

INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

NALAIYA THIRAN PROJECT BASED LEARNING

ON

**PROFESSIONAL READINESS FOR INNOVATION,
EMPLOYABILITY AND ENTREPRENEURSHIP**

A PROJECT REPORT

VIKRAM R - 19106133

DHARUN V - 19106138

SATHISH S - 19106102

STEPHEN A - 19106116

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

HINDUSTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY

Approved by AICTE, New Delhi, Accredited with 'A' Grade by NAAC

(An Autonomous Institution, Affiliated to Anna University, Chennai)

COIMBATORE - 641 032

NOVEMBER 2022

Hindusthan College of Engineering And Technology

Approved by AICTE, New Delhi, Accredited with 'A' Grade by NAAC (**An Autonomous Institution, Affiliated to Anna University, Chennai**) Valley Campus, Pollachi Highway, Coimbatore – 641 032

INTERNAL MENTOR:

DR.Suresh Kumar P

Associate Professor

Department of Electronics and Communication Engineering

Hindusthan College of Engineering and Technology,

Coimbatore - 641 032

INDUSTRY MENTOR

MR.VASUDEVA HANUSH

IBM

1. INTRODUCTION

1. Project Overview
2. Purpose

2. LITERATURE SURVEY

1. Existing problem
2. References
3. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

4. REQUIREMENT ANALYSIS

1. Functional requirement

5. PROJECT DESIGN

1. Data Flow Diagrams
2. Solution & Technical Architecture
3. User Stories

6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule
3. Reports from JIRA

7. CODING & SOLUTIONING

1. Feature 1
2. Feature 2

3. Database Schema (if Applicable)

8. TESTING

1. Test Cases

2. User Acceptance Testing

9. RESULTS

1. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

INTRODUCTION

1. Project Overview

In recent years, the correct management of inventories has become a fundamental pillar for achieving success in enterprises. Unfortunately, studies suggesting the investment and adoption of advanced inventory management and control systems are not easy to find. In this context, this article aims to analyze and present an extensive literature concerning inventory management, containing multiple definitions and fundamental concepts for the retail sector. A systematic literature review was carried out to determine the main trends and indicators of inventory management in Small and Medium-sized Enterprises (SMEs).

This research covers five years, between 2015 and 2019, focusing specifically on the retail sector. The primary outcomes of this study are the leading inventory management systems and models, the Key Performance Indicators (KPIs) for their correct management, and the benefits and challenges for choosing or adopting an efficient inventory control and management system.

2. Purpose

The main purpose of inventory management is to ensure there is enough goods or materials to meet demand without creating overstock, or excess inventory. Retail inventory management is the process of **ensuring you carry merchandise that shoppers want, with neither too little nor too much on hand.**

LITERATURE SURVEY

1.Existing problem

Inconsistent Tracking:

Using manual inventory tracking procedures across different software and spreadsheets is time-consuming, redundant and vulnerable to errors. Even small businesses can benefit from a centralized inventory tracking system that includes accounting features.

Warehouse Efficiency:

[Inventory management controls](#) at the warehouse is labor-intensive and involves several steps, including receiving and putaway, picking, packing and shipping. The challenge is to perform all these tasks in the most efficient way possible.

2.References

1.Srinivasa Rao Kasisomayajula(2014) “An Analytical Study on Inventory Management in Commercial Vehicle Industry in India”, International Journal of Engineering Research, Vol.3, Iss.6, pp.378-383.

2.Abramovitz & Modigliani, Franco (1957), “Business Reasons for Holding Inventories and Their Macro Economic Implications”, Problems of Capital Formation, Studies in Income and Wealth, Vol. 19, NBER, pp. 495-511.

