Inventory Management System for Retailers in Cloud Application Development REPORT

TEAM ID	**PNT2022TMID14132**	
TEAM MEMBER'S	**TEAM LEAD**	**SUTHARSAN.S**
	TEAM MEMBER	**SIDDARTHAN.S**
	TEAM MEMBER	**STANLY.C**
	TEAM MEMBER	**SUJITH.M**

CONTENT:

1.INTRODUCTION

➢ Project Overview :

Inventory management is one of the pillars of a successful retail operation. Retail inventory management techniques help stores and ecommerce sellers satisfy customers, reduce costs and increase profitRetail inventory management is the process of 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.

In practice, effective retail inventory management results in lower costs and a better understanding of sales patterns. Retail inventory management tools and methods give retailers more information with which to run their businesses, including:

- Product locations
- Quantities of each product type
- Which stock sells well and which doesn't, by location and sales channel.
- Profit margin by style, model, product line or item
- Ideal amount of inventory to have in back stock and storage
- How many products to reorder and how often
- When to discontinue a product
- How changing seasons affect sale

➢ Purpose:

- **Minimises Out-of-Stocks:**
  To avoid disappointing customers and missing sales, retailers want to avoid running out of inventory. Retailers can use inventory management tools to determine how much stock is "just right" to have on hand, neither too much nor too little. This amount will be larger for bestsellers than for unpopular products. Also, with real-time information on sales and stock, retailers can react quickly by reordering, transferring stock from another location or drop shipping to the customer.

- **Improves Profit Margins:**
  With lower inventory costs and enough supply to fill every order, retailers improve profitability.

- **Prevents Spoilage and Obsolescence:**
  Inventory management helps retailers address another costly inefficiency that happens when products expire or become obsolete. This phenomenon can apply to perishables that have a limited shelf life, such as milk and meat, or a non-perishable that becomes obsolete because

consumer tastes and technology change. For example, season collections or holiday-specific packaging. Or when a piece of consumer technology adds a popular new feature, the old models may face plummeting demand: Consider how the rise of smart televisions sunk demand for models that weren’t capable of streaming content.

- **Improves Multi-Channel and Omnichannel Performance and Order Fulfilment:**
  If you are selling via physical stores, your website and third-party merchants, it can be difficult to keep correct inventory counts across all channels. Having accurate inventory data across selling channels lets you use your inventory more efficiently, ultimately getting the product to consumers faster.

- **Simplifies Processes and Facilitates Growth:**
  Strong inventory management also reduces friction in your systems as sales grow. Shipping, receiving and order fulfilment run more smoothly, and you minimise errors, customer complaints and staff stress.

- **Reduces Shrinkage:**
  Shrinkage is inventory loss due to shoplifting, product damage, vendor mistakes or fraud, employee theft and administrative errors. According to a survey by the FMI food industry association, the average supermarket loses up to 3% of sales through shrinkage. A National Retail Federation survey puts average shrinkage for its members at 1.4% of sales in 2019. This data suggests that most losses stem from incorrectly recording inventory on intake, miscounting it or misplacing it. Stronger retail inventory management could reduce shrinkage by at least half.

- **Eases Supply Chain Management:**
  Having a firm grip on inventory and sales trends helps you manage your supply chain better. You can use the replenishment system that works best for you, whether that’s just-in-time ordering or fewer, bigger orders. Retail inventory management helps you determine your economic order quantity (EOQ), which is the ideal order size to minimise inventory costs including holding, shortage and ordering expenses. The EOQ formula, which factors in demand in units, ordering costs such as shipping charges and holding costs, works best when these variables remain consistent over time. Learn more about the EOQ formula.

- **Improves Customers Satisfaction:**
  When customers get the products they want faster with fewer mistakes or out-of-stocks, it increases customer loyalty.

- **Improves Forecasting:**
  You can use data such as historical sales results and available inventory to project future sales, growth and capital needs. These forecasts are vital to your budgeting and guide spending for marketing, product development and staffing

2 .LITERATURE SURVEY

Literature Survey: Inventory Management System for Retailers

Inventory Management System for Retailers is a management of process of distributing the products to the Shop owners from retailers. This management reduce the working cost & functional cost and also helps to know the demand over the customers.

1.Existing Solution:

• iVentRetail

Reference Link: https://ivend.com/retail-inventory-management-software/

• Zoho Inventory

Reference Link: https://www.zoho.com/in/inventory/

• Inventory management for retail companies: A literature review and

current trends

Reference Link:

https://www.researchgate.net/publication/352235223_Inventory_management_for

_retail_companies_A_literature_review_and_current_trends

• Development of inventory management system

Reference Link: https://ieeexplore.ieee.org/document/5478077

Inference from the existing solution:

The requirement for an indication, a tool, and a methodology to determine the business's trend and control inventories.

The decision for inventory management is based on

- Data from different key performance metrics, which take into account several aspects of the inventory influencing the business and

▪ Methodology (such as AUD and MDP) to forecast revenue and discount on

the products.

▪ Tools (such as RFID and barcodes) to maintain correct records across digital

and physical databases.

Drawbacks of Inventory management:

- Poor tracking due to the customer behaviour.
- Changing demand and Overselling.
- Supply chain and logistics complexity including warehouse

management.

- Insufficient order management.
- Forecasting inventory seasonal buys due to depreciation, market

downs, product damage and theft.

- Stock predictions.
- Chat bot or any other AI tools for communication will be lag.
- No individual email tracking.
- Completely in manual input.

[1] Automated Reordering and In-Stock Information, And System Crash

ADVANTAGE: 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.

DISADVANTAGE: 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 the data they need. Business owners should back up data regularly to protect against data loss.

[2] Integration With Accounting and Malicious Hacks

ADVANTAGE: 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.

DISADVANTAGE: Hackers look for any way to get a 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

[3] Forecasting and Planning and Reduced Physical Audits

ADVANTAGE: Inventory management software does more than track where inventory is located and when to reorder it. A data collection system is used to create needed forecasting and strategic planning reports. Business owners review trends regarding which products do well in certain months or during specific cyclical seasons. Business owners use this data to plan for growth and order inventory intelligently to best utilize cash flow resources.

DISADVANTAGE: When everything is automated, it is easy to forego time-consuming physical inventory audits. They may no longer seem necessary when the computers are doing their work. However, it is important to continue to do regular audits to identify losses such as spoilage or breakage. Audits also help business owners identify potential internal theft and manipulation of the computerized inventory system.

[4] Simplicity and Technical Issues

ADVANTAGE: Most business owners are not accountants or bookkeepers by trade and find it challenging to do most accounting tasks. This is where accounting software programs give a business owner advantages. A wide variety of accounting software programs are consumer friendly. Business owners can shop around to find a program that is easy to install, learn and use. Many programs provide prompts for the type of data that should be entered in each section. Once the system is established with bank accounts, debts, and vendors, the business owner only needs to update information

DISADVANTAGE: When dealing with computers, issues can arise. You may be completing year-end data for your accountant and experiencing a power outage. Computers might acquire a virus and fail. There is also the potential of users incorrectly performing software tasks that they are not familiar with. If a user tries to do one thing but inadvertently does something else, it might take some work to undo the error

[5] Reliability, Cost-Effectiveness, and Incorrect Information

ADVANTAGE: Most business owners are not accountants or bookkeepers by trade and find it challenging to do most accounting tasks. This is where accounting software programs give a business owner advantages. A wide variety of accounting software programs are consumer friendly. Business owners can shop around to find a program that is easy to install, learn and use. Many programs provide prompts for the type of data that should be entered in each section. Once the system is established with bank accounts, debts, and vendors, the business owner only needs to update information as it comes Most of the major software programs make using the program simple. The math is accurate and reliable, so a business owner can accurately determine available funds at any time

DISADVANTAGE: Business owners can do a lot to mitigate the disadvantages and potential problems

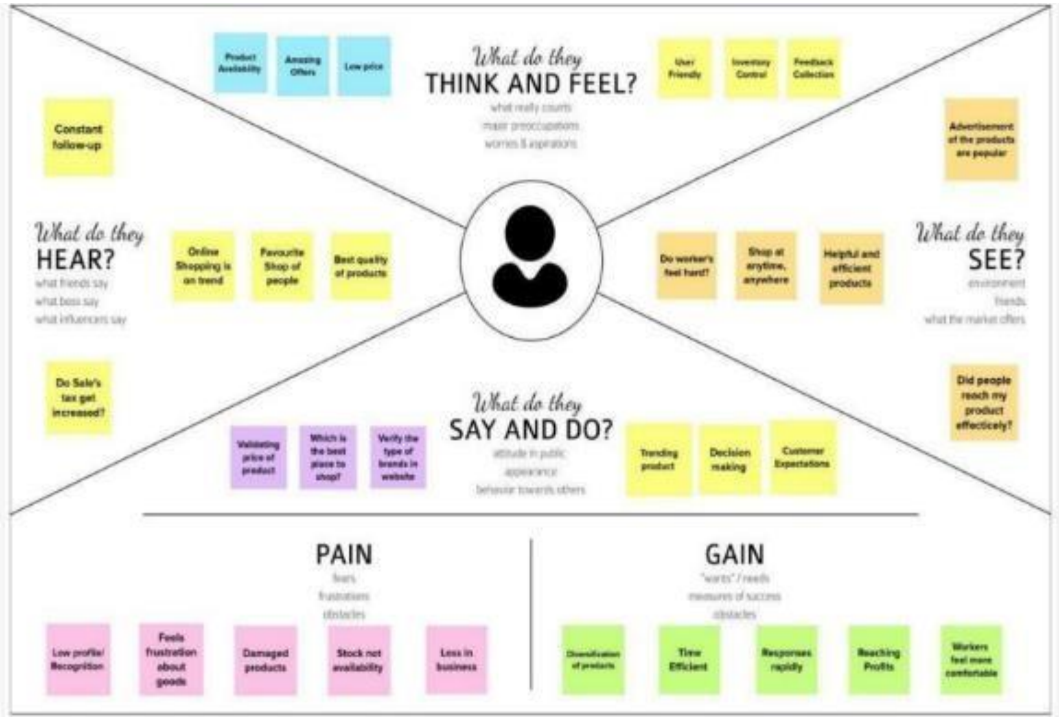
Reference:

[1] J.C.F. Ehrenthal et.al, “Demand seasonality in retail inventory management”, European Journal of Operational Research, Volume 238, Issue 2,2014, Pages 527-539, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2014.03.030. [2] Xin Tian a,b, Haoqing Wang a,b, Erjiang E c,*, “Forecasting intermittent demand for inventory management by retailers: A new approach”, Journal of Retailing and Consumer Services, Volume 62, September 2021, 102662,

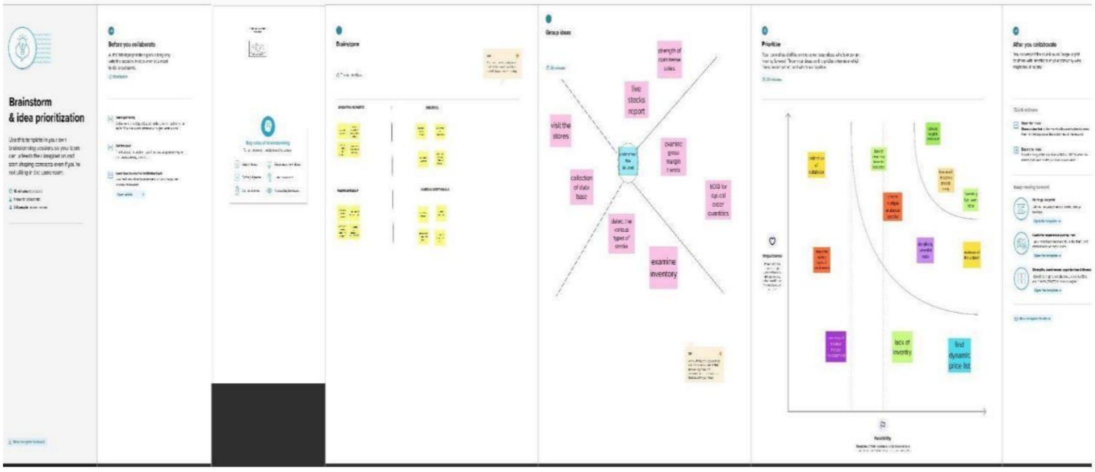
3 .IDEATION & PROPOSED SOLUTION:

➢ Empathy Map Canvas

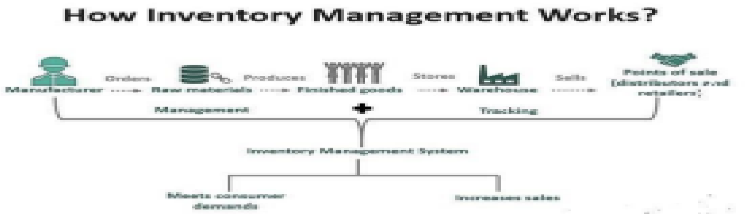
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes



