

# PROJECT REPORT

## INVENTORY MANAGEMENT SYSTEM

**TEAM ID : PNT2022TMID53370**

### TEAM MEMBERS

NAME	ROLL NUMBER
PRADEEP KUMAR R	2127190501088
KEERTHANA SRI S	2127190501057
LAKSHMAN G	2127190501061
LAKSHMI GAYATHRI S	2127190501062

# INDEX

## **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 hold 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.LITERATURE SURVEY

**Anish Singh Maharjan, Mandip Humagain** stated that this project is aimed at developing a desktop-based application named Inventory Management System for managing the inventory system of any organization. The Inventory Management System (IMS) refers to the system and processes to manage the stock of organization with the involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update 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 revolving 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 This application is not suitable for those organization where there is large quantity of product and different level of warehouses wholesaler when it is time to record. Inventory Management System is also an important means of automatically tracking large shipment. An automated Inventory Management System helps to minimize the errors while recording the stock.

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

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

### 2.1 Existing problem

In the application stated by Anish Singh Maharjan and Mandip Humagain is not suitable for those organization where there is large quantity of product and different level of warehouses.

The application stated by MS. Dhruvika Patel contains Manual Errors at the time of entering the data can't be check, only the validation required w.r.t proposed system is checked.

It is difficult to find records due to the file management system in the application developed by Raj Kumar and Neelesh Kumar Singh

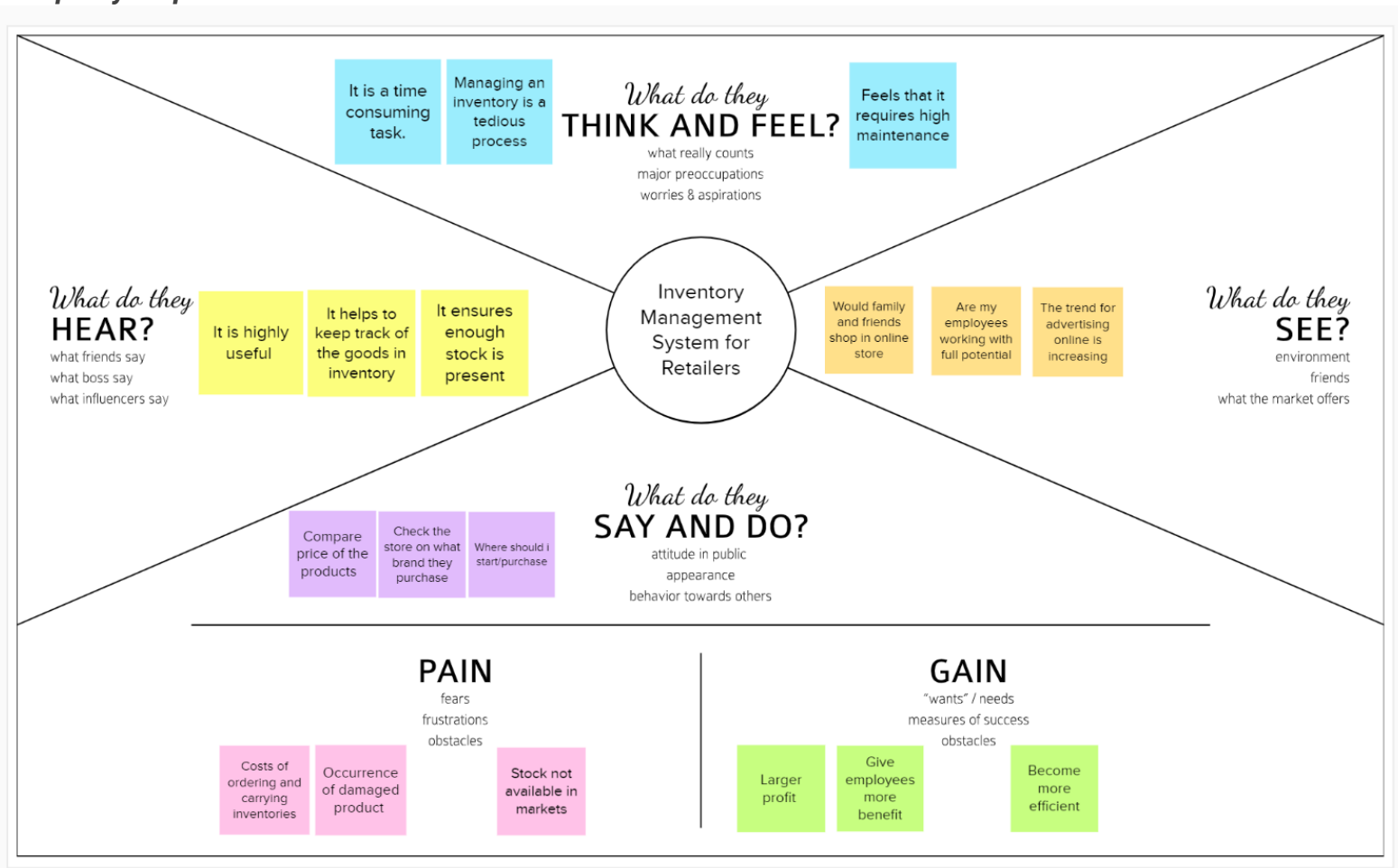
## 2.2 References :

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., Moinezhadeh, 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. **Agarwal, 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)