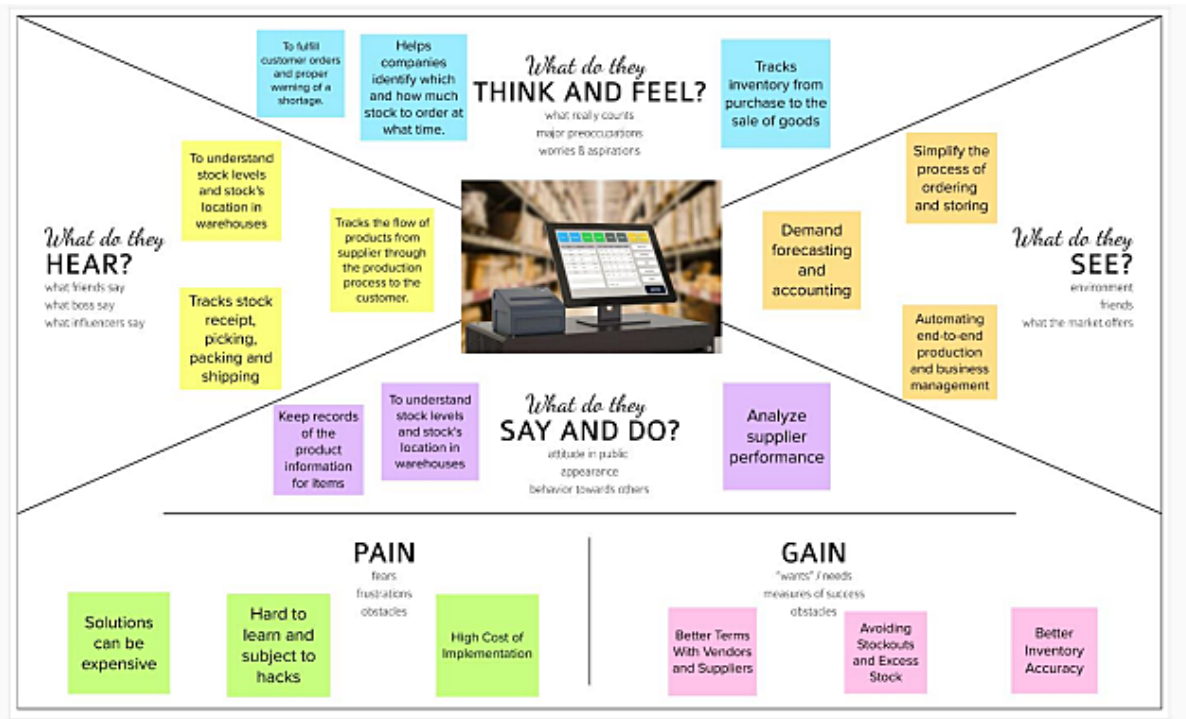
3.Problem Statement Definition

Problem Stateme nt (PS)	I am (Custome r)	I'm trying to	But	Because	Which makes me feel
ps-1	Retailer	Find number of stocks in my inventory	It's difficult	My inventory has n number of product difficult to count manually	Tired
ps-2	Retailers	Maintaini ng the invoice of inventory	It's difficult and unsecure	It may get lost or damage	Afraid
ps-3	Retailers	Access a particular stock	I can't find the stock	There are n number of stock in my	Time consuming

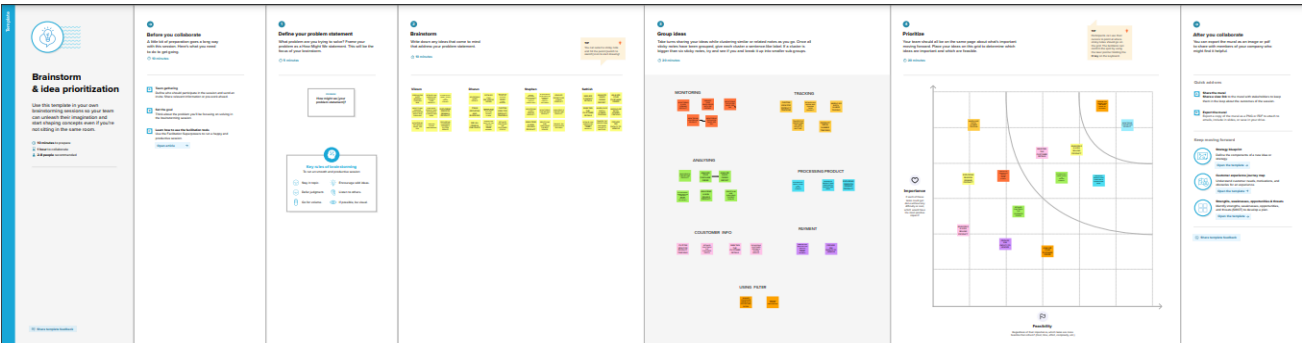
				inventory	
--	--	--	--	-----------	--

IDEATION & PROPOSED SOLUTION

1. Empathy Map



2. Ideation & Brainstorming



3.Proposed Solution

No	Parameter	Description
1.	Problem Statement	The retailers need a systematic software procedure to keep track of their inventory data because they only keep it in the logbook and not properly organized so that they are able to record the inventory data quickly and safely.
2.	Idea / Solution description	We are proposing a solution that focuses on tracking the inventory of the retailers and comparing them with the threshold value which was set by the user and preparing an analysis that results in the overall analysis of the sales.
3.	Novelty / Uniqueness	The project will be able to provide real time statistics about revenue, stocking and real-time tracking of shipped goods similar to amazon.
4.	Social Impact / Customer Satisfaction	The retailers need a way to maintain a systematic software procedure to keep track of their inventory data because they only keep it in the logbook and not properly organized so that they are able to record the inventory data quickly and safely. This will greatly aid the retailer on their vision for improving their business by providing future

		<p>statistics</p> <p>and analysis of day-to-day sales.</p>
5.	Business Model (RevenueModel)	<p>Providing the app as a service with having certain features as subscribable and available to premium users who will be a charged over</p> <p>a specific period of time depending upon the plan they choose to use.</p>
6.	Scalability of the Solution	<p>The project will be having the Python Flask for the development of the backend, which makes it easy to run on any web browsers and it will affect the collection of data from the user side and the sending of prediction analysis from the IBM Watson.</p> <p>The server IBM Cloud will be used for making it work without any slow loading or delay of the prediction of the website.</p>