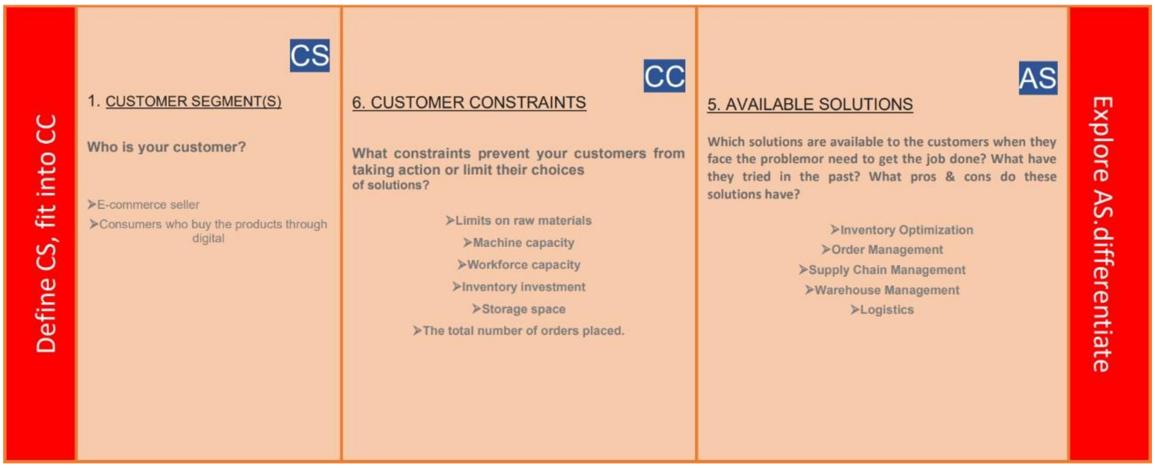
➢ **Brainstorming :**



- **Proposed Solution**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Retailers who run their business with large scale or small scale stocks. It is crucial for an organization today to understand its inventory to achieve both efficient and fast operations, that too, at an affordable cost. Lack of the right inventory at the right time can mean back orders, excess inventory, etc. These drive up costs. Late delivery due to stock-outs is bound to give you a bad reputation. Inaccurate calculations of stock and price. Late deliveries are due to late planning. Poor tracking may lead to back orders. Overstocking of discounted products and neglecting the trends of seasonal sales may result in excess inventory. Therefore considering the economic crisis of the retailers and to reduce the manpower efficiently while handling data, it is very important to have a best inventory management system for retailers.
2.	Idea / Solution description	Applications have been developed to help retailers track and manage stocks related to their own products. The System will ask retailers to create their accounts by providing essential details. Retailers can access their accounts by logging into the application. Once retailers successfully log in to the application they can update their inventory details, also users will be able to add new stock by submitting essential details related to the stock. They can view details of the current inventory. The System will automatically send an email alert to the retailers if there is no stock found in their accounts. So that they can order new stock.
3.	Novelty / Uniqueness	User can track the record of goods available using the application. Inventory tracking helps to improve inventory management and ensures that having optimal stock available to fulfil orders. Reduces manpower , cost and saves time. Emails will be sent automatically While stocks are not available. Makes the business process more efficient. Improves organizations performance.
4.	Social Impact / Customer Satisfaction	Customer satisfaction is the key for success of a business.The availability of product is just one way in which an inventory management system creates customer satisfaction. Inventory management systems are designed to monitor product availability, determine purchasing schedules for better customer interaction.
5.	Business Model (Revenue Model)	 How Inventory Management Works? Manufacturer Orders Raw materials Produces Finished goods Stores Warehouse Sells Points of sale (distributors and retailers) Management Tracking Inventory Management System Meets consumer demands Increases sales
6.	Scalability of the Solution	Scalability is an aspect or rather a functional quality of a system, software or solution. This proposed system for inventory management system can accommodate expansion without restricting the existing workflow and ensure an increase in the output or efficiency of the process.

Problem Solution fit2:



Define CS, fit into CC

CS

1. CUSTOMER SEGMENT(S)

Who is your customer?

➢E-commerce seller

➢Consumers who buy the products through digital

CC

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions?

➢Limits on raw materials

➢Machine capacity

➢Workforce capacity

➢Inventory investment

➢Storage space

➢The total number of orders placed.

AS

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problemor need to get the job done? What have they tried in the past? What pros & cons do these solutions have?

➢Inventory Optimization

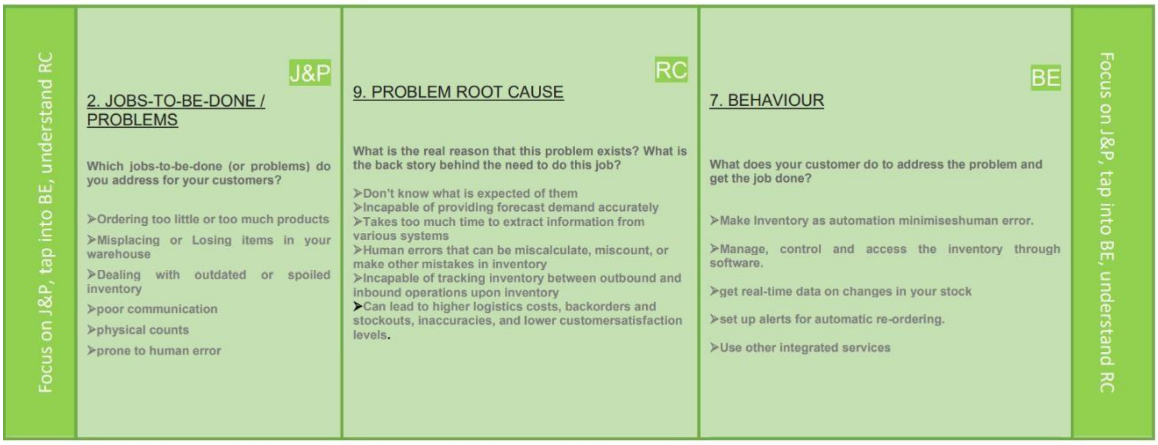
➢Order Management

➢Supply Chain Management

➢Warehouse Management

➢Logistics

Explore AS.differentiate



Focus on J&P, tap into BE, understand RC

J&P

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers?

➢Ordering too little or too much products

➢Misplacing or Losing items in your warehouse

➢Dealing with outdated or spoiled inventory

➢poor communication

➢physical counts

➢prone to human error

RC

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists? What is the back story behind the need to do this job?

➢Don't know what is expected of them

➢Incapable of providing forecast demand accurately

➢Takes too much time to extract information from various systems

➢Human errors that can be miscalculate, miscount, or make other mistakes in inventory

➢Incapable of tracking inventory between outbound and inbound operations upon inventory

➢Can lead to higher logistics costs, backorders and stockouts, inaccuracies, and lower customersatisfaction levels.

BE

7. BEHAVIOUR

What does your customer do to address the problem and get the job done?

➢Make Inventory as automation minimiseshuman error.

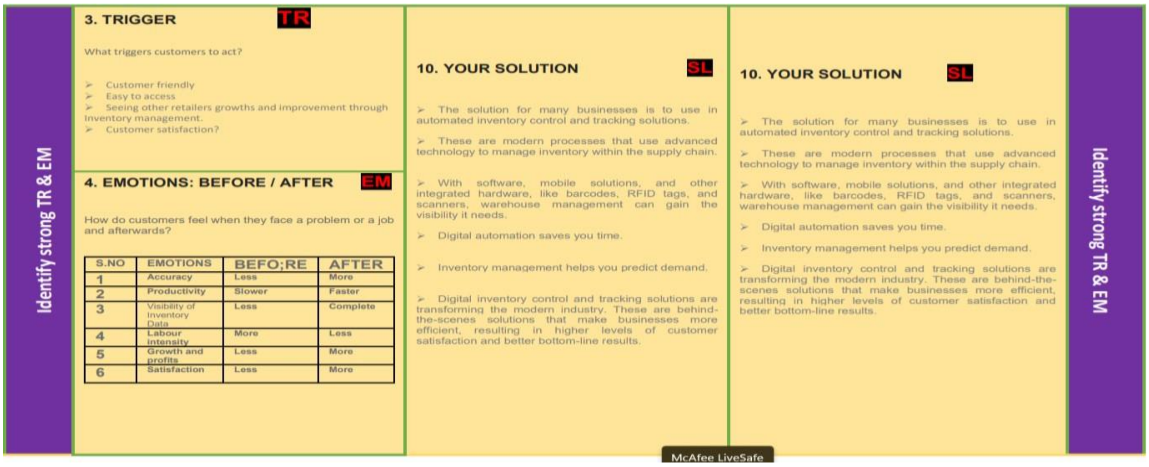
➢Manage, control and access the inventory through software.

➢get real-time data on changes in your stock

➢set up alerts for automatic re-ordering.

➢Use other integrated services

Focus on J&P, tap into BE, understand RC



Identify strong TR & EM

3. TRIGGER

TR

What triggers customers to act?

➢ Customer friendly

➢ Easy to access

➢ Seeing other retailers growths and improvement through Inventory management.

➢ Customer satisfaction?

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?

S.NO	EMOTIONS	BEFO;RE	AFTER
1	Accuracy	Less	More
2	Productivity	Slower	Faster
3	Visibility of Inventory Data	Less	Complete
4	Labour intensity	More	Less
5	Growth and profits	Less	More
6	Satisfaction	Less	More

10. YOUR SOLUTION

SL

➢ The solution for many businesses is to use in automated inventory control and tracking solutions.

➢ These are modern processes that use advanced technology to manage inventory within the supply chain.

➢ With software, mobile solutions, and other integrated hardware, like barcodes, RFID tags, and scanners, warehouse management can gain the visibility it needs.

➢ Digital automation saves you time.

➢ Inventory management helps you predict demand.

➢ Digital inventory control and tracking solutions are transforming the modern industry. These are behind-the-scenes solutions that make businesses more efficient, resulting in higher levels of customer satisfaction and better bottom-line results.

10. YOUR SOLUTION

SL

➢ The solution for many businesses is to use in automated inventory control and tracking solutions.

➢ These are modern processes that use advanced technology to manage inventory within the supply chain.

➢ With software, mobile solutions, and other integrated hardware, like barcodes, RFID tags, and scanners, warehouse management can gain the visibility it needs.

➢ Digital automation saves you time.

➢ Inventory management helps you predict demand.

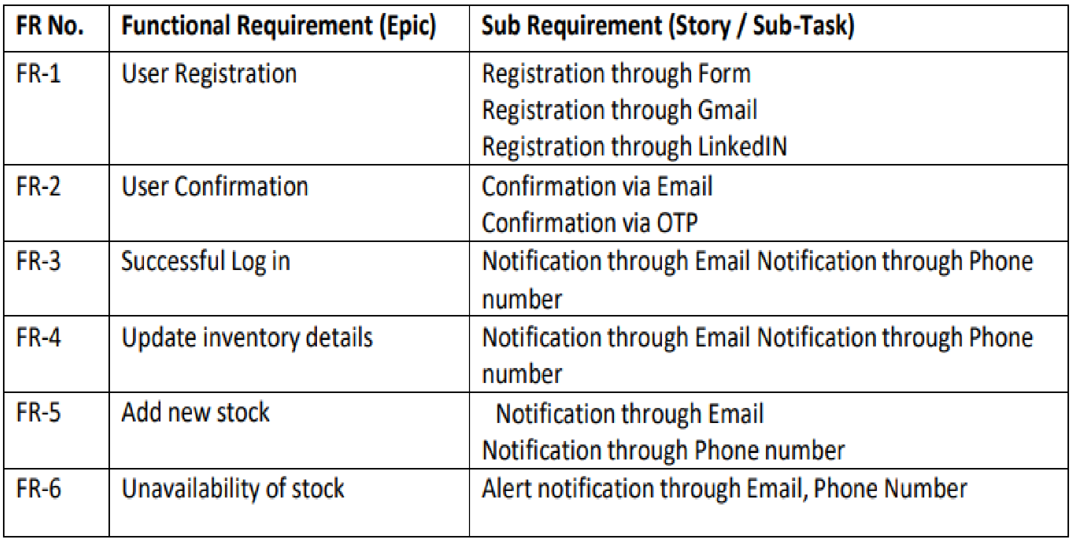
➢ Digital inventory control and tracking solutions are transforming the modern industry. These are behind-the-scenes solutions that make businesses more efficient, resulting in higher levels of customer satisfaction and better bottom-line results.