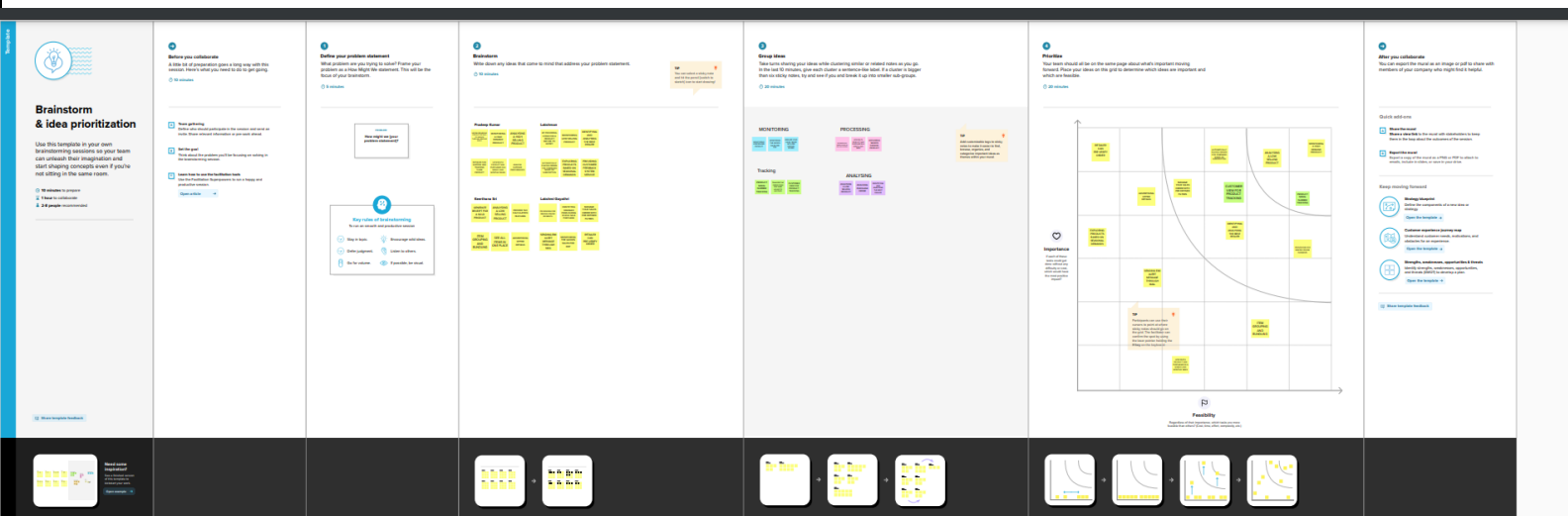
## 2.3 Problem Statement Definition :

Retail inventory management is ensuring you carry merchandise that shoppers want, with neither too little nor too much on hand. By managing inventory, retailers meet customer demand without running out of stock or carrying excess supply. To create effective retail inventory management that results in lower costs and a better understanding of sales patterns. It must include tools and methods that give retailers more information on which to run their businesses. It should ask retailers to create their accounts by providing essential details. Retailers should be able to access their accounts by logging into the application. Once retailers successfully log in to the application they should be able to update their inventory details, also users will be able to add new stock by submitting essential details related to the stock. They should be able to view details of the current inventory. The System should automatically send an email alert to the retailers if no stock is found in their accounts. So that they can order new stock

### 3.1 Empathy Map :



### 3.2 Ideation & Brainstorming :



### 3.3 Proposed Solution :

S.NO	Parameter	Description
1	Problem Statement(Problem to be solved)	Retail store stock inventory analytics is implemented to analyze the historical sales data of a Brazilian top retailer. By deeply understanding the dataset, identifying pattern, relationships and connection using IBM cog nos 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 desktop based application named Inventory Management System for managing the inventory system of any organization. The Inventory Management System (IMS) refers to the system and processes to manage the stock of organization with the involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update 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 revolving 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 automated Inventory Management System helps to minimize the errors while recording the stock
3	Novelty / Uniqueness	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 back order, so you don't oversell products. Stock costs money until it sells. Carrying costs include storage handling and transportation fees, insurance-and employee salaries. 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
4	Social Impact / Customer Satisfaction	Customer satisfaction entirely depends on the effective supply chain management which is not 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

		manufacture 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. 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.
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
6	Scalability of the solution	The visualization of sales data makes the retailer to estimate accurate inventory to be maintained.

### 3.4 Problem Solution Fit :

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Who is your customer? i.e. working parents of 0-5 y.o. kids <div>Customer who wants to buy inventory</div>	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices. <div> <ul style="list-style-type: none"> <li>Time constraints</li> <li>Lack of money</li> <li>Emergency need of inventory</li> </ul> </div>	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking <div> <ul style="list-style-type: none"> <li>Retailer to maintain safety stock on hand.</li> <li>Have in track the buying and sales on inventory.</li> <li>Make customer hold on to their shop by providing quality products.</li> </ul> </div>	Explore AS, differentiate
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <div> <ul style="list-style-type: none"> <li>Too much customer and lack of inventory.</li> <li>Use accurate dataset for visualization.</li> <li>Get customer review.</li> </ul> </div>	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations. <div> <ul style="list-style-type: none"> <li>The problem is faced by customer if the retailer miss to keep the stock on hand that are needed for them.</li> <li>The retailer face issue as the sales is not the same all time.</li> <li>If retailer makes too inventory it leads to excess capital cost.</li> </ul> </div>	<b>7. BEHAVIOUR</b> <span>BE</span> What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) <div> <p>Directly Associated:</p> <ul style="list-style-type: none"> <li>Have direct contact with customer requirements.</li> <li>Regular feedback from customer.</li> </ul> <p>Indirectly Associated:</p> <ul style="list-style-type: none"> <li>Convince if out of stock.</li> <li>Make inventory to get available at earliest time possible.</li> </ul> </div>	
<b>3. TRIGGERS</b> <span>TR</span> What triggers customers to act? i.e. seeing their chavior installingsolar panels, reading about a more efficient solution in the news. <div>Review from neighbors about the product. Disappointment if out of stock.</div>	<b>10. YOUR SOLUTION</b> <span>SL</span> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer chavior. <div>The proposed solution is to visualize the sales data set using different visualization charts and to make an interactive dashboard to give insights on how much inventory should be maintained for making happy customers and not make loss of money due to excess inventory. Different category of data are been visualized using appropriate charts.</div>	<b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span> ONLINE What kind of actions do customers take online? Extract online channels from #7 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. <div> <p>Online:</p> <ul style="list-style-type: none"> <li>Online buying of inventory is simple and feasible</li> <li>User interactive website available.</li> </ul> <p>Offline:</p> <ul style="list-style-type: none"> <li>Direct review to retailer.</li> <li>Check quality of product before buying.</li> </ul> </div>	Identify strong TR & EM	
<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design. <div> <p>Before - Disappointed when out of stock Lose hope &amp; interest on shop.</p> <p>After - Happy with retailer stock maintenance Satisfied.</p> </div>				



## 4 REQUIREMENT ANALYSIS :

### 4.1 Functional Requirements :

Following are the functional requirements of the proposed solution

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Utilizing a Form for Registration signing up with Gmail Using a username and password to register
FR-2	User Confirmation	Email confirmation required Reassurance via OTP
FR-3	Sign In	Log in to the program using your Gmail account, username, and password.
FR-4	Dashboard	can see product information.
FR-5	Ordering	Put necessary items in a cart first and then place an order for them
FR-6	Restocking	increasing product orders when the supply is low.