3.Problem Solution Fit

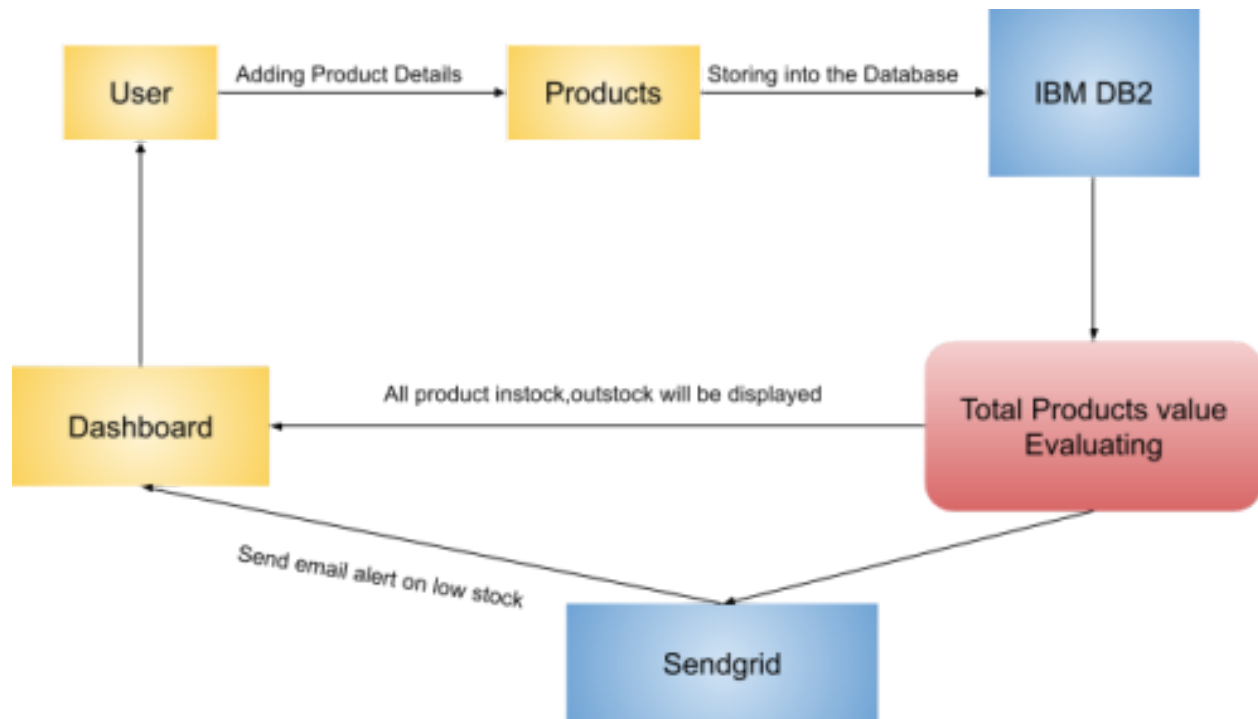
Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? i.e. working parents of 0-5 y.o. kids <ul style="list-style-type: none"> o General consumers who are in need of a product. o They can be of all ages. 	6. CUSTOMER CONSTRAINTS CC What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices. <ul style="list-style-type: none"> o Cost of product. o Quality product. o Lack of network connection. o Delivery cost. o Product delivery delay. o Device to order. 	5. AVAILABLE SOLUTIONS AS 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 <ul style="list-style-type: none"> o They can compare the cost of product and purchase their desired choice. o They can return if the quality does not satisfy their expectation. o They can see when the delivery date is and they can decide to purchase the product or not. 	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <ul style="list-style-type: none"> o Maintaining up-to-date products. o Ordering the right amount and not in excess. o Purchasing the products in lower price than selling price. o Having competitive stock prices. o Product demand forecasting. o Not having enough bandwidth to support 'n' number of consumers in the site at a time. 	9. PROBLEM ROOT CAUSE RC 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. <ul style="list-style-type: none"> o Can't predict customer's needs in short period of time. Need data to have an accurate stock prediction. o Contacting suppliers and getting good deals from them. o Having low bandwidth to hold sufficient consumers in the site. 	7. BEHAVIOUR BE 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) <ul style="list-style-type: none"> o Estimation of sales prediction to stock up by having customer feedback. o Finding good suppliers with low cost of product. o Customer feedback for improvement of application. o Having sufficient bandwidth to support on demand consumers. 	