Identify strong TR & EM

McAfee LiveSafe

4. REQUIREMENT ANALYSIS

Functional requirement:



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Successful Log in	Notification through Email Notification through Phone number
FR-4	Update inventory details	Notification through Email Notification through Phone number
FR-5	Add new stock	Notification through Email Notification through Phone number
FR-6	Unavailability of stock	Alert notification through Email, Phone Number

***IBM Cloud Account** *** IBM Cloud CLI**

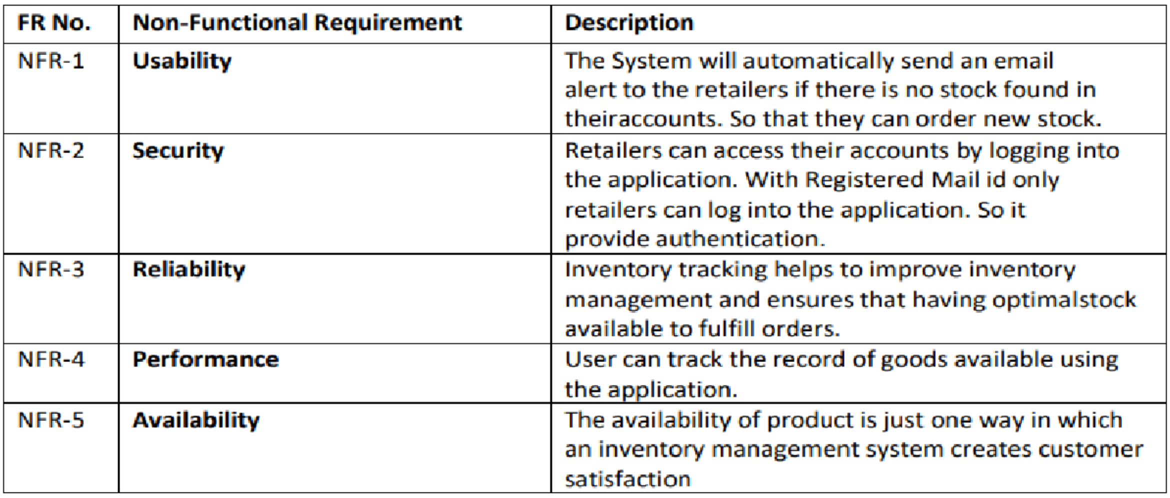
***Docker CLI** ***SendGrid** ***Node js**

***HTML & CSS** ***TimeScript** *** Angular Modules**

***Mongo DB** ***Mongo DB Express** ***IBM Object storage**

***Mongo DB Database IBM cloud**

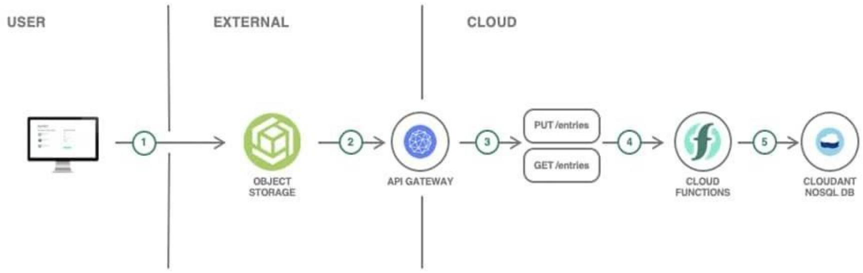
NON - Functional requirement:



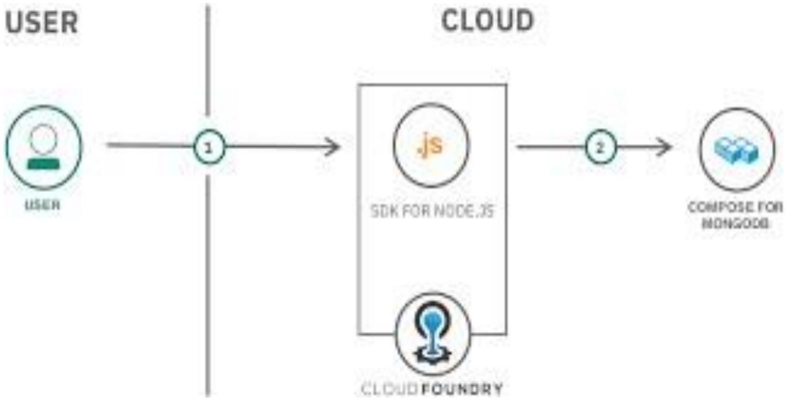
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The System will automatically send an email alert to the retailers if there is no stock found in theiraccounts. So that they can order new stock.
NFR-2	**Security**	Retailers can access their accounts by logging into the application. With Registered Mail id only retailers can log into the application. So it provide authentication.
NFR-3	**Reliability**	Inventory tracking helps to improve inventory management and ensures that having optimalstock available to fulfill orders.
NFR-4	**Performance**	User can track the record of goods available using the application.
NFR-5	**Availability**	The availability of product is just one way in which an inventory management system creates customer satisfaction

5. PROJECT DESIGN

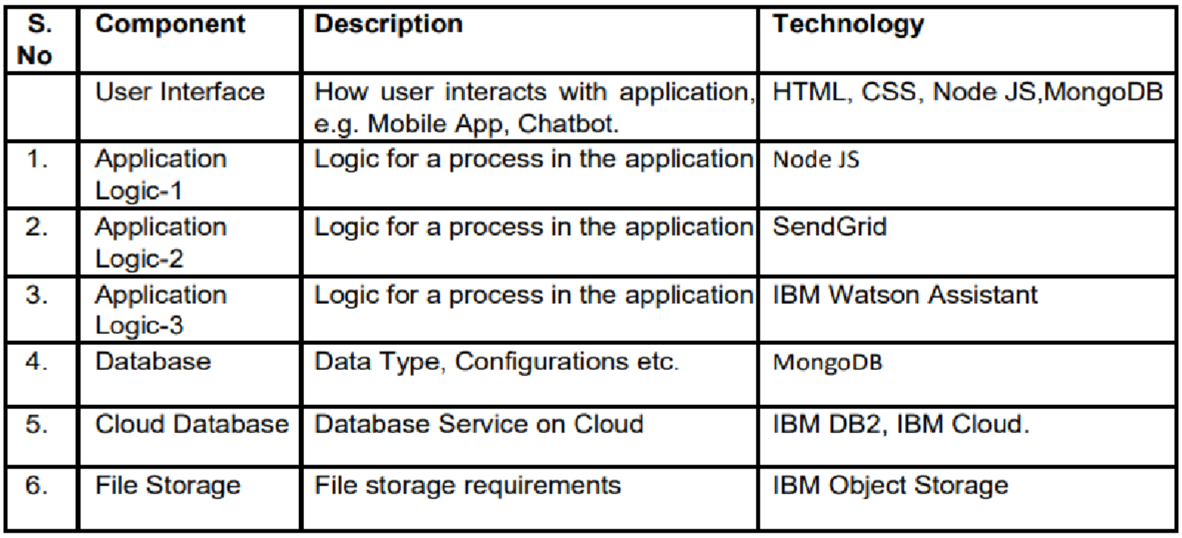
Data Flow Diagrams:



Technical Architecture:

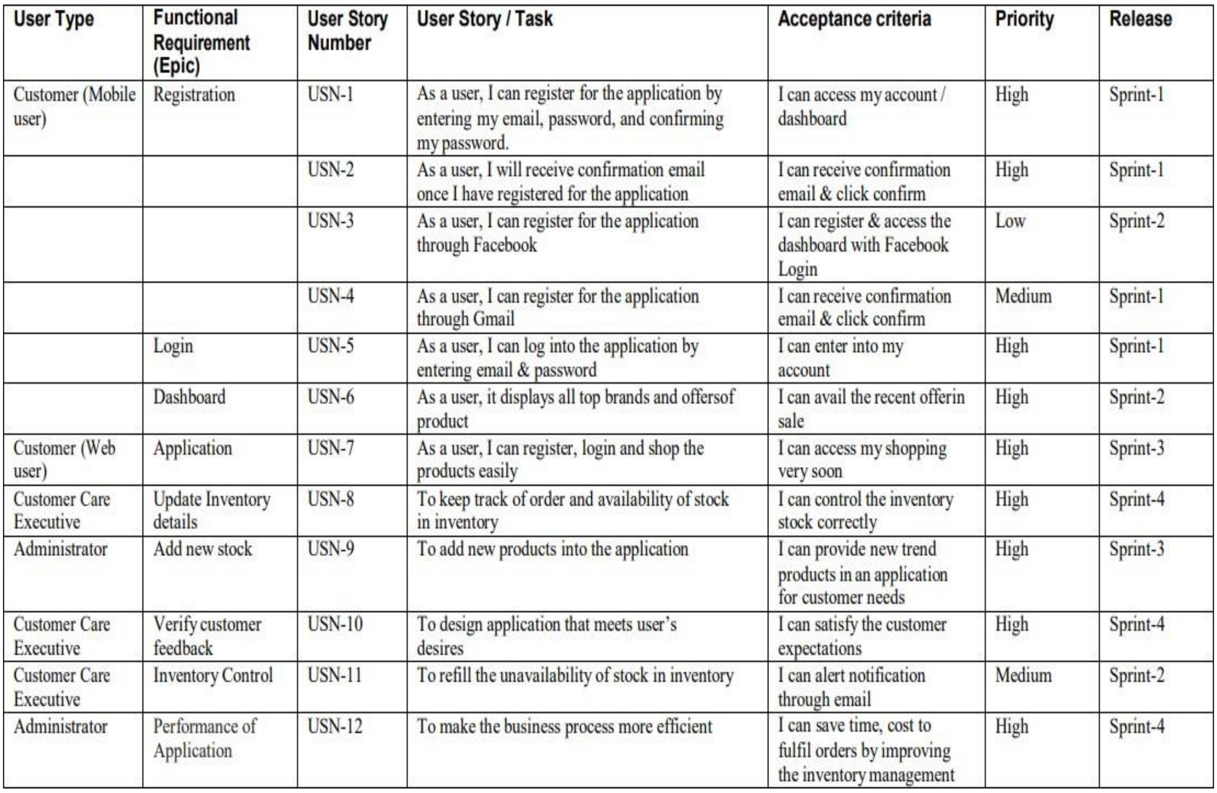


MONGO DB
(name: mongo, type DB)
express.js
EXPRESS JS
Web dev framework for Node JS
ANGULAR JS
Super Heroic Frontend Framework
NODE
NODE JS
Event based Concurrency Environment



S. No	Component	Description	Technology
	User Interface	How user interacts with application, e.g. Mobile App, Chatbot.	HTML, CSS, Node JS,MongoDB
1.	Application Logic-1	Logic for a process in the application	Node JS
2.	Application Logic-2	Logic for a process in the application	SendGrid
3.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
4.	Database	Data Type, Configurations etc.	MongoDB
5.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloud.
6.	File Storage	File storage requirements	IBM Object Storage

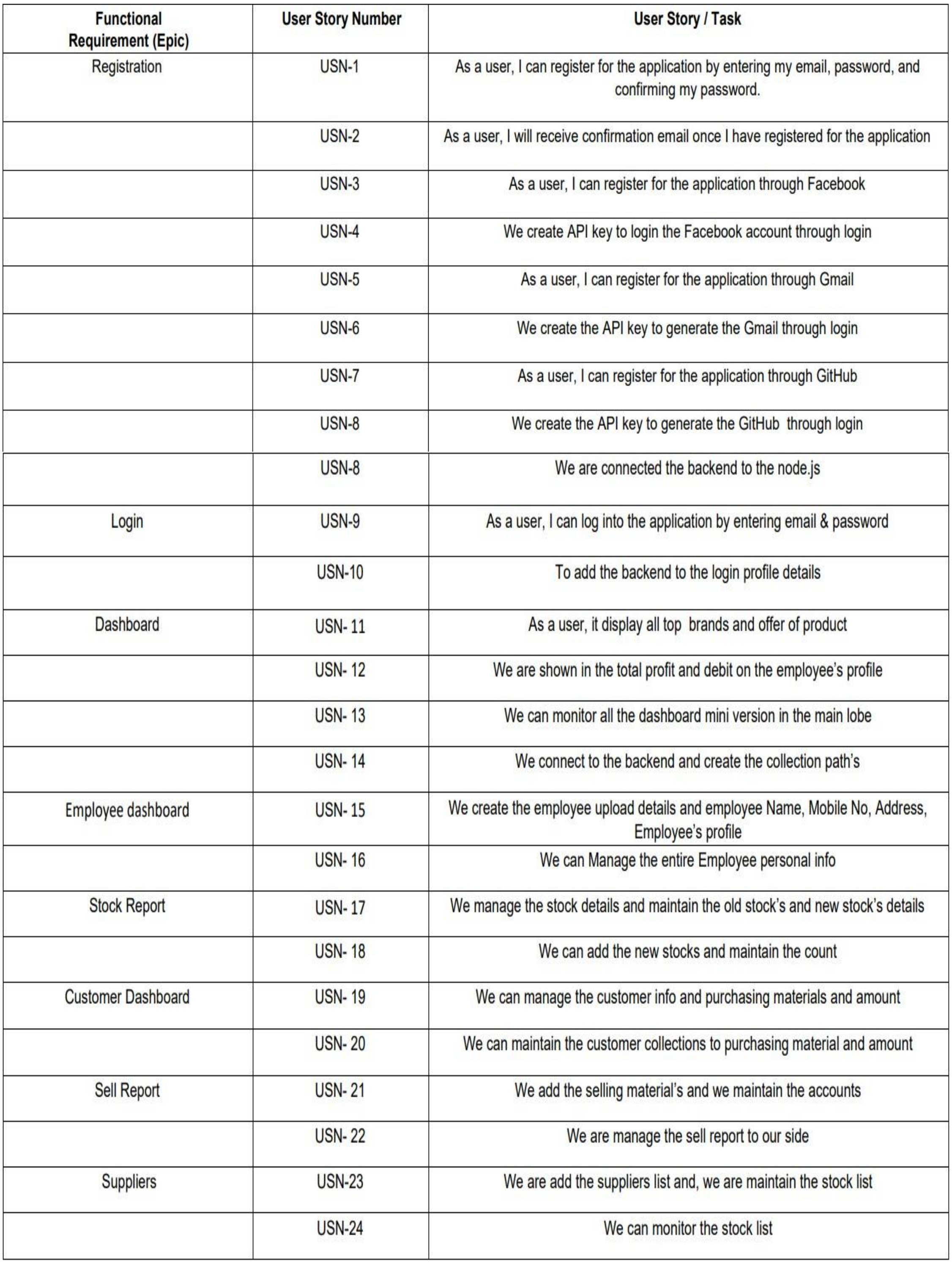
User Stories:



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application 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 receive confirmation email & click confirm	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can enter into my account	High	Sprint-1
	Dashboard	USN-6	As a user, it displays all top brands and offersof product	I can avail the recent offerin sale	High	Sprint-2
Customer (Web user)	Application	USN-7	As a user, I can register, login and shop the products easily	I can access my shopping very soon	High	Sprint-3
Customer Care Executive	Update Inventory details	USN-8	To keep track of order and availability of stock in inventory	I can control the inventory stock correctly	High	Sprint-4
Administrator	Add new stock	USN-9	To add new products into the application	I can provide new trend products in an application for customer needs	High	Sprint-3
Customer Care Executive	Verify customer feedback	USN-10	To design application that meets user's desires	I can satisfy the customer expectations	High	Sprint-4
Customer Care Executive	Inventory Control	USN-11	To refill the unavailability of stock in inventory	I can alert notification through email	Medium	Sprint-2
Administrator	Performance of Application	USN-12	To make the business process more efficient	I can save time, cost to fulfil orders by improving the inventory management	High	Sprint-4

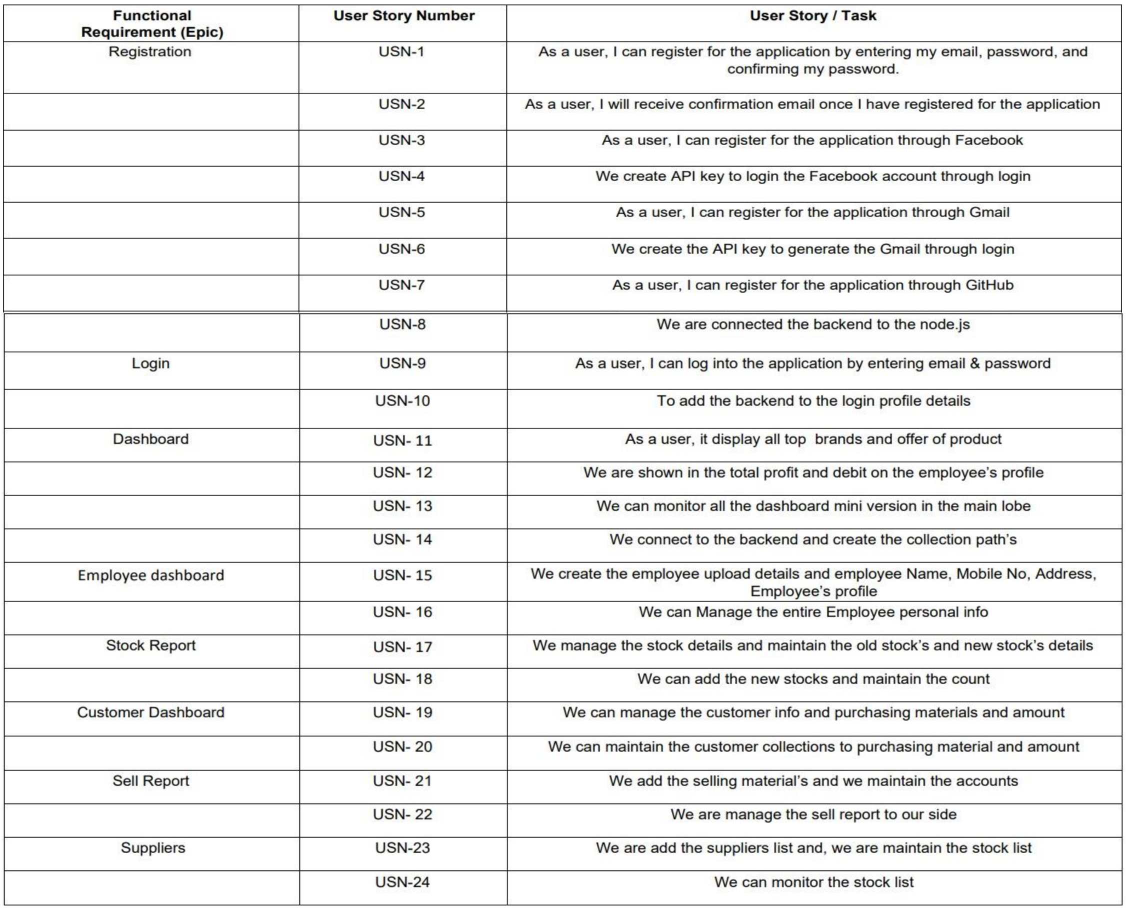
6. PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation:



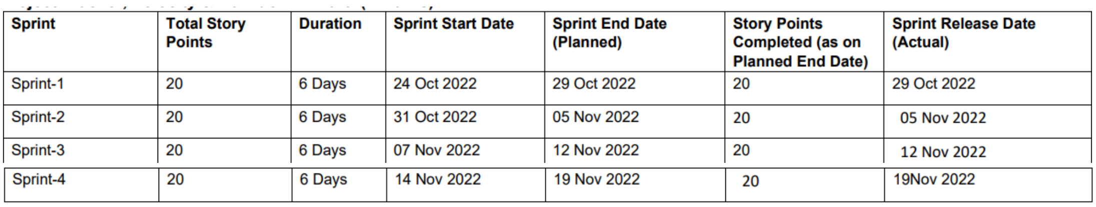
Functional Requirement (Epic)	User Story Number	User Story / Task
Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.
	USN-2	As a user, I will receive confirmation email once I have registered for the application
	USN-3	As a user, I can register for the application through Facebook
	USN-4	We create API key to login the Facebook account through login
	USN-5	As a user, I can register for the application through Gmail
	USN-6	We create the API key to generate the Gmail through login
	USN-7	As a user, I can register for the application through GitHub
	USN-8	We create the API key to generate the GitHub through login
	USN-8	We are connected the backend to the node.js
Login	USN-9	As a user, I can log into the application by entering email & password
	USN-10	To add the backend to the login profile details
Dashboard	USN- 11	As a user, it display all top brands and offer of product
	USN- 12	We are shown in the total profit and debit on the employee's profile
	USN- 13	We can monitor all the dashboard mini version in the main lobe
	USN- 14	We connect to the backend and create the collection path's
Employee dashboard	USN- 15	We create the employee upload details and employee Name, Mobile No, Address, Employee's profile
	USN- 16	We can Manage the entire Employee personal info
Stock Report	USN- 17	We manage the stock details and maintain the old stock's and new stock's details
	USN- 18	We can add the new stocks and maintain the count
Customer Dashboard	USN- 19	We can manage the customer info and purchasing materials and amount
	USN- 20	We can maintain the customer collections to purchasing material and amount
Sell Report	USN- 21	We add the selling material's and we maintain the accounts
	USN- 22	We are manage the sell report to our side
Suppliers	USN-23	We are add the suppliers list and, we are maintain the stock list
	USN-24	We can monitor the stock list

Sprint Delivery Schedule:



Functional Requirement (Epic)	User Story Number	User Story / Task
Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.
	USN-2	As a user, I will receive confirmation email once I have registered for the application
	USN-3	As a user, I can register for the application through Facebook
	USN-4	We create API key to login the Facebook account through login
	USN-5	As a user, I can register for the application through Gmail
	USN-6	We create the API key to generate the Gmail through login
	USN-7	As a user, I can register for the application through GitHub
	USN-8	We are connected the backend to the node.js
Login	USN-9	As a user, I can log into the application by entering email & password
	USN-10	To add the backend to the login profile details
Dashboard	USN- 11	As a user, it display all top brands and offer of product
	USN- 12	We are shown in the total profit and debit on the employee's profile
	USN- 13	We can monitor all the dashboard mini version in the main lobe
	USN- 14	We connect to the backend and create the collection path's
Employee dashboard	USN- 15	We create the employee upload details and employee Name, Mobile No, Address, Employee's profile
	USN- 16	We can Manage the entire Employee personal info
Stock Report	USN- 17	We manage the stock details and maintain the old stock's and new stock's details
	USN- 18	We can add the new stocks and maintain the count
Customer Dashboard	USN- 19	We can manage the customer info and purchasing materials and amount
	USN- 20	We can maintain the customer collections to purchasing material and amount
Sell Report	USN- 21	We add the selling material's and we maintain the accounts
	USN- 22	We are manage the sell report to our side
Suppliers	USN-23	We are add the suppliers list and, we are maintain the stock list
	USN-24	We can monitor the stock list

Reports from JIRA:



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

7. CODING & SOLUTIONING

Index.html

Index.html : (Compressed code)

```
<!doctype html><html lang="en"><head> <meta charset="utf-8"> <title>Inventory_Managing_System</title>
<base href="/"> <meta name="viewport" content="width=device-width, initial-scale=1"> <link rel="icon"
type="image/x-icon" href="favicon.ico"> <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"> <link
rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css">
</head><body > <div class="g-signin2" data-onsuccess="onSignIn" data-longtitle="true"></div > <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-
oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js" integrity="sha384-
IDwe1+LCz02ROU9k972gdyvl+AESN10+x7tBKgc9I5HFtuNz0wWnPclzo6p9vxnk"
crossorigin="anonymous"></script> <app-root>Loading. ..</app-root></body></html>
```

Header

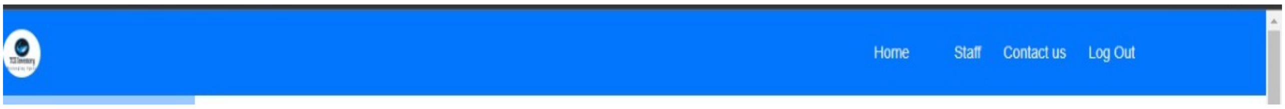
Header.component.html : (Compressed code)

```
<header id="my-header"> <!-- Nav Bar --> <nav class="navbar bg-dark navbar-expand-lg"> <a
href="index.html" class="navbar-brand"> <img src="assets\media\517c8a2df53f4cbfb2be39bdcca3a27b.png"
class="d-inline-block align-top" height="50" alt="TCE" loading="lazy" /> </a> <!-- Sidebar Collapse Button-->
<button class="navbar-toggler ml-auto mr-2" type="button" data-toggle="collapse" data-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation" > <i class="fas fa-bars"></i>
</button> <div class="collapse navbar-collapse" id="navbarNav" style="flex-direction: row-reverse; margin-
right: 8rem" > <ul class="navbar-nav ml-auto"> <li class="nav-item"> <a class="nav-link" href="dashboard">
<span class="item_outer"> Home </span> </a> </li> <li class="nav-item"> </li> <li class="nav-item"> <a
class="nav-link" href="employee"> <span class="item_outer"> Staff</span> </a> </li> <li class="nav-item"> <a
class="nav-link" href="contactus"> <span class="item_outer"> Contact us </span> </a> </li> <li class="nav-
item"> <a class="nav-link" href="/"> <i class="fas fa-sign-out-alt"></i> <span>Log Out</span> </a> </li> </ul>
</div> </nav> <!-- //Nav Bar --></header><!-- //HEADER -->
```

Header.component.ts: (Compressed code)

```
import { Component, OnInit } from '@angular/core';@Component({ selector: 'app-header', templateUrl:
'./header.component.html', styleUrls: ['./header.component.css']})export class HeaderComponent implements
OnInit { constructor() { } ngOnInit(): void { }}
```

Output:



Footer.component

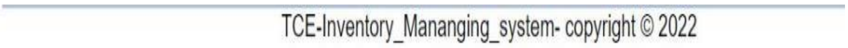
footer.component.html : (code)

```
<h5>TCE-Inventory_Mananging_system- copyright © 2022</h5>
```

footer.component.ts : (code)

```
import { Component, OnInit } from '@angular/core';@Component({ selector: 'app-footer', templateUrl: './footer.component.html', styleUrls: ['./footer.component.css']})export class FooterComponent implements OnInit { constructor() { } ngOnInit(): void { }}
```

Output:


TCE-Inventory_Mananging_system- copyright © 2022

Signup page

signup.component.html : (Compressed code)

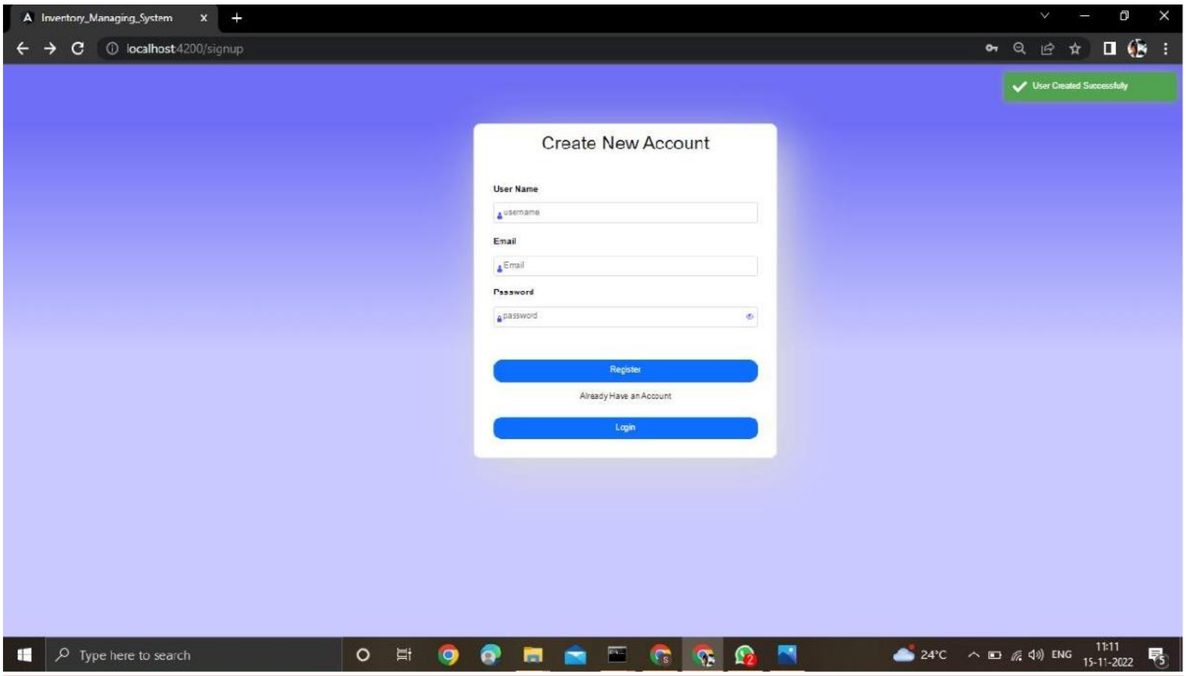
```
<div class="outer"> <div class="container"> <div class="row"> <div class="offset-md-2 col-lg-5 col-md-7 offset-
lg-4 offset-md-3"> <div class="panel border bg-white"> <div class="panel-heading"> <h2 class="pt-3 font-
weight-bold">Create New Account</h2> </div> <div class="panel-body pt-3 p-3 "> <form
[formGroup]="signupForm"> <div class="form-group mt-1 p-3 "> <label style="font-weight: bold;">User
Name</label> <div class="input-field"> <span class="fa fa-user pt-2"></span> <input
[class.error]="signupForm.controls['username'].dirty && signupForm.hasError('required','username')"
formControlName="username" type="text" placeholder="username"> </div> </div> <div class="form-group
mt-1 p-3 pt-0"> <label style="font-weight: bold;">Email</label> <div class="input-field"> <span class="fa fa-
user pt-2"></span> <input [class.error]="signupForm.controls['email'].dirty &&
signupForm.hasError('required','email')" formControlName="email" type="email" placeholder="Email"> </div>
</div> <div class="form-group p-3 pt-0"> <label style="font-weight: bold;">Password</label> <div
class="input-field"> <span class="fa fa-lock pt-2"></span> <input
[class.error]="signupForm.controls['password'].dirty && signupForm.hasError('required','password')"
formControlName="password" [type]="type" placeholder="password"> <span (click)="hideShowPass()"
class="fa {{eyeIcon}}"></span> </div> </div> <div class="p-3"> <button (click)="(onSubmit())" type="submit"
class="btn btn-primary btn-block mt-4 w-100">Register</button> </div> <div class="col-12 text-center">
<span>Already Have an Account</span> </div> <div class="p-3 pt-0"> <button routerLink="/" type="submit"
class="btn btn-primary btn-block mt-4 w-100">Login</button> </div> </form> </div> <!-- <div class="mx-3
my-2 py-2 bordert"> <div class="text-center py-3"> <a href="https://www.facebook.com" target="_blank"
class="px-2" ><img
src="https://www.google.com/imgres?imgurl=https%3A%2F%2Fupload.wikimedia.org%2Fwikipedia%2Fcomm
ons%2Fthumb%2F0%2F05%2FFacebook_Logo_%25282019%2529.png%2F768px-
Facebook_Logo_%25282019%2529.png&imgrefurl=https%3A%2F%2Fen.m.wikipedia.org%2Fwiki%2FFile%3AF
acebook_Logo_%25282019%2529.png&tbnid=88uK7HVkxanI6M&vet=12ahUKEwjdmuq2xPb6AhV0SXwKHf3A
CLsQMygBegUIARDcAQ..i&docid=_uHtFfQJL3HJjM&w=768&h=768&q=facebook%20icon%20png&ved=2ahUKE
wjdmuq2xPb6AhV0SXwKHf3ACLsQMygBegUIARDcAQ" alt=""></a> <a href="https://www.google.com"
target="_blank" class="px-2" ><img
src="https://www.google.com/imgres?imgurl=https%3A%2F%2Fcdn.imgbin.com%2F17%2F10%2F21%2Fgoogl
e-suite-icon-google-icon-
LmAAJV07.jpg&imgrefurl=https%3A%2F%2Fimgbin.com%2Fpng%2FLmAAJV07%2Fgoogle-suite-icon-google-
icon-png&tbnid=ial2z-6JPh4u8M&vet=12ahUKEwj3toL9xPb6AhWVumMGHZS-
Dk4QMygDegUIARDHAQ..i&docid=mECpCMCgMOTADM&w=800&h=800&q=google%20icon%20png&ved=2ah
UKEwj3toL9xPb6AhWVumMGHZS-Dk4QMygDegUIARDHAQ" alt=""></a> <a href="https://www.github.com"
target="_blank" class="px-2" ><img src="https://www.google.com/imgres?imgurl=https%3A%2F%2Flogos-
world.net%2Fwp-content%2Fuploads%2F2020%2F11%2FGitHub-Logo.png&imgrefurl=https%3A%2F%2Flogos-
world.net%2Fgithub-
logo%2F&tbnid=8AE_J5pAtu3iiM&vet=12ahUKEwj7iKKMxfb6AhWG_jgGHeuQDhMQMygFegUIARDMAQ..i&do
cid=TXCcjLeV5gKBaM&w=3840&h=2160&q=github%20icon%20png&ved=2ahUKEwj7iKKMxfb6AhWG_jgGHeu
QDhMQMygFegUIARDMAQ" alt=""></a> </div> </div>--> </div> </div> </div> </div></div>
```

signup.component.ts (compressed Code):

```
        import { Component, OnInit } from '@angular/core';import { FormBuilder, FormControl, FormGroup,
Validators,} from '@angular/forms';import ValidateForm from 'src/app/helpers/validateform';import {
AuthService } from 'src/app/services/auth.service';import { ToastrService } from 'ngx-toastr';import { Router }
```

```
from '@angular/router';@Component({ selector: 'app-signup', templateUrl: './signup.component.html',
styleUrls: ['./signup.component.css'],})export class SignupComponent implements OnInit { type: string =
'password'; isText: boolean = false; eyeIcon: string = 'fa fa-eye-slash'; signupForm!: FormGroup; constructor(
private toastr: ToastrService, private fb: FormBuilder, public authService: AuthService, public router: Router ) {}
ngOnInit(): void { this.signupForm = this.fb.group({ email: ['', Validators.required], username: ['',
Validators.required], roles: [['admin'], Validators.required], password: ['', Validators.required], }); }
hideShowPass() { this.isText = !this.isText; this.isText ? (this.eyeIcon = 'fa-eye') : (this.eyeIcon = 'fa-eye-flash');
this.isText ? (this.type = 'text') : (this.type = 'password'); } onSubmit() { if (this.signupForm.valid) {
this.authService.signup(this.signupForm.value).subscribe( (response) => { this.toastr.success('User Created
Successfully'); this.router.navigate(['/']); }, (err) => { this.toastr.error(err.message); } ); } else {
ValidateForm.validateAllFormFileds(this.signupForm); this.toastr.warning('Please Enter All the Fields'); } }}
```

OUTPUT:



localhost:4200/signup

User Created Successfully

Create New Account

Login Page

login.component.html: (Compressed Code)

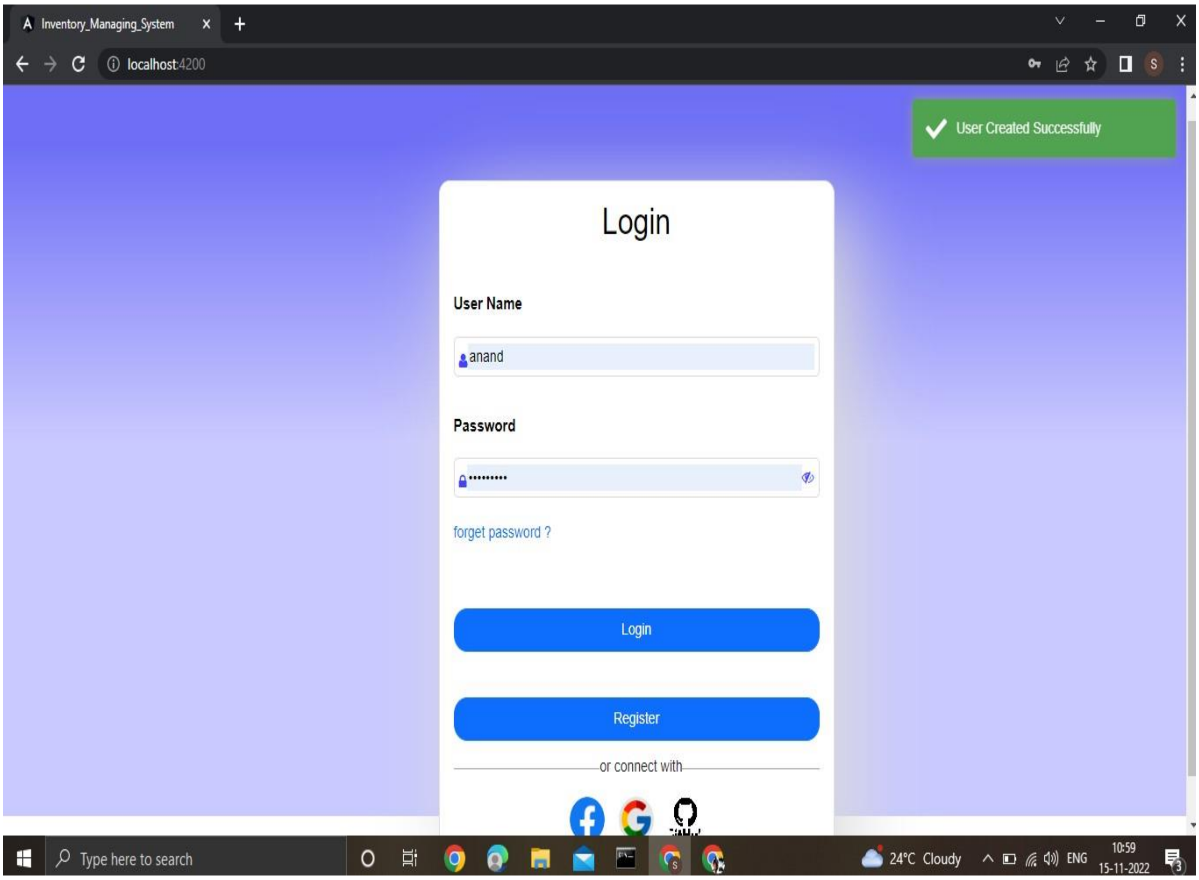
```
<div class="outer"> <div class="container"> <div class="row"> <div class="offset-md-2 col-lg-5 col-md-7 offset-lg-4 offset-md-3"> <div class="panel border bg-white"> <div class="panel-heading"> <h2 class="pt-3 font-weight-bold">Login</h2> </div> <div class="panel-body pt-3"> <form [formGroup]="loginForm"> <div class="form-group pt-3 p-3 "> <label style="font-weight: bold;">User Name</label> <div class="input-field mt-1" [class.error]="loginForm.controls['username'].dirty && loginForm.hasError('required','username')"> <span class="fa fa-user pt-2"></span> <input formControlName="username" type="text" placeholder="username"> </div> </div> <small *ngIf="loginForm.controls['username'].dirty && loginForm.hasError('required','username')" class="text-danger p-3">*username is required</small> <div class="form-group pt-3 p-3"> <label style="font-weight: bold;">Password</label> <div class="input-field mt-1" [class.error]="loginForm.controls['password'].dirty && loginForm.hasError('required','password')"> <span class="fa fa-lock pt-2"></span> <input formControlName="password" [type]="type" placeholder="password"> <span (click)="hideShowPass()" class="fa {{eyeIcon}}"></span> </div> </div> <small *ngIf="loginForm.controls['password'].dirty && loginForm.hasError('required','password')" class="text-danger p-3">*password is required</small> <div class="form-inline fr p-3 pt-1"> <a id="forget" class="font-weight-bold">forget password ?</a> </div> <div class="p-3"> <button (click)="(onSubmit())" type="submit" class="btn btn-primary btn-block mt-4 w-100">Login</button> </div> <div class="p-3 pt-0"> <button routerLink="/signup" type="submit" class="btn btn-primary btn-block mt-4 w-100">Register</button> </div> </form> </div> <div class="mx-3 my-2 py-2 bordert"> <div class="text-center py-3"> <a href="https://www.facebook.com" target="_blank" class="px-2"><img src="https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Facebook_Logo_%282019%29.png/768px-Facebook_Logo_%282019%29.png" alt=""></a> <a href="https://www.google.com" target="_blank" class="px-2"><img src="https://cdn.imgbin.com/17/10/21/google-suite-icon-google-icon-LmAAJV07.jpg" alt=""></a> <a href="https://www.github.com" target="_blank" class="px-2"><img src="https://logos-world.net/wp-content/uploads/2020/11/GitHub-Logo.png" alt=""></a> </div> </div> </div> </div> </div> </div></div>
```