### 4.2 Non Functional Requirements :

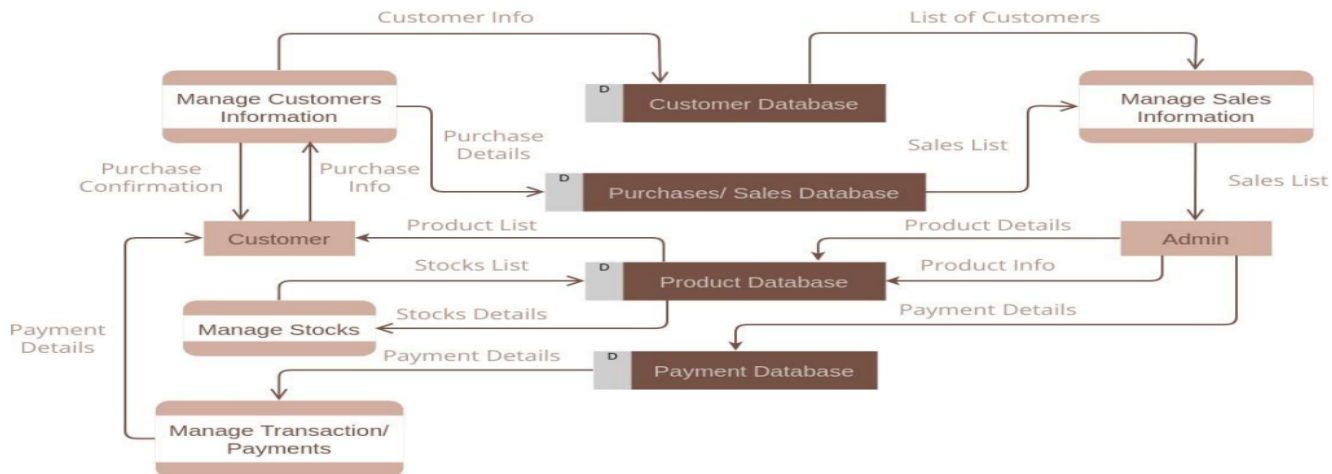
Following are the non-functional requirements of the proposed solution.

NFR NO	Non-Functional Requirement	Description
NFR-1	Usability	creating a learning curve into the site's design and development. having a user-friendly, straightforward website. Beautiful-looking website. making the website responsive for consumers on both desktops and mobile devices.
NFR-2	Security	Strong security is necessary to prevent hackers from accessing the accounts or data of authorised users. To demonstrate authentication and authorization, log in systems are utilised. Utilizing OTP can improve security. Cookies-based security mechanism for user authentication and enhanced website user experience.
NFR-3	Reliability	When the website is active, it should be able to manage the necessary number of users without slowing or causing any inconvenience to the user. While running the apps, there should be few mistakes. should be accessible even during disasters
NFR-4	Performance	This has the advantage of cutting down on the time needed for aisle and product searches, among other conveniences. It decreases expenses, saves time during restocking, and forecasts the top-selling goods. Due to the business's streamlined management system, it is more productive and profitable
NFR-5	Availability	To provide high availability of database servers and performances, this employs IBM DB2.

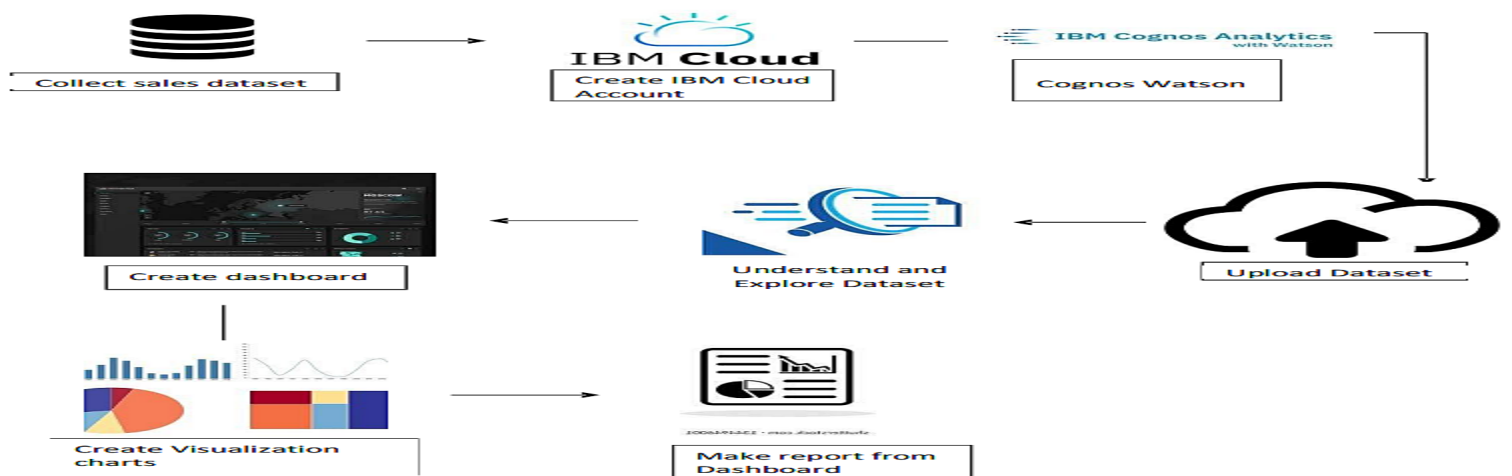
NFR-6	Scalability	As DB2 is highly scalable, the coding can be produced and developed efficiently and new features can be introduced easily. Reusing the code can be done to add any new features. IBM Container in Docker registry is used which is highly scalable.
-------	-------------	---

## 5 PROJECT DESIGN :

### 5.1 Dataflow 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
Retailer( Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I will be redirected to login page	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can verify the OTP number	Medium	Sprint-1
		USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
		USN-6	As a user,I can update stock in & out count details	Updation can be made through barcode scanning	High	Sprint-2
		USN-7	As a user,I can check the low stock details through alert message	Alert message can be received by registered mail	High	Sprint-1
		USN-8	As a user,I can check the total product details	I can view the value of total products in the stock	Medium	Sprint-2
		USN-9	As a user,I can check the high demand product details	I can update sales details of the products	High	Sprint-2
		USN-10	As a user,I can generate the invoice details	I can add incoming stock details	High	Sprint-1

--

## 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	● Create Flask Project ● Create IBM Cloud Account ● Install IBM Cloud CLI ● Docker CLI	Pradeep Kumar,Lakshman,Lakshmi Gayathri,Keerthana Sri	Review