REQUIREMENT ANALYSIS



PROJECT DESIGN

1.Data Flow Diagrams



2.Solution & Technical Architecture

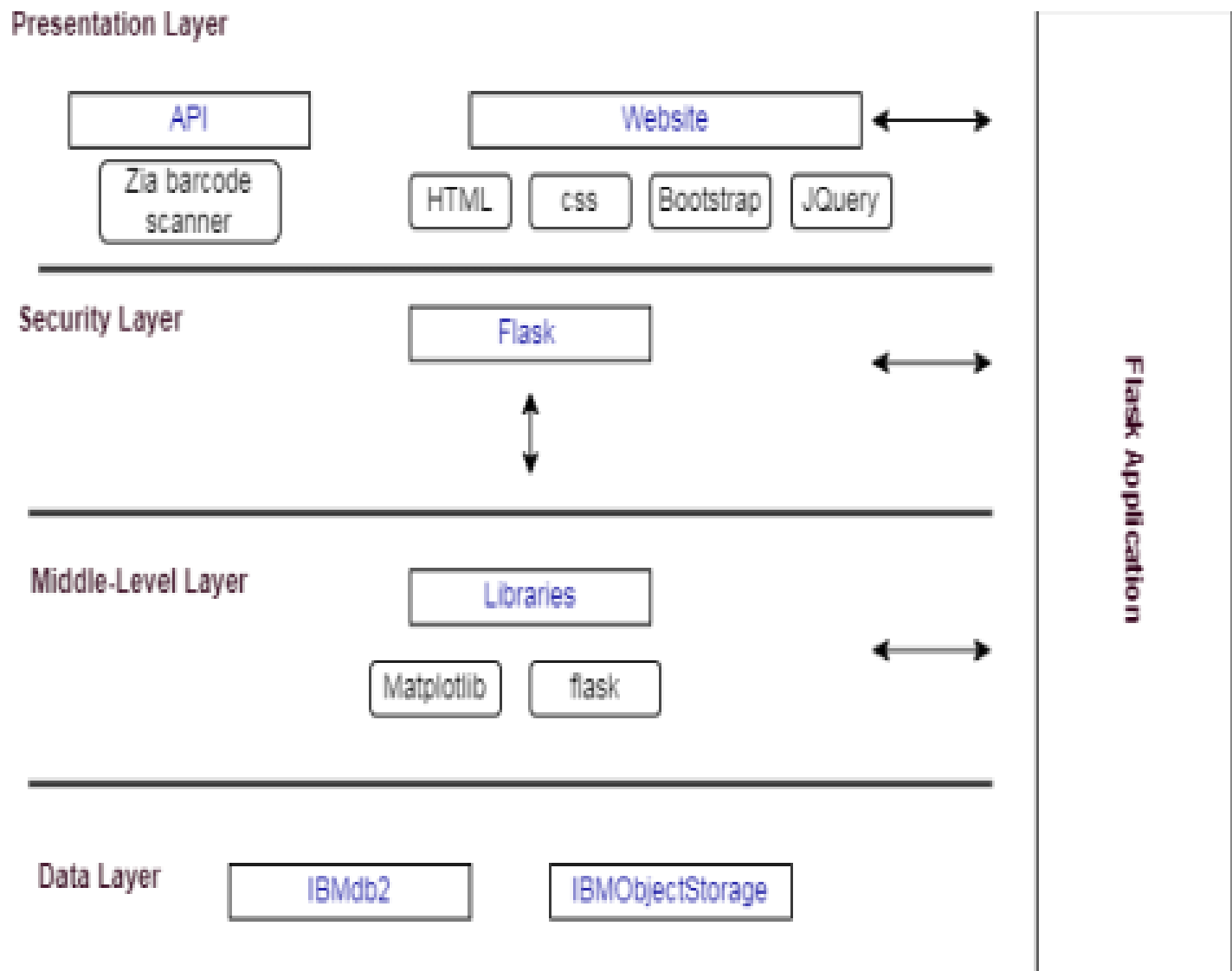


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web UI with Chatbot	HTML, CSS, Bootstrap, Jquery
2	Calculating Products Count	By entering barcode details into the	Zia Barcode Scanner

		application	
3	Showing high demand product	By the products data in IBMdb2	Data Visualization using Python Bar plot by Matplot Library
4	Alert and Notification	Alerting the retailers regarding the low stock count of the product	SendGrid
5	Chat	Chat with watson assistant	IBM Watson Assistant
6	Cloud Database	Database Service on Cloud	IBM DB2
7	File Storage	File storage requirements	IBM Object Storage
8	External API-1 Barcode	To Scan the product barcode	Zia Barcode Scanner
9	Infrastructure (Server / Cloud)	Cloud Server Configuration	Cloud Foundry, Kubernetes

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Styling our page,Python flask microframework	Python Flask, Bootstrap
2.	Security Implementations	For securing our cloud data	SSL Certificates
3.	Scalable Architecture	Three – tier architecture (MVC)	Web server - HTML, CSS, Javascript Application server - Python Flask, Docker, Container Registry Database server - IBM DB2
4.	Availability	availability of application	IBM Load Balancer
5.	Performance	5 requests per seconds, Use of Local Machine Cache Memory	IBM Cloud, CDN

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
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	As a user,I can update stock in & out count details	Updation can be made through barcode scanning	High	Sprint -2
	Dashboard	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

PROJECT PLANNING & SCHEDULING

1.Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story User Story / Task Number	Story Points	Priority	Team Members
Sprint-1	Registration	As a user, I can register for the application by using my email & password and confirming my login credentials.	3	High	Vikram R Dharun V Stephen A Sathish S
Sprint-1		USN-2 As a user, I can login through my E-mail.	3	Medium	Vikram R Dharun V Stephen A Sathish S

Sprint-2	Dashboard	USN-5 As a user, I can view the products that are available currently.	4	High	Vikram R Dharun V Stephen A Sathish S
----------	-----------	--	---	------	--

Sprint-2	Stocks update	ser, I can add products which are not available in the inventory and restock the products.	3	Medium	Vikram R Dharun V Stephen A Sathish S
Sprint-3	Sales prediction	ser, I can get access to sales prediction tool which can help me to predict better restock management of product.	6	Medium	Vikram R Dharun V Stephen A Sathish S
Sprint-4	Request for customer care	ser, I am able to request customer care to get in touch with the administrators and enquire the doubts and problems.	4	Medium	Vikram R Dharun V Stephen A Sathish S

2. Sprint Delivery Schedule

Sprint	Total Story Duration Sprint Start Date Sprint End Points Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	11 6 Days 24 Oct 2022 29 Oct	11	29 Oct 2022
Sprint-2	2022 7 6 Days 31 Oct 2022	7	05 Nov 2022

	05 Nov 2022		
Sprint-3	6 6 Days 07 Nov 2022 12 Nov 2022	6	12 Nov 2022
Sprint-4	7 6 Days 14 Nov 2022 19 Nov 2022	7	19 Nov 2022