login.component.ts:

```
            <div class="outer"> <div class="container"> <div class="row"> <div class="offset-md-2 col-lg-5 col-md-7 offset-lg-4 offset-md-3"> <div class="panel border bg-white"> <div class="panel-heading"> <h2 class="pt-3 font-weight-bold">Login</h2> </div> <div class="panel-body pt-3"> <form [formGroup]="loginForm"> <div class="form-group pt-3 p-3 "> <label style="font-weight: bold;">User Name</label> <div class="input-field mt-1" [class.error]="loginForm.controls['username'].dirty && loginForm.hasError('required','username')"> <span class="fa fa-user pt-2"></span> <input formControlName="username" type="text" placeholder="username"> </div> </div> <small *ngIf="loginForm.controls['username'].dirty && loginForm.hasError('required','username')" class="text-danger p-3">*username is required</small> <div class="form-group pt-3 p-3"> <label style="font-weight: bold;">Password</label> <div class="input-field mt-1" [class.error]="loginForm.controls['password'].dirty && loginForm.hasError('required','password')"> <span class="fa fa-lock pt-2"></span> <input formControlName="password" [type]="type" placeholder="password"> <span (click)="hideShowPass()" class="fa {{eyeIcon}}"></span> </div> </div> <small *ngIf="loginForm.controls['password'].dirty && loginForm.hasError('required','password')" class="text-danger p-3">*password is required</small> <div class="form-inline fr p-3 pt-1"> <a id="forget" class="font-weight-bold">forget password ?</a> </div> <div class="p-3"> <button (click)="(onSubmit())" type="submit" class="btn btn-primary btn-block mt-4 w-100">Login</button> </div> <div class="p-3 pt-0"> <button routerLink="/signup" type="submit" class="btn btn-primary btn-block mt-4 w-100">Register</button> </div> </form> </div> <div class="mx-3 my-2 py-2 bordert"> <div class="text-center py-3"> <a href="https://www.facebook.com" target="_blank" class="px-
```

2"><img src="https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Facebook_Logo_%282019%29.png/768px-Facebook_Logo_%282019%29.png" alt=""></a> <a href="https://www.google.com" target="_blank" class="px-2"><img src="https://cdn.imgbin.com/17/10/21/google-suite-icon-google-icon-LmAAJV07.jpg" alt=""></a> <a href="https://www.github.com" target="_blank" class="px-2"><img src="https://logos-world.net/wp-content/uploads/2020/11/GitHub-Logo.png" alt=""></a> </div> </div> </div> </div> </div> </div></div>

OUTPUT:



User Created Successfully

Login

User Name

anand

Password

••••••••••

NavBar Page

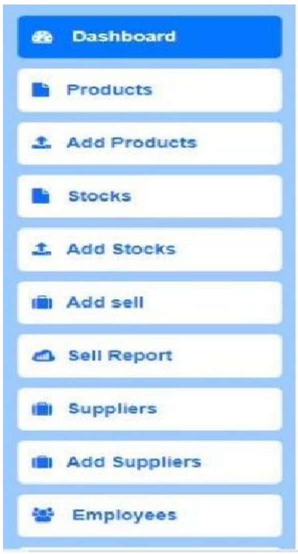
NAVBAR.COMPONENT HTML: *(Compressed code)

```
<!-- SIDEBAR --><nav id="sidebar"> <ul class="nav nav-pills nav-fill"> <br><br> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/dashboard') ? 'active' : '' " href="dashboard"> <i class="fa fa-dashboard"></i> Dashboard</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/report') ? 'active' : '' " href="report"><i class="fa fa-file"></i>Products</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/addreport') ? 'active' : '' " href="addreport"><i class="fa fa-upload"></i>Add Products</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/stock') ? 'active' : '' " href="stock"><i class="fa fa-file"></i>Stocks</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/addstock') ? 'active' : '' " href="addstock"><i class="fa fa-upload"></i>Add Stocks</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/addcustomer') ? 'active' : '' " href="addcustomer"><i class="fa fa-suitcase"></i>Add sell</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/customer') ? 'active' : '' " href="customer"><i class="fa fa-sellsy"></i>Sell Report</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/suppliers') ? 'active' : '' " href="suppliers"><i class="fa fa-suitcase"></i>Suppliers</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/addsuppliers') ? 'active' : '' " href="addsuppliers"><i class="fa fa-suitcase"></i>Add Suppliers</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/employee') ? 'active' : '' " href="employee"><i class="fa fa-users"></i> Employees</a> </li> <li class="nav-item"> <a class="nav-link" [ngClass]="router.url.includes('/addemployee') ? 'active' : '' " href="addemployee"><i class="fa fa-upload"></i>New Employees</a> </li><br><br><br><br><br><br><br><br><br><br> </ul></nav><!-- //SIDEBAR MENU -->
```

NAVBAR.COMPONENT.ts: *(Compressed code)

```
import { Component, OnInit } from '@angular/core';import { Router } from '@angular/router';@Component({ selector: 'app-navbar', templateUrl: './navbar.component.html', styleUrls: ['./navbar.component.css']})export class NavbarComponent implements OnInit { public customerRoute: string = '/customer'; constructor(public router: Router) { } ngOnInit(): void { }}
```

OUTPUT:



Dashboard

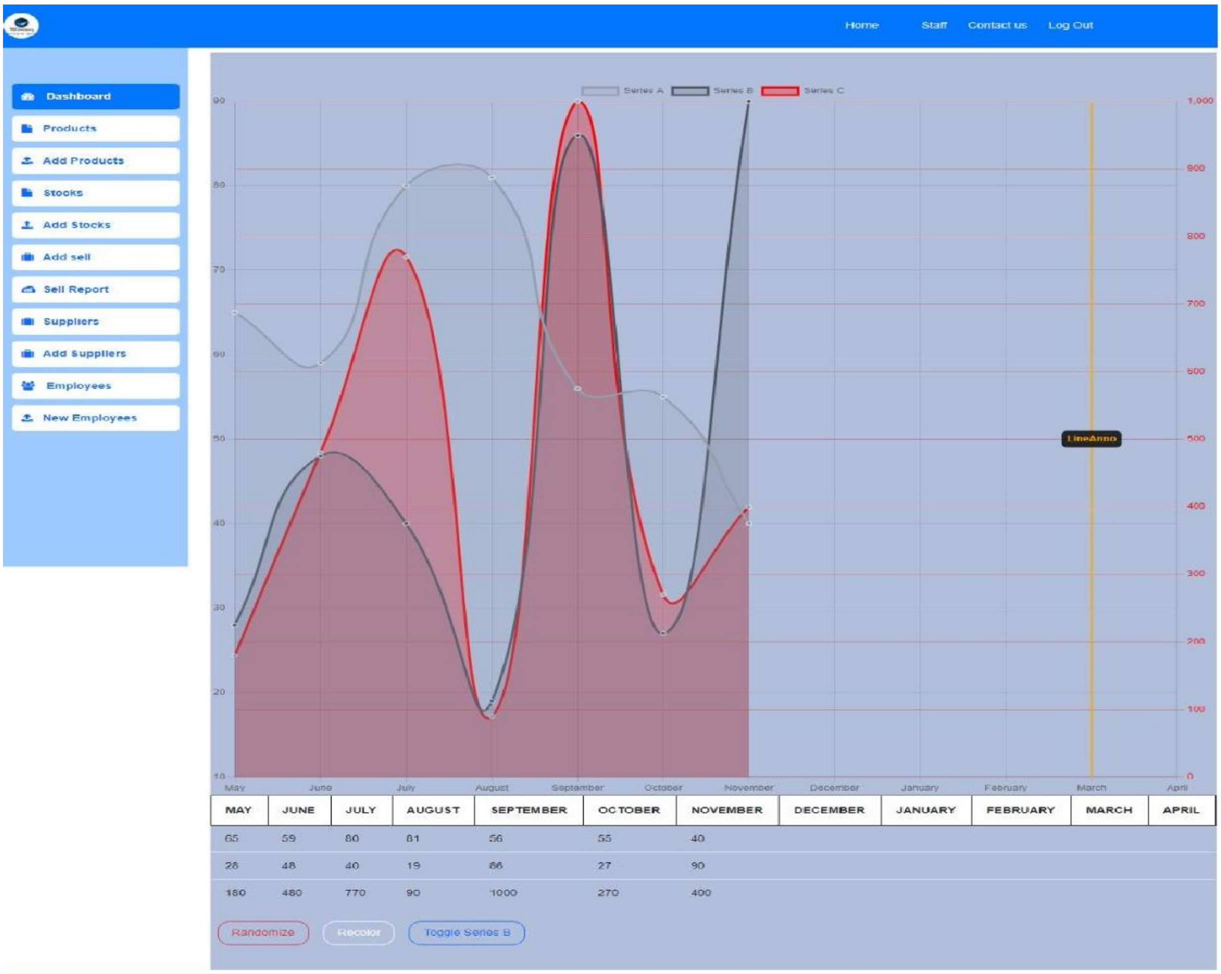
dashboard.component.html(Compressed code)

```
<app-header></app-header><div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div
class="col-10"> <div class="content-wrapper mt-2"> <app-line-chart class="p-3"></app-line-chart> </div>
</div></div>
```

dashboard.component.ts(Compressed code)

```
import { Component, OnInit } from '@angular/core';@Component({ selector: 'app-dashboard', templateUrl:
'./dashboard.component.html', styleUrls: ['./dashboard.component.css']})export class DashboardComponent
implements OnInit { constructor() { } ngOnInit(): void { }}
```

OUTPUT:



MAY	JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER	JANUARY	FEBRUARY	MARCH	APRIL
65	59	80	81	56	55	40					
28	48	40	19	86	27	90					
180	480	770	90	1000	270	400					