		Installation • Create An Account In Sendgrid		
2	Implementing Web Application	• Create UI To Interact With Application	Pradeep Kumar	Review
3	Integrating SendGrid Service	• SendGrid Integration With Python Code	Pradeep Kumar	Review
4	Deployment of App in IBM Cloud	• Containerize The App • Upload Image to IBM Container Registry • Deploy in Kubernetes	Pradeep Kumar,Lakshman,Lakshmi Gayathri,Keerthana Sri	Review
5	Ideation Phase	• Literature Survey On The Selected Project & Information Gathering • Prepare Empathy Map • Ideation	Pradeep Kumar,Lakshman,Lakshmi Gayathri,Keerthana Sri	Review
6	Project Design Phase – I	• Proposed Solution • Problem Solution Fit • Solution Architecture	Pradeep Kumar,Lakshman,Lakshmi Gayathri,Keerthana Sri	Review
7	Project Planning Phase	• Prepare Milestone & Activity List • Sprint Delivery Plan	Pradeep Kumar,Lakshman,Lakshmi Gayathri,Keerthana Sri	Review
8	Project Development Phase	• Delivery Of Sprint-1 • Delivery Of Sprint-2 • Delivery Of Sprint-3 • Delivery Of Sprint-4	Pradeep Kumar,Lakshman,Lakshmi Gayathri,Keerthana Sri	Review

## 6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	5
		USN- 2	As a user, I can register for the	1	Medium	5





### Project Tracker, Velocity & Burndown Chart:

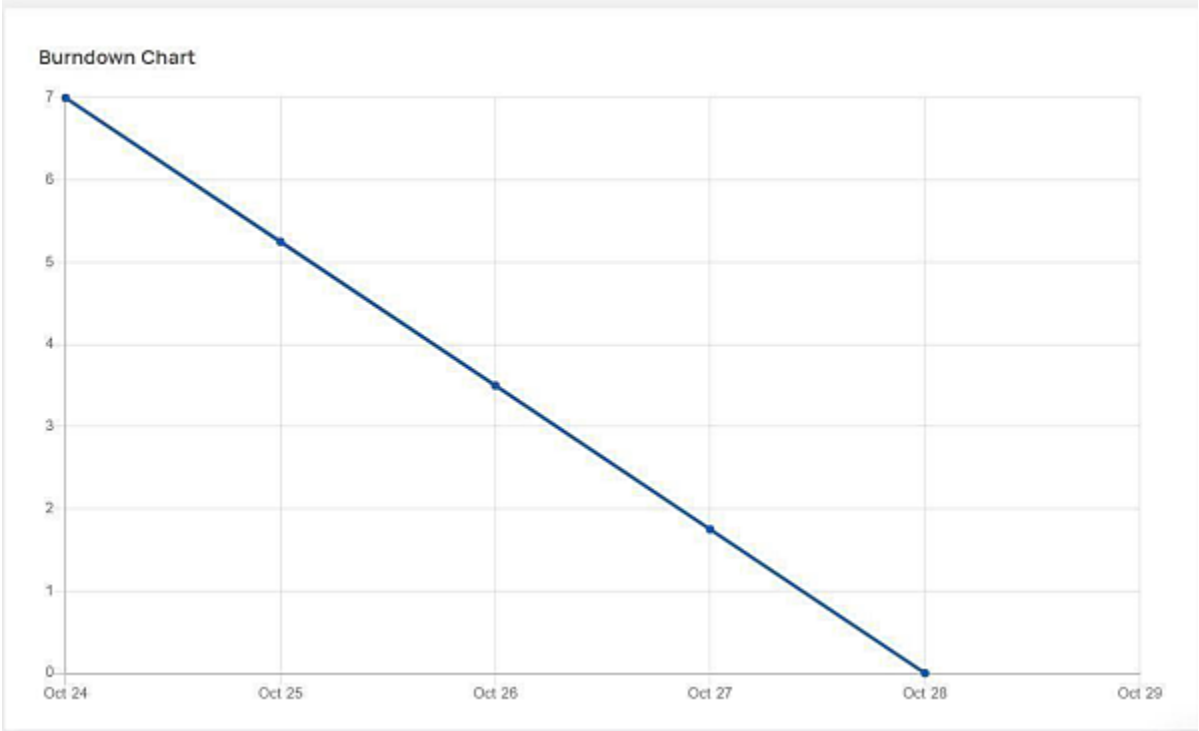
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	7	6 Days	24 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-2	9	6 Days	31 Oct 2022	08 Nov 2022	9	08 Nov 2022
Sprint-3	5	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

**Velocity:**

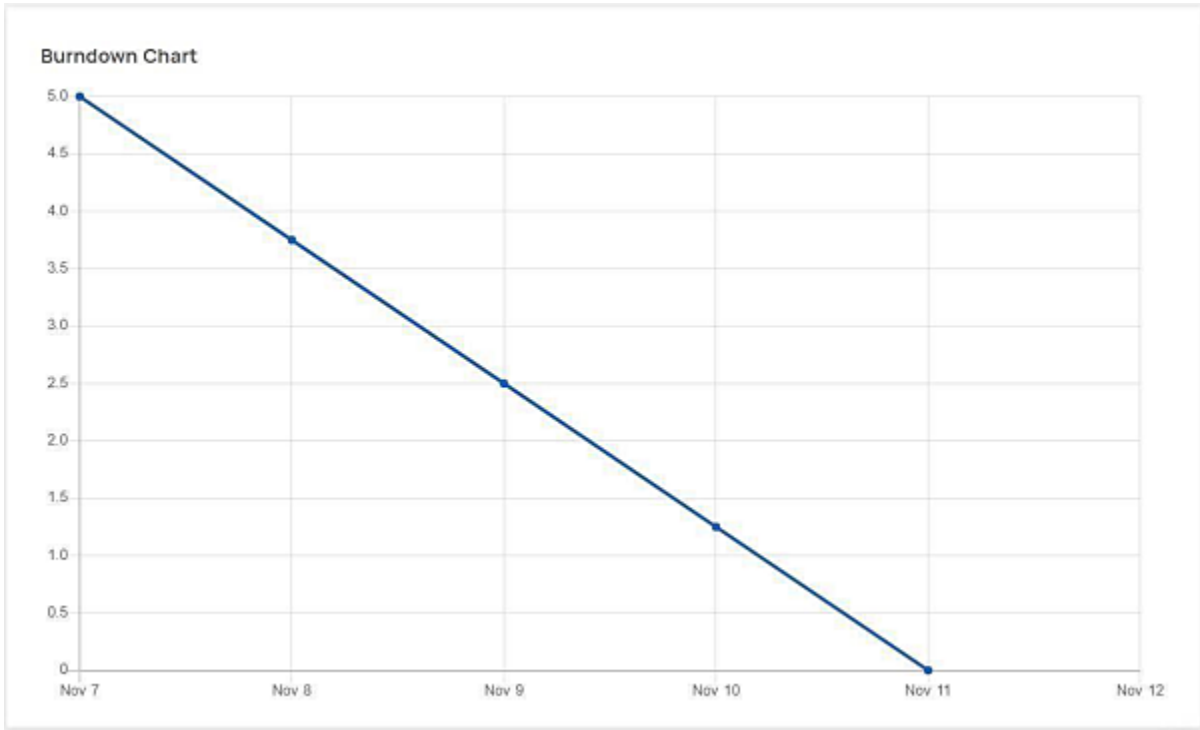
Sprints	Sprint Duration Velocity Actual Velocity	Sprints Sprint Duration Velocity Actual Velocity	Sprints Sprint Duration Velocity Actual Velocity
Sprint1	6	7	0.85
Sprint2	6	9	0.66
Sprint3	6	5	1.2
Sprint4	6	10	0.6