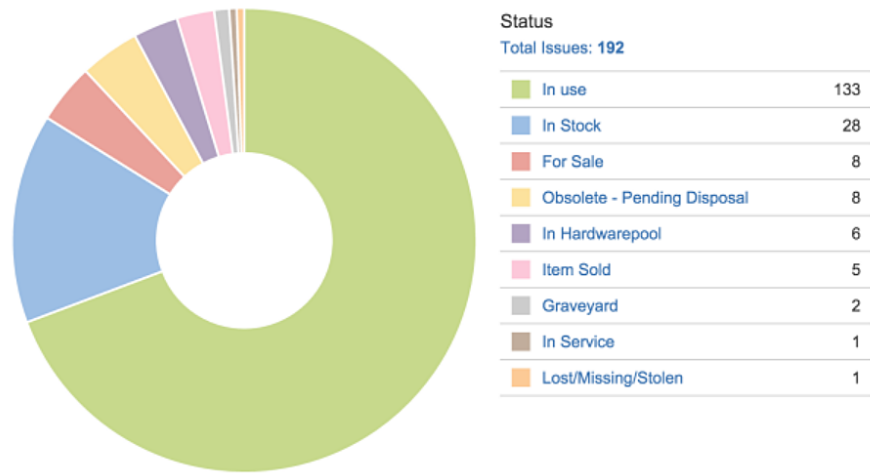
3.Reports from JIRA

Two Dimensional Filter Statistics: All Active Inventory Assets					
Status	Computer Asset	Misc. Hardware Asset	Monitor/TV Asset	Phone/Tablet Asset	T:
ENTER ITEM DETAILS	0	1	0	1	2
IN USE	133	216	105	27	481
IN STOCK	28	43	11	8	90
IN SERVICE	1	0	0	0	1
OBSOLETE - PENDING...	8	0	2	0	10
FOR SALE	8	0	0	0	8
GRAVEYARD	2	10	0	0	12
ITEM SOLD	5	3	0	0	8
LOST/MISSING/STOLEN	1	4	0	0	5
IN HARDWAREPOOL	6	20	20	81	127
Total Unique Issues:	192	297	138	117	744

Grouped by: Issue Type

Showing 10 of 10 statistics.

Pie Chart: All Computer Assets



CODING & SOLUTION

1.Feature 1

Register

Service Details - IBM Cloud


IBM Db2 on Cloud

localhost:5000/register

WhatsAppAAA COLLEGE OF E...IBM CloudIBM-EPBL/Assignm...kshyam/ibm-trainin...Docker HubReactJS Tutorial - 2...IMSFR board - Agil...Free 3d document...Successful purchas...

IMS

HomeRegisterLoginDropdownLogoutList



Register

Submit

Already have an Account? **Login**

2FEATURE 2


Report email - 2022-11-07 - dai...

Login

localhost:5000/login

IMS

HomeRegisterLoginDropdownLogoutList



Login

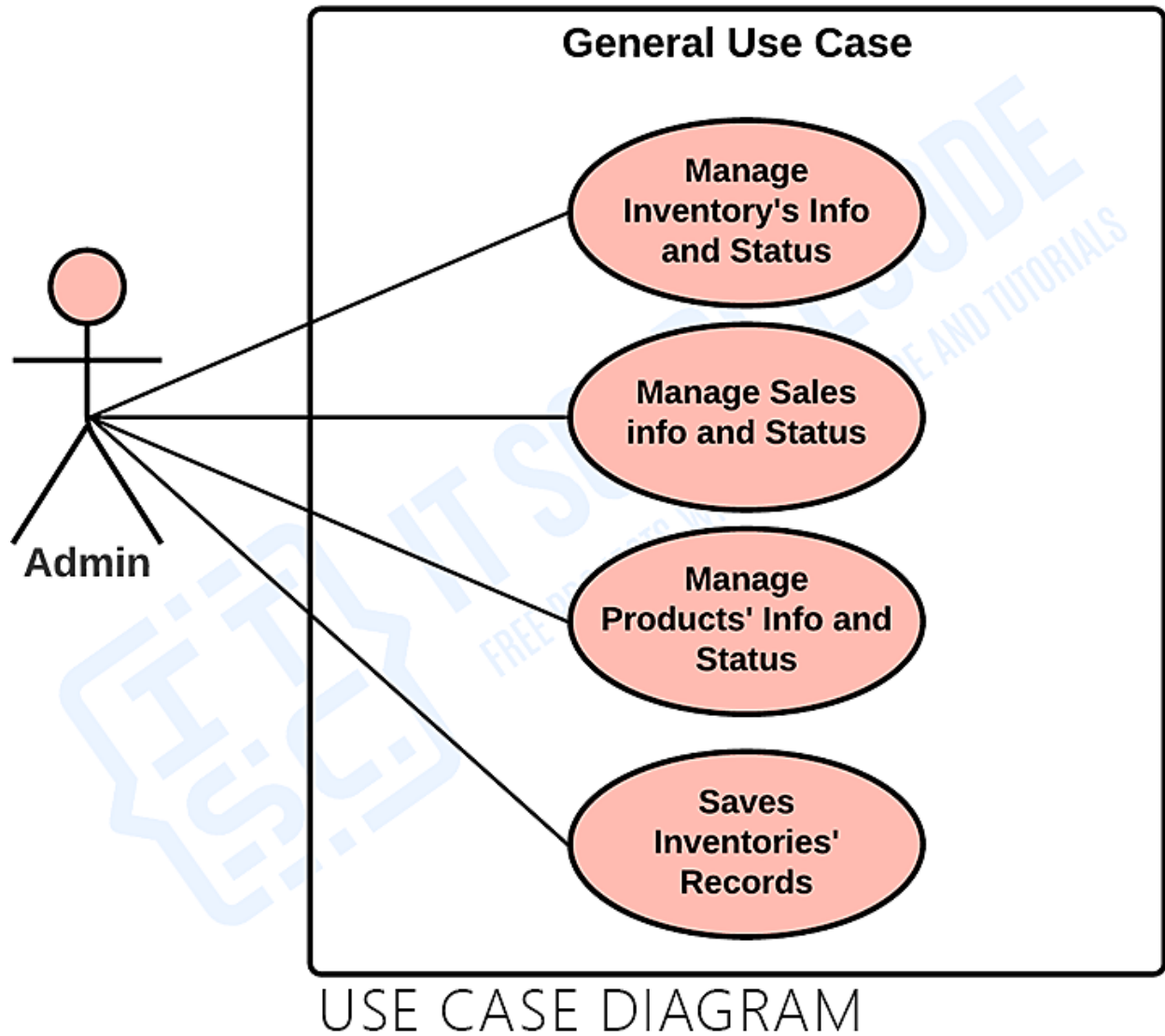
Submit

Don't have an Account? **Register**

TESTING

1.Test Cases

INVENTORY MANAGEMENT SYSTEM



2. User Acceptance Testing

Test case : Testing the Add Recipe Interface and its functioning

Case 1: Testing the Quantity input field.