ADDCUSTOMER.COMPONENT

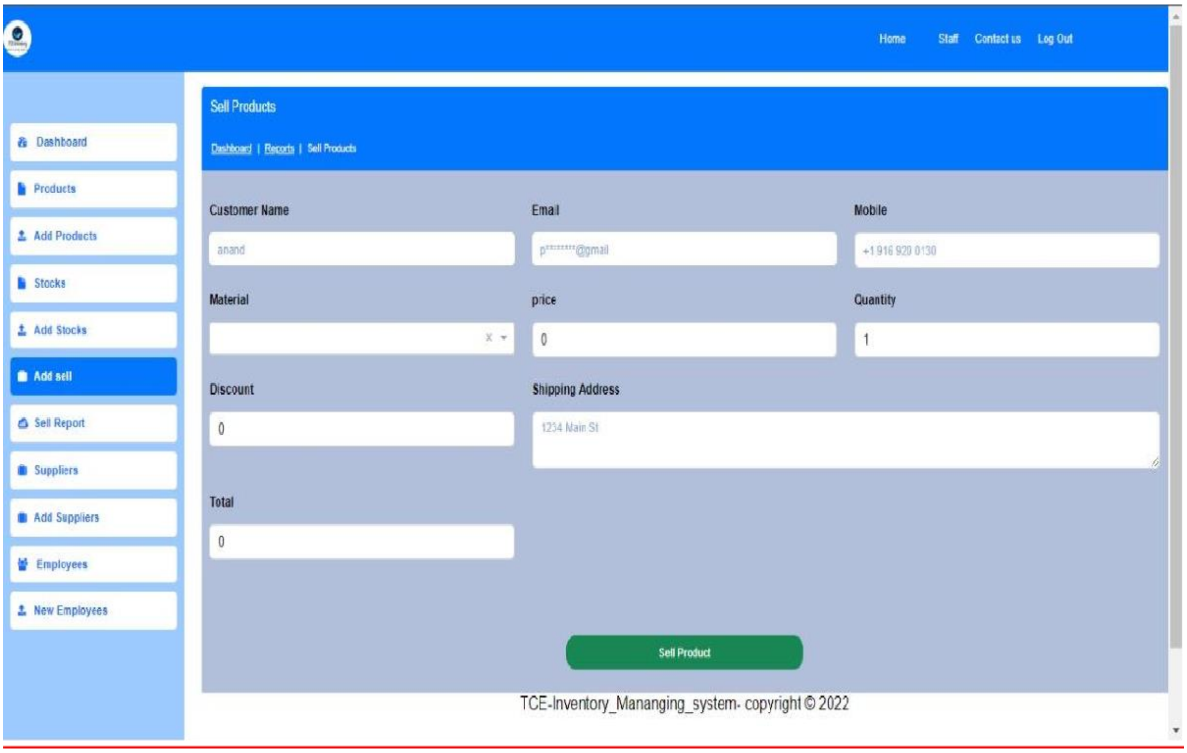
addcustomer.component.html:(Compressed Code)

```
<div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0"> <div>
<section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius clearfix"> <div
class="float-left"> <h3>Sell Products</h3> </div> <div class="float-right"> <div class="d-flex"> <nav aria-
label="breadcrumb"> <ol class="breadcrumb border-0"> <li class="breadcrumb-item"> <a
href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"><a href="customer">Reports</a></li> <li
class="breadcrumb-item active" aria-current="page"> Sell Products </li> </ol> </nav> </div> </div> </div>
<div class="content-wrapper"> <div class="row mx-0 mt-2"> <div class="col-lg-12"> <div class="form-row
row"> <div class="form-group col-md-4"> <label for="firstname">Customer Name</label> <input type="text"
class="form-control" id="name" placeholder="anand" [(ngModel)]="name" /> </div> <div class="form-group
col-md-4"> <label for="Email">Email</label> <input type="email" class="form-control" id="name"
placeholder="p********@gmail" [(ngModel)]="email" /> </div> <div class="form-group col-md-4"> <label
for="mobile">Mobile</label> <input type="number" class="form-control" id="mobile" placeholder="+1 916
920 0130" [(ngModel)]="mobile" /> </div> <div class="form-group col-md-4 mt-4"> <label
for="lasttname">Material</label> <ng-select [items]="report" #select bindLabel="ptitle"
dropdownPosition="auto" bindValue="ptitle" labelForId="album" placeholder="Select album"
[virtualScroll]="true" [(ngModel)]="material" (change)="onAdd($event)"> <ng-template ng-option-tmp let-
item="item" let-search="searchTerm"> <div><span>Title: </span><span>{{item.ptitle}}</span></div>
<small><b>P-code:</b> {{item.pcode}}</small> <small><b>Brand:</b> {{item.pbrand}}</small> </ng-
template> </ng-select> </div> <div class="form-group col-md-4 mt-4"> <label for="price">price</label> <input
type="number" class="form-control" id="price" placeholder="item" [(ngModel)]="price" readonly /> </div>
<div class="form-group col-md-4 mt-4"> <label for="price">Quantity</label> <input type="number"
class="form-control" (change)="changeData()" id="quantity" placeholder="item" [(ngModel)]="qty" /> </div>
<div class="form-group col-md-4 mt-4"> <label for="price">Discount</label> <input type="number"
class="form-control" id="discount" placeholder="item" [(ngModel)]="discount" /> </div> <div class="form-
group col-md-8 mt-4"> <label for="inputAddress">Shipping Address</label> <textarea type="text"
class="form-control" id="inputAddress" placeholder="1234 Main St" [(ngModel)]="address"></textarea>
</div> <div class="form-group col-md-4 mt-4"> <label for="price">Total</label> <input type="number"
class="form-control" id="discount" placeholder="item" readonly [(ngModel)]="total" /> </div> </div>
<br><br><br> <div class="row text-center"> <div class="col-12 text-center"> <button class="btn btn-success w-
25" (click)="onSubmit()">Sell Product</button> </div> </div> </div> </div> </div> </div> </section> <!--
//PAGE END --> </div> </div></div>
```

addcustomer.component.ts(Compressed Code):

```
import { Component, OnInit } from '@angular/core';import { ActivatedRoute, Router } from
'@angular/router';import { Customer } from 'src/app/models/customer.model';import { Report } from
'src/app/models/report.model';import { CustomerService } from 'src/app/services/customer.service';import {
ReportService } from 'src/app/services/report.service';import { ToastrService } from 'ngx-
toastr';@Component({ selector: 'app-addcustomer', templateUrl: './addcustomer.component.html', styleUrls:
['./addcustomer.component.css'],})export class AddCustomerComponent implements OnInit { public name = '';
public mobile = ''; public email = ''; public address = ''; public material = ''; public price: number = 0; public total:
number = 0; public qty: number = 1; public discount: number = 0; public id: any = null; public report: Report[] =
[]; pid: any; totalStock: any; constructor( private customerService: CustomerService, private reportService:
ReportService, private route: ActivatedRoute, private toastr: ToastrService, private router: Router ) {
this.getReport(); } ngOnInit(): void { this.route.queryParams.subscribe((params) => { this.id = params['id'];
this.getDetails(); }); } getReport() { this.reportService.getAll().subscribe({ next: (data: any) => { this.report =
data; }, error: (err: any) => {}, }); } getDetails() { this.customerService.get(this.id).subscribe({ next: (data: any) =>
{ this.name = data.name; this.email = data.email; this.mobile = data.mobile; this.material = data.material;
this.address = data.address; this.price = data.price; this.total = data.total; this.qty = data.qty; this.discount =
data.discount; }, error: (err) => {}, }); } onSubmit(): void { if ( this.name && this.email && this.mobile &&
this.material && this.qty && this.total && this.price && this.address ) { if (parseInt(this.totalStock) < this.qty) {
this.toastr.warning( `Only ${this.totalStock} Stocks are available for this product` ); return; } var data = { name:
this.name, email: this.email, mobile: this.mobile, material: this.material, address: this.address, price: this.price,
total: this.total, qty: this.qty, discount: this.discount, pid: this.pid, tstock: parseInt(this.totalStock) - this.qty, };
if (this.id) { this.customerService.update(this.id, data).subscribe({ next: (data: any) => {
this.toastr.success('Sales Updated Successfully'); this.navigate(); }, error: (err: any) => {}, }); } else {
this.customerService.create(data).subscribe({ next: (data: any) => { this.toastr.success('Sales Created
Successfully'); this.navigate(); }, error: (err: any) => {}, }); } } else { this.toastr.warning('Please Fill all the fields of
the form'); } } navigate() { this.router.navigate(['customer']); } onAdd(event: any) { console.log(event); if
(event.id) { this.pid = event.id; } else if (event._id) { this.pid = event._id; } this.totalStock = event.tstock; if
(!isNaN(parseFloat(event.cpi))) { this.price = parseFloat(event.cpi); this.total = this.price * this.qty; } else {
this.price = 0; this.qty = 0; } } changeData() { if (this.qty > 0) { this.total = this.price * this.qty; } else { this.qty =
0; } } checkDiscount() { if (this.qty > 0) { this.total = this.price * this.qty; } else { this.qty = 0; } }}
```

output:



Sell Products

Dashboard | Reports | Sell Products

Customer Name: anand

Email: p********@gmail

Mobile: +1 916 920 0130

Material:

price: 0

Quantity: 1

Discount: 0

Shipping Address: 1234 Main St

Total: 0

TCE-Inventory_Mananging_system- copyright © 2022

CUSTOMER.COMPONENT

customer.component.html(Compressed Code):

```
<div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0"> <div>
<section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius clearfix"> <div
class="float-left"> <h3>Sales Report</h3> </div> <div class="float-right"> <div class="d-flex"> <nav aria-
label="breadcrumb"> <ol class="breadcrumb"> <li class="breadcrumb-item"> <a
href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"> <a href="addcustomer">Sell
Products</a> </li> <li class="breadcrumb-item active" aria-current="addcustomer"> Sales Report </li> </ol>
</nav> </div> </div> </div> <div class="content-wrapper"> <div class="row mt-2"> <div class="col-lg-12"> <!--
Table start --> <div class="table-container"> <div class="table-responsive"> <table id="sort-table" class="table
custom-table m-0"> <thead> <tr> <th>Name</th> <th>Email</th> <th>Mobile</th> <th>Address</th>
<th>Material</th> <th>Price</th> <th>Quantity</th> <th>Discount</th> <th>Total</th> </tr> <tr *ngFor="let
items of customer"> <td>{{ items.name }}</td> <td>{{ items.email }}</td> <td>{{ items.mobile }}</td> <td>{{
items.address }}</td> <td>{{ items.material }}</td> <td>{{ items.price }}</td> <td>{{ items.qty }}</td> <td>{{
items.discount }}</td> <td>{{ items.total }}</td> </tr> </thead> </table> </div> </div> <!-- Table end --> </div>
</div> </div> </div> </section> <!-- //PAGE END -->
```

customer.component.ts(Compressed Code):

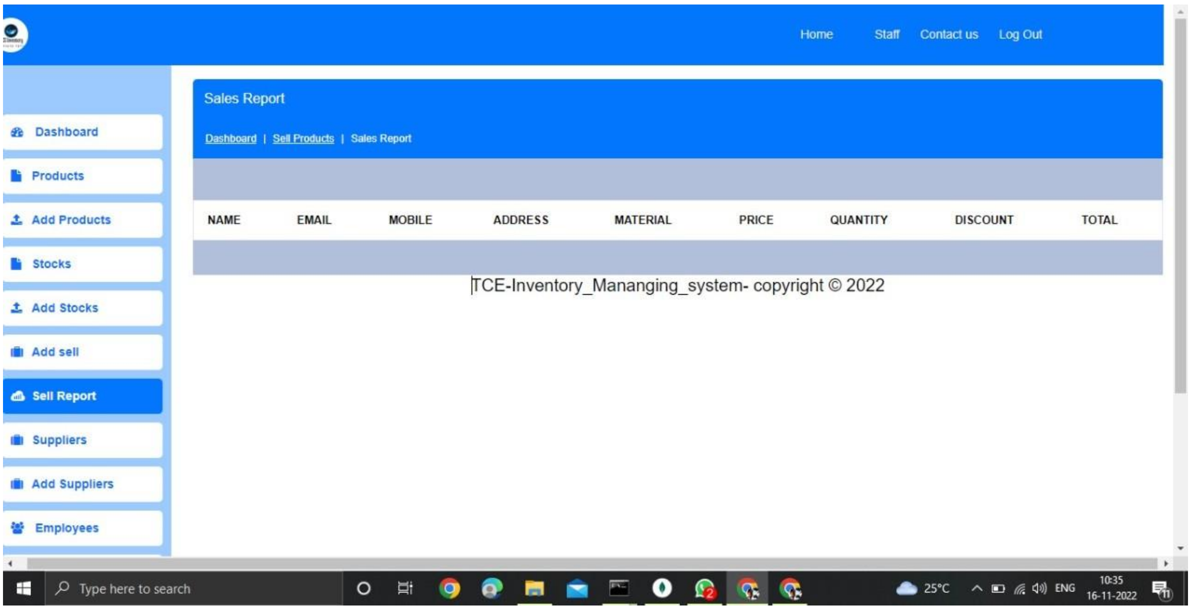
```
import { Component, OnInit } from '@angular/core';import { ActivatedRoute, Router } from
'@angular/router';import { Customer } from 'src/app/models/customer.model';import { CustomerService }
from 'src/app/services/customer.service';@Component({ selector: 'app-customer', templateUrl:
'./customer.component.html', styleUrls: ['./customer.component.css']})export class CustomerComponent
implements OnInit { public customer: Customer[] = []; constructor(private customerService: CustomerService,
private route: ActivatedRoute, private router: Router) { } ngOnInit(): void { this.getCustomer(); } getCustomer()
{ this.customerService.getAll().subscribe({ next: (data: any) => { this.customer = data; }, error: (err: any) => { }
}); } nextPage(id: any) { if (id._id) { id = id._id; } else { id = id.id; } this.router.navigate(['addcustomer'], {
queryParams: { id: id } }); }}
```