Burndown Chart:



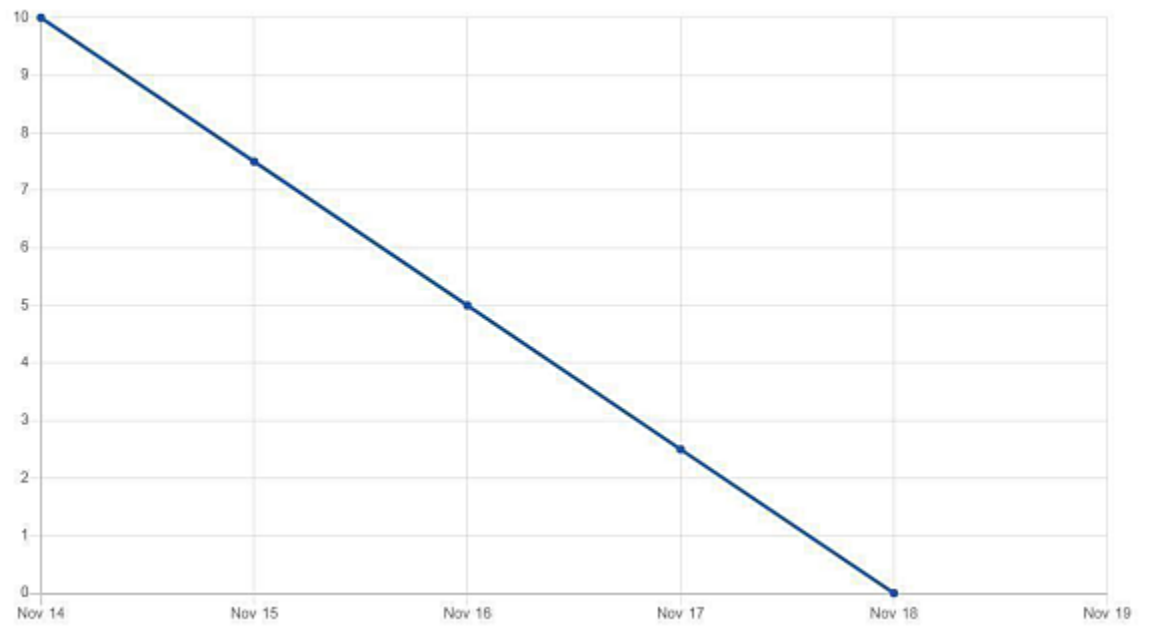
SPRINT1



SPRINT2



SPRINT3



SPRINT4

## 6.3 Reports from JIRA

Projects / Asisa Team

### AT board

TO DO 1 ISSUE

Time frame of sprint

LEARN SPRINT CREATION USING J...