Case 2: Testing the Recipe Name field.

Case 3: Testing the Ingredients in recipe list and Quantity of ingredient list.

Case 4: Testing the available ingredients list.

Case 5: Testing the all the above cases together and checking if the entries are updated to the tables in database.

Test Case : Check Threshold Interface

Case 1: Check if the Ingredients under the threshold values are shown in the Ingredients below threshold list.

Case 2: Check if the Create order button asks the user to enter values for all the ingredients listed under the ingredients below threshold list.

Case 3: Check if pressing the Process Order button creates a file with the order details in it.

Test Case : Testing the Update after sales interface

Case 1: Test the Recipe list box.

Case 2: Test the quantity text field..

Case 3: Test the recipe sold list box quantity sold list box.

Case 4: Test if the details are updated to the database when requested.

RESULTS

The screenshot shows the Jira Software interface for a project named 'Inventory Management'. The left sidebar contains a 'Reports' section with options like 'Overview', 'Burnup report', 'Sprint burndown chart' (selected), 'Velocity report', 'Cumulative flow diagram', 'Cycle time report', and 'Deployment frequency report'. The main content area displays 'Completed issues' for the current sprint. Below this, there is a section for 'Issues completed outside of sprint'.

Key	Summary	Issue type	Epic	Status	Assignee	Story points
IMSFR-2	As a user, I can register for the application by entering my email...	Story	REGISTRATION	DONE	P	2
IMSFR-3	As a user, I will receive a confirmation email once I have register...	Story	REGISTRATION	DONE	RM	1
IMSFR-5	As a user, I will confirm the registration once I have received the ...	Story	CONFIRMATION	DONE	S	2
IMSFR-7	As a user, I can log in to the application through Gmail & Passw...	Story	LOGIN	DONE	AA	2

The screenshot shows the 'Register' page of the IMS application. The page has a header with the 'Register' title and navigation links: Home, Register, Login, Dropdown, Logout, and List. The main content area is titled 'Register' and contains a form with several input fields for user registration.

Register

Padmanaban P

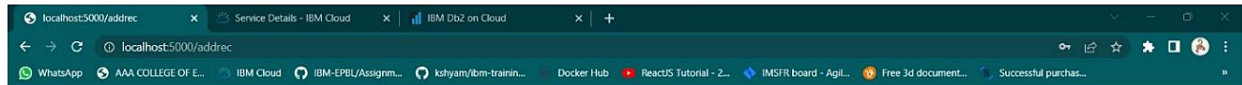
IBM Tamil Nadu

Virudhunagar 626001

8637683771 darksoul10a@gmail.com

Submit

Already have an Account? [Login](#)



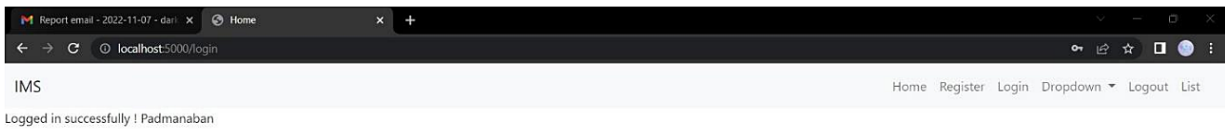
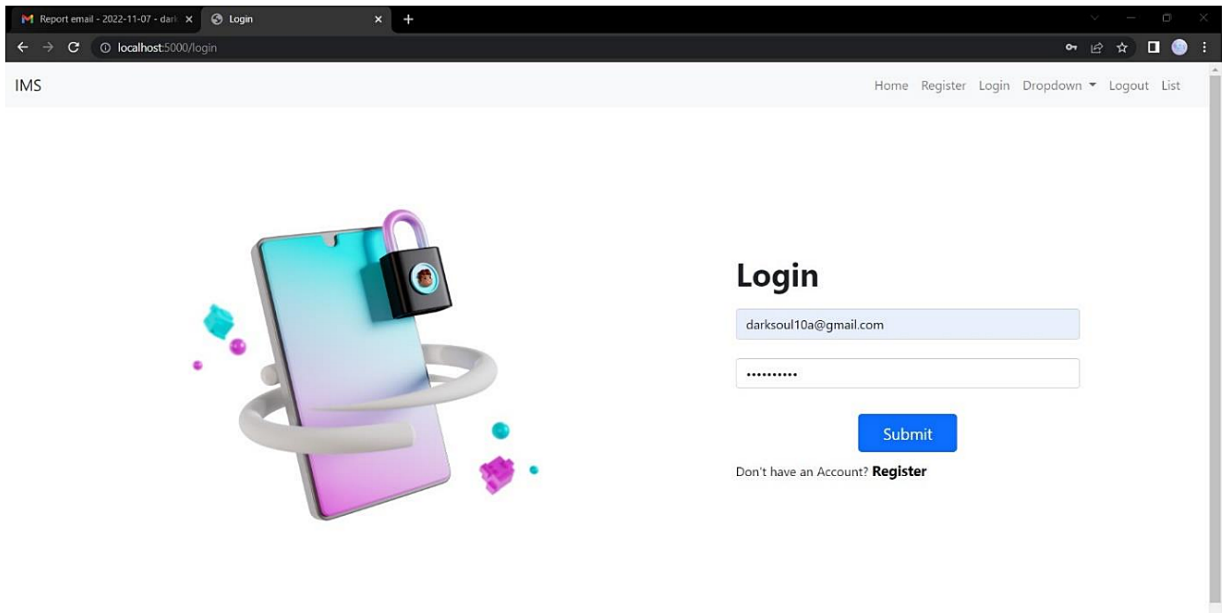
We have sent a confirmation mail to your registered E-mail.
Please confirm the mail to continue Registration.



