_row(stmt) != False:
dic=dict()
dic['invoice']=ibm_db.result(stmt, 1)
dic['product']=ibm_db.result(stmt, 3)
dic['price']=ibm_db.result(stmt, 4)
price=ibm_db.result(stmt, 4)
dic['quantity']=ibm_db.result(stmt,5)
quantity=ibm_db.result(stmt,5)
dic['total']=int(price)*int(quantity)
total=int(price)*int(quantity)      amount
+=total      bill.append(dic)

print(bill)
print(amount)
return render_template('viewbill.html',datas=bill)

@app.route("/minimum",methods=['GET', 'POST']) def
minimum():
    sql ="SELECT * FROM products"
    # stmt = ibm_db.prepare(conn, sql)
    # ibm_db.bind_param(stmt,1,name)    #
    ibm_db.bind_param(stmt,2,password)
    stmt = ibm_db.exec_immediate(conn, sql)

    datas=[]    while
ibm_db.fetch_row(stmt) != False:
dic=dict()
dic['product']=ibm_db.result(stmt, 0)
dic['stock']=ibm_db.result(stmt, 1)
dic['price']=ibm_db.result(stmt, 2)
dic['alert']=ibm_db.result(stmt, 3)
datas.append(dic)    print(datas)
    # ibm_db.execute(stmt)
    # account = ibm_db.fetchall(stmt)

```

```

        # print (account)
    return render_template('minimum.html',datas=datas)

@app.route("/dashboard",methods=['GET', 'POST']) def
dashboard():
    sql="SELECT * FROM products"
    stmt = ibm_db.exec_immediate(conn, sql)

    datas=[]  low=0  count=0  while
ibm_db.fetch_row(stmt) != False:
    dic=dict()
    dic['product']=ibm_db.result(stmt, 0)
    dic['stock']=ibm_db.result(stmt, 1)
    stock=ibm_db.result(stmt, 1)
    dic['price']=ibm_db.result(stmt, 2)
    dic['alert']=ibm_db.result(stmt, 3)
    alert=ibm_db.result(stmt, 3)    if
int(stock)< int(alert):        low += 1
    datas.append(dic)        count+=1
    print(datas)
    sql="SELECT * FROM billing"

    stmt = ibm_db.exec_immediate(conn, sql)

    bill=[]
    amount=0  bill_count=0  while
ibm_db.fetch_row(stmt) != False:
    dic=dict()
    dic['invoice']=ibm_db.result(stmt, 1)
    dic['product']=ibm_db.result(stmt, 3)
    dic['price']=ibm_db.result(stmt, 4)
    price=ibm_db.result(stmt, 4)
    dic['quantity']=ibm_db.result(stmt,5)
    quantity=ibm_db.result(stmt,5)
    dic['total']=int(price)*int(quantity)
    total=int(price)*int(quantity)    amount
+=total
    bill_count+=1
    bill.append(dic)

```

```

    print(bill)
print(amount)    return
render_template('dashboard.html',datas=datas,low=low,amount=amount,count=
count,bill_count=bill_count)

```

```

@app.route("/billing",methods=['GET', 'POST'])
def billing():
    date=datetime.datetime.today().date()    if
    request.method == 'POST':
    invoice=request.form.get('invoice')
    date=request.form.get('date')
    product=request.form.get('product')
    quantity=request.form.get('quantity')
    price=request.form.get('price')
        insert=f"insert into billing values ( {invoice[-1]}, '{invoice}', '{date}',
'{product}',          {int(quantity)},          {int(price)})"
    table=ibm_db.exec_immediate(conn,insert)          sql =
"SELECT * FROM products WHERE product =?"
        stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,product)
    ibm_db.execute(stmt)          account =
    ibm_db.fetch_assoc(stmt)          try:
        count=account['STOCK']
    alert=account['Alert']
    update_count=int(count)-int(quantity)          if
    int(update_count) < int(alert):
        print('email code')

        #write sendgrid email code here

    message = Mail(

        from_email='shajalraj333@gmail.com',
    to_emails='shijonida2@gmail.com',          subject='Sending
with Twilio SendGrid is Fun',

        html_content='<strong>and easy to do anywhere, even with
Python</strong>')
        try:
sg

```

```
=SendGridAPIClient('SG.ugbhjeYMQkKqLEAXNHd3ig.ovmpPK1LNdf_oXy  
bocXoEx qsjavVbSEefk0NvjqCEJs')
```

```
    response = sg.send(message)  
print(response.status_code)  
print(response.body)  
print(response.headers)          except  
Exception as e:  
    print(e)
```

```
    update=f"UPDATE products SET  stock = {update_count} WHERE  
product = '{product}'"
```

```
    table=ibm_db.exec_immediate(conn,update)  
    return redirect(url_for('dashboard'))  
except:
```

```
    return redirect(url_for('dashboard'))
```

```
else:
```

```
    return render_template('billing.html',date=date,invoice=invoice_no())
```

```
@app.route("/addproduct",methods=['GET', 'POST']) def  
addproduct():
```

```
    if request.method == 'POST':  
Product=request.form.get('Product')  
    Stock=request.form.get('Stock')  
    Price=request.form.get('Price')  
Alert=request.form.get('Alert')  
print(Product,Stock,Price,Alert)  
    insert=f"insert into products values ( '{Product}', {int(Stock)},  
{int(Price)}, {int(Alert)})"  
    table=ibm_db.exec_immediate(conn,insert)
```

```
    return redirect(url_for('dashboard'))
```

```
else:
```

```
    return render_template('addproduct.html')
```

```
def invoice_no():
```

```
sql ="SELECT id FROM billing ORDER BY ID DESC limit 1"
stm=ibm_db.exec_immediate(conn,sql) while
ibm_db.fetch_row(stm) != False: count=ibm_db.result(stm,0)
if count:
    return f'bill00{int(count)+1}'
else: return 'bill001'
```

```
app.run(debug=True)
```

GitHub & Project Demo Link:

Github link: <https://github.com/IBM-EPBL/IBM-Project-462491664791837>

Demo link: <https://youtu.be/crv36BBvJzo>