AT-7

+ Create issue

IN PROGRESS

DONE ✓

GROUP BY: None

Quickstart

Projects / Asisa Team

### Roadmap

Give feedback Share Export

AT-5 Learn sprint creation using Jira

AT-6 Learn work item creation in Jira

+ Create Epic

Today Weeks Months Quarters

Quickstart

Projects / Asisa Team

### Roadmap

Give feedback Share Export

AT-5 Learn sprint creation using Jira

AT-6 Learn work item creation in Jira

AT-8 Login

AT-9 Register

AT-10 Dashboard

AT-11 Add product

AT-12 Billing

+ Create Epic

Today Weeks Months Quarters

Quickstart

## 7. CODING AND SOLUTIONING

```
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$0J"
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
templ
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://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-awesome.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
secondtext fw-bold active"><i class="fas fa-project-diagram me2"></i> Add Product</a> <a href="{{ url_for('searchproduct')
}}" class="list-group-item listgroup-item-action bg-transparent second-text fwbold"><i
class="fas fa-chart-line me-2"></i> Search Product</a>
<a href="{{ url_for('minimum') }}" class="list-group-item list-groupitemaction 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"
```

```
<div>  
    <div class="input-box">  
        <span class="details">Product  
Price<small>&nbsp;/<(kg)</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 class="button">  
        <input type="submit" value="Add Product">  
    </div>  
</form>  
</div>  
</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-itemaction 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-itemaction 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>
</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" databstoggle="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://cdn.jsdelivr.net/npm/font-awesome@5.15.3/css/all.min.css"
rel="stylesheet" />
<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 listgroupitem-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 me2"></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-groupitemaction 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" databstoggle="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 justifycontentaround 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 justifycontentaround 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 justifycontentaround 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 justifycontentaround 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>
</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-itemaction 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-itemaction 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" databstoggle="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-itemaction 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>
</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" databstoggle="collapse"
```

```
data-bs-target="#navbarSupportedContent"
aria-controls="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" cellspacing="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 listgroupitem-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-groupitemaction 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>
</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" databstoggle="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; fontfamily: 'Poppins',sans-serif; }
.container{ display: flex;
justify-content: center; alignitems: center; padding:25px;
min-height:calc( 100vh - 88px) ;
line-height: 2rem;
}
.container .title{ fontsize: 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; borderbottom-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%; borderradius: 5px;
border: none;
color: #fff; fontsize: 18px; fontweight: 500; letterspacing: 1px;
cursor: pointer;
transition: all 0.3s
ease;
background: linear-gradient(135deg, #71b7e6, #9b59b6); marginleft: 25%;
}

form .button input:hover{
/* transform: scale(0.99); */
background: linear-gradient(-135deg, #71b7e6, #fff23d);
}