Your Registration was successful!
Now, Log In to your account

Login





The screenshot shows the IBM Db2 on Cloud interface. The table 'LKC93724.USERS' is displayed with the following data:

ID	FIRSTNAME	LASTNAME	COMPANYNAME	STATE	CITY	MOBILENO	EMAILID	PASSWORD	PINCODE
1	Padmanaban	P	IBM	Tamil Nadu	Virudhunagar	8637683771	darksoul10a@gmail.com	darksoul10	626001

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.

Many

Lean practitioners claim that inventory performance is the single best indicator

of the overall operational performance of a facility. Inventory performance

looks at and is measured using either Inventory Days OnHand (DOH) or Inventory Turns.

- Inventory Days On-Hand: The number of days it would take to

consume

current on-hand inventory. Always measure multiple inventory item numbers in terms of currency (i.e. COGS).

- Inventory Turns: The number of times inventory is replaced in a year.

ADVANTAGES & DISADVANTAGES

ADVANTAGES :

Automated Reordering and In-Stock Information

Computerized inventory informs employees and customers within seconds whether an item is in stock. Because the inventory is synced with sales, there is a **running tally of what is in stock and what isn't**. This helps flag reordering needs and provides better service to customers. As inventory drops below a specific threshold, new orders are placed with vendors and tracked to let customers know when the new products will arrive.

Integration With Accounting

Many of the computerized inventory platforms **integrate with accounting software to track cash flow**. This makes the process of transferring inventory costs and assets between programs seamless and reduces the need for additional bookkeeping costs. Financial statements are more easily generated with shared data between inventory and bookkeeping.

DISADVANTAGES:

System Crash

One of the biggest problems with any computerized system is the **potential for a system crash**. A corrupt hard drive, power outages and other technical issues can result in the loss of needed data. At the least, businesses are interrupted when they are unable to access data they need. Business owners should back up data regularly to protect against data loss.

Malicious Hacks

Hackers look for any way to get company or consumer information. An inventory system connected to point-of-sale devices and accounting is a **valuable resource to hack into** in search of potential financial information or personal details of owners, vendors or clients. Updating firewalls and anti-virus software can mitigate this potential issue.

CONCLUSIONS

The project “Inventory Management System for Retailers” mainly as the name suggests deals with the calculation of the available and processed resources for an accurate inventory control and process management for a domain specific client who are related to the subject of food chains/outlets. This enables the inventory to be applied at every level in the hierarchy of the products and its complex combinations of recipes. A system that accurately calculates the atomic ingredients used for making a recipe then automatically performs the back end operation pertaining to a database of many relational tables onto which the changes are being made with each and every operation performed on the front end and which also shows up if at the time of retrieval. The most important part of Inventory controlling is its ability to check for threshold levels and alert the manager to replenish the stock before it reaches a danger zone. So

as when an ingredient level goes below the threshold level then it routes an alert to the manager. Then if needed accordingly an automated order form is produced so as to each specific vendor along with the quantities needed for replenishment. As a part of the standard maintaining a drill of risk management is done in order to sustain during the days of special occasion or holidays when the demand reaches to rather more different scale as compared to other days. These occasions call on for special inclusions into the menu which reflects on the recipes and in turn reflects the ingredients being used up eventually. Thus was provided the liberty of adding special recipe to the menu for some special occasion and is regarded as a key feature.

FUTURE SCOPES

1. Stock control for omnichannel retailing
2. Inventories that power experiential retail
3. Streaming Analytics
4. Advanced sales forecasting
5. Season-based product recommendations

APPENDIX

App.py

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

```
app = Flask(name)
```

```
app.secret_key = 'a'
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-
```

```
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=wjy24066;PWD=3w6H3sui635KMvWX",",")
```

```
print(conn)
```

```
print("Connecting Successful!!!!!!!!!!")
```

```
@app.route('/')
```

```
def homer()
```

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

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

```
def login():
```

```
global userid
```

```
msg = "
```

```
if request.method == 'POST' :
```

```
username = request.form['username']
```

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

```
sql = "SELECT * FROM users WHERE username =? AND password=?"
```

```
stmt = ibm_db.prepare(conn, sql)
```

```
ibm_db.bind_param(stmt,1,username)
```

```

ibm_db.bind_param(stmt,2,password)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print (account)
if account:
    session['loggedin'] = True
    session['id'] = account['USERNAME']
    userid= account['USERNAME']
    session['username'] = account['USERNAME']
    msg = 'Logged in successfully !'
    return render_template('dashboard.html', msg = msg)
else:
    msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

```

```

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

```

```

        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
            elif not re.match(r'^@]+@^[^@]+\.[^@]+', email):

```

```

msg = 'Invalid email address !'
elif not re.match(r'[A-Za-z0-9]+', username):
msg = 'name must contain only characters and numbers !'
else:
insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prepare_stmt, 1, username)
ibm_db.bind_param(prepare_stmt, 2, email)
ibm_db.bind_param(prepare_stmt, 3, password)
ibm_db.execute(prepare_stmt)
msg = 'Please fill out the form !'
if request.method == 'POST':
msg = 'You have successfully registered! Please login !'
return render_template('register.html', msg = msg)

```

```

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

```

```

def add_stock():
msg=""
if request.method == "POST":
prodname=request.form['prodname']
quantity=request.form['quantity']
warehouse_location=request.form['warehouse_location']
sql='SELECT * FROM product WHERE prodname =?'
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,prodname)
ibm_db.execute(stmt)
acnt=ibm_db.fetch_assoc(stmt)
print(acnt)