@media(max-width: 584px){
.container{ maxwidth: 100%;
} form .user-details .inputbox{ 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; texttransform: 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; texttransform: uppercase; textalign: 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; borderradius: 10px;
}
.container form .submit-btn1{
width: 100%; padding:12px;
font-size: 17px;
background: #ff3300;
color:#fff; margintop: 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; flexdirection: row; justify-content:
space-between; gap:20px;
}
input .danger{ backgroundcolor: 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 { minheight: 100vh; margin-left: -  
15rem;  
-webkit-transition: margin 0.25s ease-out;  
-moz-transition: margin 0.25s ease-out; -otransition: 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; fontfamily: '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{ fontsize: 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: spacebetween; 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: spacebetween;
}
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{ maxwidth: 100%;
} form .user-details .inputbox{ 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{ flexdirection: 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{ marginleft: auto; marginright: auto; textalign: center;
}

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

#sidebar-wrapper .sidebar-heading {
padding: 0.875rem 1.25rem; fontsize: 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 { marginleft: 0;
}

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

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

## 7.1 OUTPUT SCREENSHOTS:

127.0.0.1:5501/template/loginpage.html

### Login

Username  
Enter your username

Password  
Enter your password

Login

Don't have an account? [Register here](#)

Activate Windows  
Go to Settings to activate Windows.

Search the web and Windows

1304 10-11-2022

127.0.0.1:5501/template/register.html

### Registration

Full Name  
Enter your name

Username  
Enter your username

Email  
Enter your email

Phone Number  
Enter your number

Password  
Enter your password

Confirm Password  
Confirm your password

Gender  
☐ Male ☐ Female ☐ Others

Register

Activate Windows  
Go to Settings to activate Windows.

Search the web and Windows

1304 10-11-2022

127.0.0.1:5501/template/addproduct.html

### Add Product

Product Name

Stock Quantity

Product Price / (kg)

Stock Alert min. Qty

Add Product

Activate Windows  
Go to Settings to activate Windows.

Search the web and Windows

1301 16-11-2022

127.0.0.1:5501/template/billing.html

### BILLING

Invoice No :

Bill Date:

Product Name :

Product Quantity:

Price:

Save Cancel

Activate Windows  
Go to Settings to activate Windows.

Search the web and Windows

1302 16-11-2022



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

Activate Windows  
Go to Settings to activate Windows.

Search the web and Windows

1102 16-11-2022

127.0.0.1:5501/template/searchproduct.html

search product?

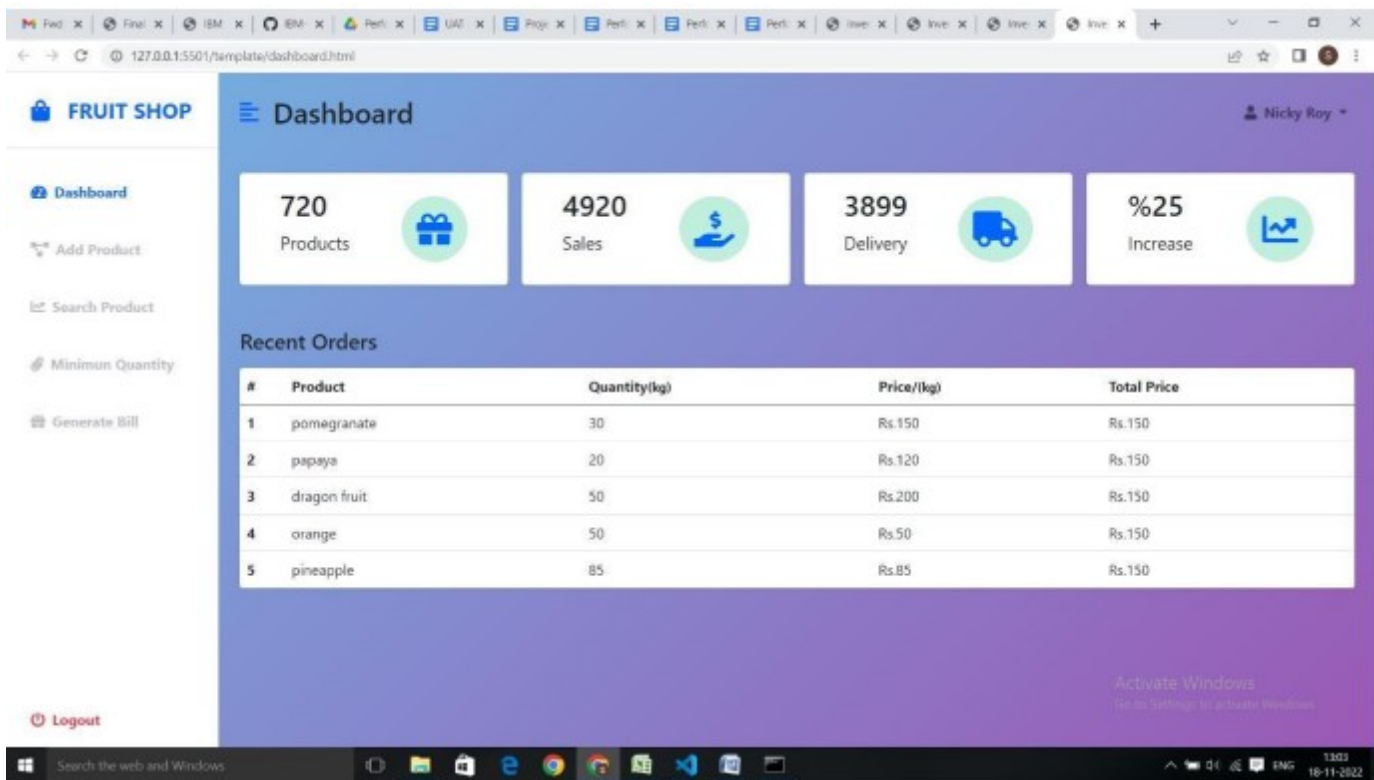
### Search Product

#	Product	Customer	Price
1	pomegranate	rocky	Rs.1500
2	papaya	rahul	Rs.1000
3	dragon fruit	selvi	Rs.2000
4	orange	babu	Rs.500
5	pineapple	roja	Rs.200

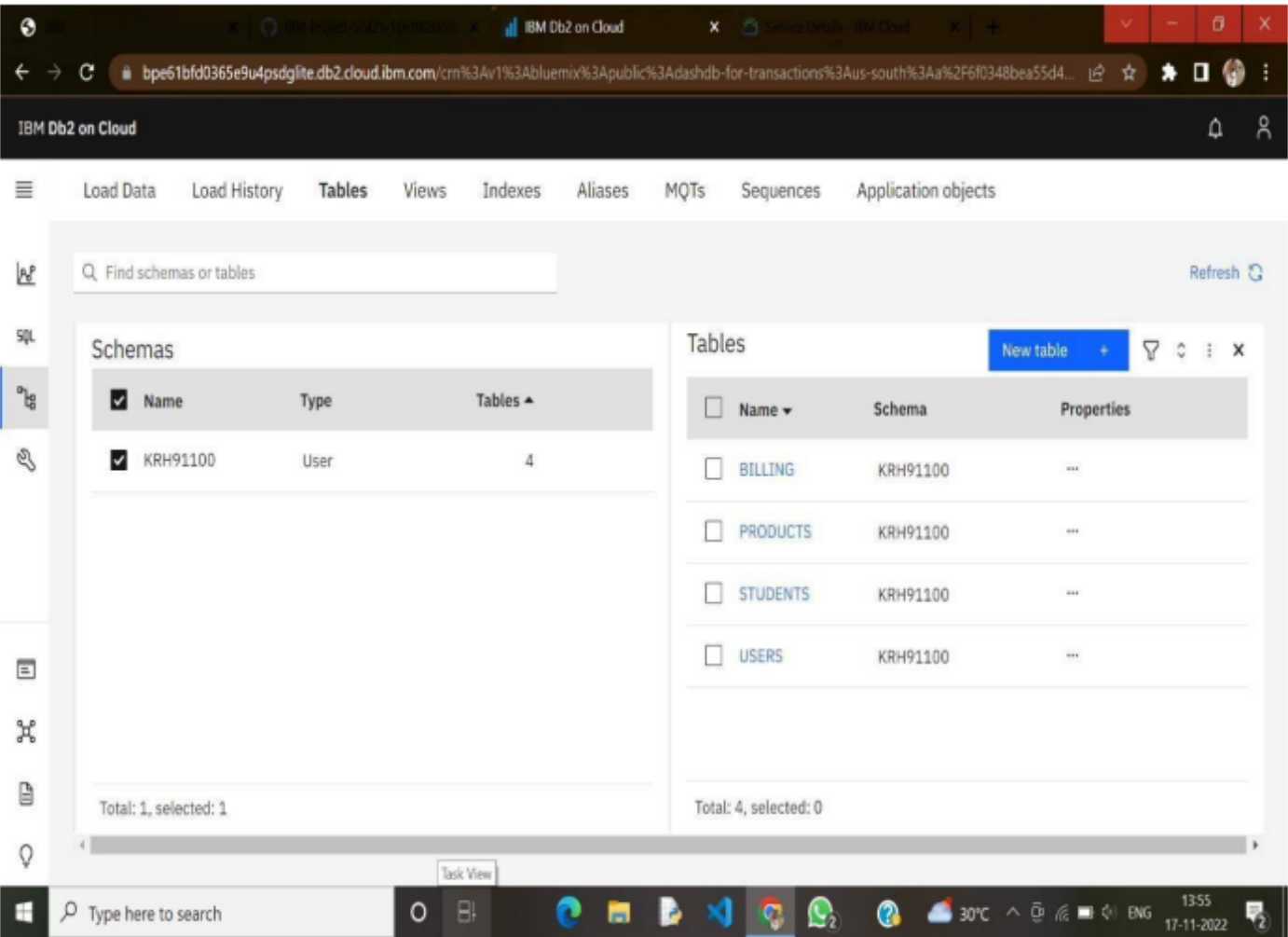
Activate Windows  
Go to Settings to activate Windows.

Search the web and Windows

1104 16-11-2022



7.2 Database schema:



## Source code:

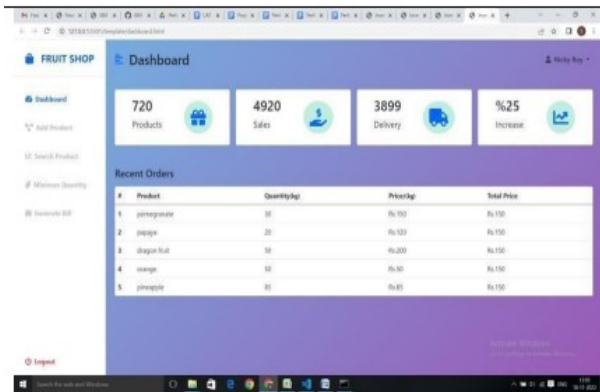
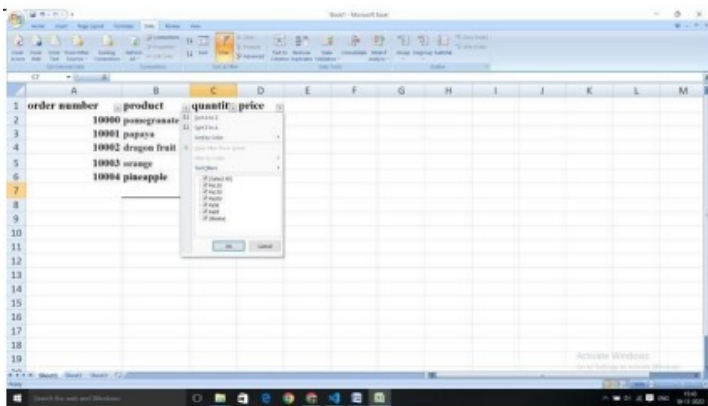
```
dsn_hostname = "9938aec0-8105-433e-8bf9-  
0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud" # e.g.:  
"54a2f15b-5c0f-46df-8954-  
7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"  
dsn_uid = "krh91100" # e.g. "abc12345" dsn_pwd =  
"gSPyVtDpdim5wKGL" # e.g. "7dBZ3wWt9XN6$0J"
```

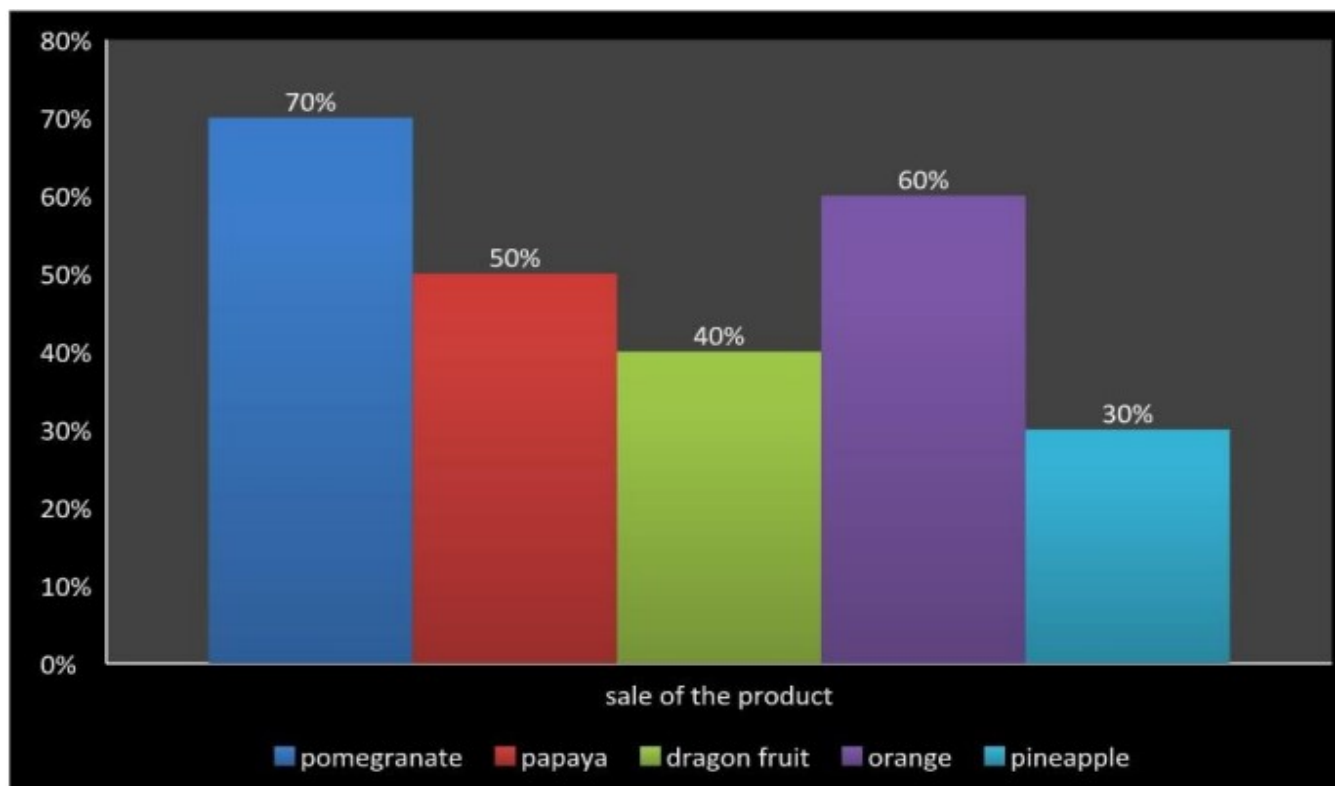
```
dsn_driver = "{IBM DB2 ODBC DRIVER}"  
dsn_database = "bludb" # e.g. "BLUDB" dsn_port  
= "32459" # e.g. "32733" dsn_protocol =  
"TCPIP" # i.e. "TCPIP" dsn_security = "SSL"  
#i.e. "SSL"
```

```
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()  
)
```

## 8.TESTING

S.NO	PARAMETER	SCREENSHOT / VALUES																														
	Dashboard design																															
	Data Responsiveness	<ul style="list-style-type: none"><li>• Responds quickly to unpredictable demand.</li><li>• Have a lower utilization rate.</li><li>• Utilise an excess buffer capacity.</li><li>• Invest in lead time reduction.</li></ul>																														
	Amount Data to Rendered (DB2 Metrics)	8																														
	Utilization of Data Filters																															
	Effective User Story	<ul style="list-style-type: none"><li>• Dashboard is used to accessing the other pages</li><li>• Add product page is to add the product they need.</li><li>• Minimum quantity page is to view the present quantity and minimum quantity of the product they need</li><li>• Search product page is used to search the product quickly.</li></ul>																														
	Descriptive Reports	<table><thead><tr><th>Item number</th><th>Description</th><th>Unit of stock</th><th>Unit price</th><th>Total price</th></tr></thead><tbody><tr><td>10001</td><td>pomegranate</td><td>20</td><td>Rs100</td><td>Rs2000</td></tr><tr><td>10002</td><td>papaya</td><td>0</td><td>Rs80</td><td>0</td></tr><tr><td>10003</td><td>dragon fruit</td><td>25</td><td>Rs200</td><td>Rs2500</td></tr><tr><td>10004</td><td>orange</td><td>40</td><td>Rs90</td><td>Rs3600</td></tr><tr><td>10005</td><td>pineapple</td><td>0</td><td>Rs70</td><td>0</td></tr></tbody></table>	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
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																												



### 8.1 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

[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 etc.	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

1 | Page

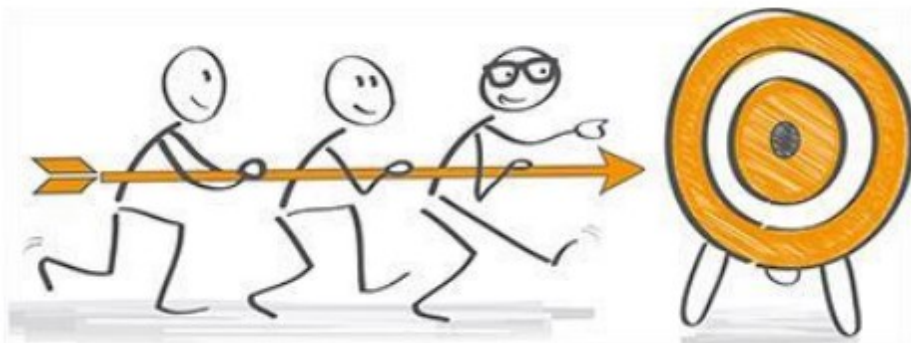
© [Name of Company]

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 :

```
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 LINK :** <https://github.com/IBM-EPBL/IBM-Project-42355-1660660478>

**DEMO-VIDEO LINK :** [https://drive.google.com/file/d/13oyvTTa5-Lg02dQvKZYJIG2ceIE\\_kPIL/view?usp=share\\_link](https://drive.google.com/file/d/13oyvTTa5-Lg02dQvKZYJIG2ceIE_kPIL/view?usp=share_link)