```

```

if acnt:

```



```
msg='Product already exists!!'
else:
insert_sql='INSERT INTO product VALUES (?, ?, ?)'
pstmt=ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(pstmt,1,prodname)
ibm_db.bind_param(pstmt,2,quantity)
ibm_db.bind_param(pstmt,3,warehouse_location)
ibm_db.execute(pstmt)
msg='You have successfully added the products!!'
return render_template("dashboard.html")
```

```
else:
msg="fill out the form first!"
return render_template('add_stock.html',meg=msg)
```

```
@app.route('/delete_stock',methods=['GET','POST'])
def delete_stock():
if(request.method=="POST"):
prodname=request.form['prodname']
sql2="DELETE FROM product WHERE prodname=?"
stmt2 = ibm_db.prepare(conn, sql2)
ibm_db.bind_param(stmt2,1,prodname)
ibm_db.execute(stmt2)
```

```
flash("Product Deleted", "success")
```

```
return render_template("dashboard.html")
```

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

```

mg=""
if request.method == "POST":
    prodname=request.form['prodname']
    quantity=request.form['quantity']
    quantity=int(quantity)
    print(quantity)
    print(type(quantity))
    warehouse_location=request.form['warehouse_location']
    sql='SELECT * FROM product WHERE prodname ='
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,prodname)
    ibm_db.execute(stmt)
    acnt=ibm_db.fetch_assoc(stmt)

    print(acnt)

    if acnt:
        insert_sql='UPDATE product SET quantity=?,warehouse_location=?
        WHERE prodname=? '
        pstmt=ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(pstmt,1,quantity)
        ibm_db.bind_param(pstmt,2,warehouse_location)
        ibm_db.bind_param(pstmt,3,prodname)
        ibm_db.execute(pstmt)
        mg='You have successfully updated the products!!'
        limit=5 print(type(limit))
        if(quantity<=limit):
            ("Please update the quantity of the product {}, Atleast {} number of
            pieces must be added!".format(prodname,10))
            return render_template("dashboard.html",meg=mg)

    else:

```

```
mg='Product not found!!'
```

```
else:
```

```
msg="fill out the form first!"
```

```
return render_template('update_stock.html',meg=msg)
```

```
@app.route('/view_stock')
```

```
def view_stock():
```

```
sql = "SELECT * FROM product"
```

```
stmt = ibm_db.prepare(conn, sql)
```

```
result=ibm_db.execute(stmt)
```

```
print(result)
```

```
products=[]
```

```
row = ibm_db.fetch_assoc(stmt)
```

```
print(row)
```

```
while(row):
```

```
products.append(row)
```

```
row = ibm_db.fetch_assoc(stmt)
```

```
print(row)
```

```
products=tuple(products)
```

```
print(products)
```

```
if result>0:
```

```
return render_template('view.html', products = products)
```

```
else:
```

```
msg='No products found'
```

```
return render_template('view.html', msg=msg)
```

```
@app.route('/delete')
def delete():
    return render_template('delete_stock.html')
```

```
@app.route('/update')
def update():
```

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

```
@app.route('/logout')
```

```
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return render_template('home.html')
```

```
if name == 'main':
    app.run(host='0.0.0.0')
```

home.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
```

```
<style>
body {
font-family: "Lato", sans-serif;

}
```

```
/* Fixed sidenav, full height */
.sidenav {
height: 100%;
width: 300px;
position: fixed;
z-index: 1;
top: 0;
left: 0;
background-color: #0059b3;
overflow-x: hidden;
padding-top: 20px;
}
```

```
/* Style the sidenav links and the dropdown button */
.sidenav a{
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 20px;
color: rgb(239, 239, 239);
display: block;
border: none;
background: none;
width: 100%;
text-align: left;
```

```
cursor: pointer;
outline: none;
}
```

```
/* On mouse-over *
```

```
.sidenav a:hover{
color: #111;
}
```

```
/* Some media queries for responsiveness */
```

```
@media screen and (max-height: 450px) {
.sidenav {padding-top: 15px;}
.sidenav a {font-size: 18px;}
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
> Deeps:
```

```
<div class="sidenav">
```

```
<a href="{ {url_for('add_stock') } }"><strong>Add stock<strong></a>
```

```
<a href="{ {url_for('update') } }"><strong>Update stock details<strong></a>
```

```
<a href="{ {url_for('view_stock') } }"><strong>View stock<strong></a>
```

```
<a href="{ {url_for('delete') } }"><strong>Delete stock<strong></a>
```

```
<a href="{ {url_for('logout') } }"><strong>Log out<strong></a>
```

```
</div>
```

```
<nav>
```

```
<script>
```

```
window.watsonAssistantChatOptions = {
```

```
integrationID: "4bd6f313-33d4-4e87-8825-22b90b8e3c2c", // The ID of
```

this integration.

region: "au-syd", // The region your integration is hosted in.

serviceInstanceID: "60e1396a-421f-4091-b39a-a23a546843e8", // The ID
of your service instance.

```
onLoad: function(instance) { instance.render(); }
```

```
};
```

```
setTimeout(function(){
```

```
const t=document.createElement('script');
```

```
t.src="https://web-
```

```
chat.global.assistant.watson.appdomain.cloud/versions/" +
```

```
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
```

```
"/WatsonAssistantChatEntry.js";
```

```
document.head.appendChild(
```

```
});
```

```
</script>
```

```
</nav>
```

```
</body>
```

```
</html>
```

GITHUB:

<https://github.com/IBM-EPBL/IBM-Project-33968-1660230030>

DEMO LINK:

<https://drive.google.com/file/d/1oZxu4JYWUL972YmtMYHvFXRIHKpv7Jst/view?usp=drivesdk>