customer.component.html(Compressed Code):

```
<!-- PAGE START --><h3 style="padding: 10px; text-align: center">Customer Dashboard</h3><div> <section
class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius clearfix"> <div class="float-left">
<h3>Customers</h3> </div> <div class="float-right"> <div class="d-flex"> <nav aria-label="breadcrumb"> <ol
class="breadcrumb"> <li class="breadcrumb-item"> <a href="dashboard">Dashboard</a> </li> <li
class="breadcrumb-item"> <a href="addcustomer">Add Customer</a> </li> <li class="breadcrumb-item active"
aria-current="addcustomer"> Customers </li> </ol> </nav> </div> </div> </div> <div class="content-wrapper">
<div class="row mx-0 mt-2"> <div class="col-lg-8"> <form class="form-row"> <div class="col-lg mb-2"> <input
type="text" class="form-control" value="Name" /> </div> <div class="col-lg mb-2"> <input type="text"
class="form-control" value="mobile" /> </div> <div class="col-lg my-2 d-none"> <div class="custom-control
custom-radio custom-control-inline"> <input type="radio" id="customRadioInline1"
```

```
name="customRadioInline1" class="custom-control-input" checked /> <label class="custom-control-label"
for="customRadioInline1" >Distributor</label > </div> <div class="custom-control custom-radio custom-
control-inline"> <input type="radio" id="customRadioInline2" name="customRadioInline1" class="custom-
control-input" /> <label class="custom-control-label" for="customRadioInline2" >Customer</label > </div>
</div> <div class="col-lg"> <button type="submit" class="btn btn-primary mb-2"> <i class="ti-search mr-1"></i>
Search </button> </div> </form> </div> </div> <div class="row mt-2"> <div class="col-lg-12"> <!-- Table start --
> <div class="table-container"> <div class="table-responsive"> <table id="sort-table" class="table custom-table
m-0"> <thead> <tr> <th class="nosort" style="width: 18px"> <input type="checkbox" id="selectAll" /> </th>
<th>Name</th> <th>Mobile</th> <th>Email</th> <th>Address</th> </tr><tr *ngFor="let items of
customer"><td style="cursor: pointer" (click)="nextPage(items.id)"> &#9998;</td><td>{{ items.firstname
}}</td> <td>{{ items.mobile }}</td><td>{{ items.email }}</td><td>{{ items.address }}</td> </tr> </thead>
</table> </div> </div> <!-- Table end --> </div> </div> </div> </div> </section> <!-- //PAGE END --></div>
```

OUTPUT:



Sales Report

Dashboard | Sell Products | Sales Report

NAME	EMAIL	MOBILE	ADDRESS	MATERIAL	PRICE	QUANTITY	DISCOUNT	TOTAL

TCE-Inventory_Mananging_system- copyright © 2022

EMPLOYEE.COMPONENT

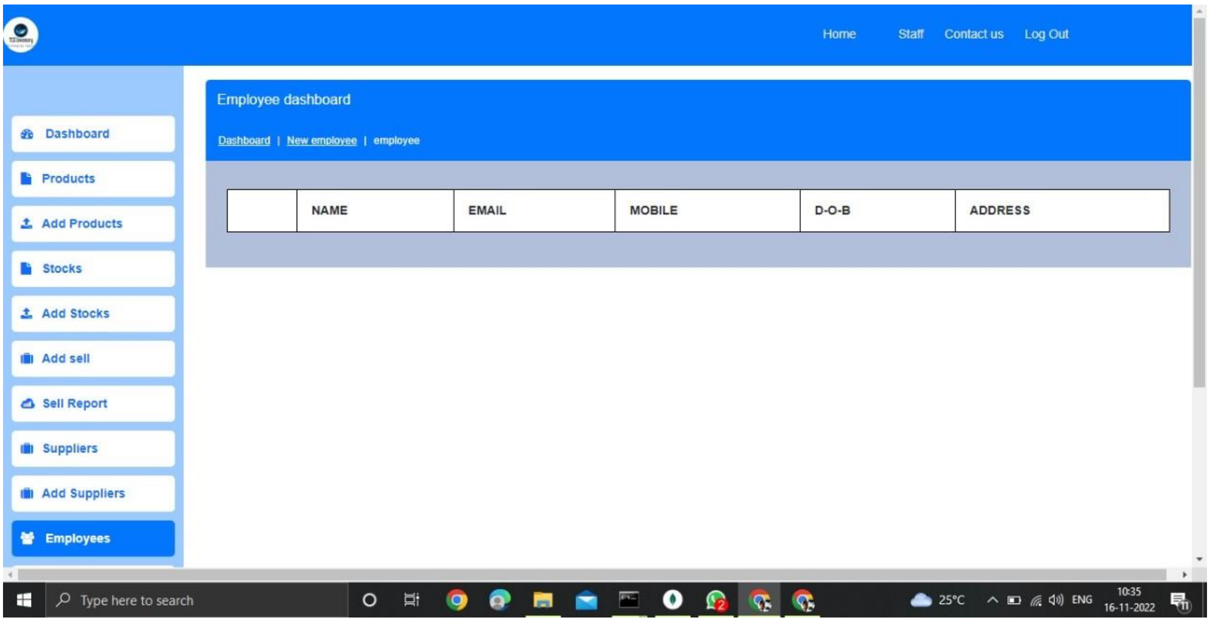
employee.component.html(Compressed Code):

```
<div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0"> <div>
<section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius clearfix"> <div
class="float-left"> <h3>Employee dashboard</h3> </div> <div class="float-right"> <div class="d-flex"> <nav
aria-label="breadcrumb"> <ol class="breadcrumb border-0"> <li class="breadcrumb-item"> <a
href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"> <a href="addemployee">New employee
</a> </li> <li class="breadcrumb-item active" aria-current="page"> employee </li> </ol> </nav> </div> </div>
</div> <div class="content-wrapper"> <div class="row mx-0 mt-2"> <div class="col-lg-12"> <form
class="theme-form"> <div class="form-colum"> <div class="container"> <div class="w3-container w3-
content"> <div> <table class="table table-responsive"> <tr> <th></th> <th>Name</th> <th>Email</th>
<th>Mobile</th> <th>D-O-B</th> <th>Address</th> </tr> <tr *ngFor="let items of employee"> <td
style="cursor: pointer" (click)="nextPage(items)"> &#9998; </td> <td>{{ items.name }}</td> <td>{{ items.email
}}</td> <td>{{ items.mobile }}</td> <td>{{ items.dob }}</td> <td>{{ items.address }}</td> </tr> </table> </div>
</div> </div> </div> </form> </div> </div> </div> </div> </section> <!-- //PAGE END -->
```

employee.component.ts(Compressed Code):

```
 import { Component, OnInit } from '@angular/core';import { ActivatedRoute, Router } from
'@angular/router';import { Employee } from 'src/app/models/employee.model';import { InventoryService }
from 'src/app/services/inventory.service';@Component({ selector: 'app-employee', templateUrl:
'./employee.component.html', styleUrls: ['./employee.component.css'],})export class EmployeeComponent
implements OnInit { public employee: Employee[] = []; constructor( private inventoryService: InventoryService,
private route: ActivatedRoute, private router: Router ) {} ngOnInit(): void { this.getEmployee(); } getEmployee()
{ this.inventoryService.getAll().subscribe( (data) => { this.employee = data; }, (err) => {} ); } nextPage(id: any) { if
(id._id) { id = id._id; } else { id = id.id; } this.router.navigate(['addemployee'], { queryParams: { id: id } }); }}
```

OUTPUT:



Employee dashboard

Dashboard | New employee | employee

	NAME	EMAIL	MOBILE	D-O-B	ADDRESS

ADD EMPLOYEE.COMPONENT

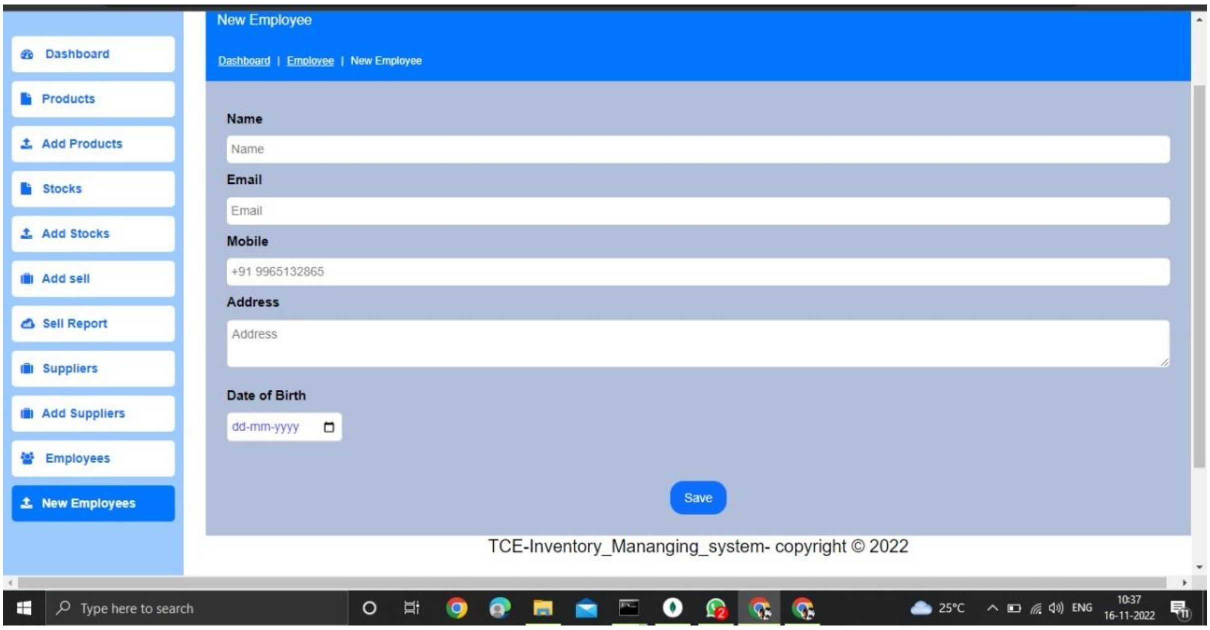
addemployee.component.html(Compressed Code):

```
div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0"> <!-- PAGE
START --> <div> <section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius
clearfix"> <div class="float-left"> <h3>New Employee</h3> </div> <div class="float-right"> <div class="d-flex">
<nav aria-label="breadcrumb"> <ol class="breadcrumb border-0"> <li class="breadcrumb-item"> <a
href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"><a href="employee">Employee</a></li>
<li class="breadcrumb-item active" aria-current="page"> New Employee </li> </ol> </nav> </div> </div>
</div> <div class="content-wrapper"> <div class="row mx-0 mt-2"> <div class="col-lg-12"> <form
class="theme-form"> <div class="form-row"> <div class="container"> <form action="action_page.php"> <div
class="row"> <div class="col-12"> <label for="fname">Name</label> </div> <div class="col-12"> <input
class="input-field" type="text" id="fname" name="name" placeholder="Name" [(ngModel)]="name" /> </div>
</div> <div class="col-12 mt-2"> <label for="fname">Email</label> </div> <div class="col-12"> <input
class="input-field" type="text" id="fname" name="email" placeholder="Email" [(ngModel)]="email" /> </div>
<div class="row mt-2"> <div class="col-12"> <label for="lname">Mobile</label> </div> <br /> <div class="col-
12"> <input class="input-field" type="text" id="lname" name="mobile" placeholder="+91 9965132865"
[(ngModel)]="mobile" /> </div> </div> <div class="row mt-2"> <div class="col-12"> <label
for="lname">Address</label> </div> <br /> <div class="col-12"> <textarea class="input-field" type="text"
id="address" name="address" placeholder="Address" [(ngModel)]="address" style="width:
100%;"></textarea> </div> </div> <br /> <div class="row"> <div class="col-12"> <label for="fname">Date of
Birth</label> </div> <div class="col-75"> <input class="input-field" type="date" id="fname" name="dob"
placeholder="05-01-1***" [(ngModel)]="dob" /> </div> </div> <br /> <div class=" text-center"> <button
class="btn btn-primary btn-block mt-4 " (click)="onSubmit()">Save</button> </div> </form> </div> </div>
</form> </div> </div> </div> </div> </section> <!-- //PAGE END -->
```

addemployee.componet.ts(Compressed Code):

```
import { Component, OnInit } from '@angular/core';import { InventoryService } from
'src/app/services/inventory.service';import { ActivatedRoute, Router } from '@angular/router';import {
ToastrService } from 'ngx-toastr';@Component({ selector: 'app-addemployee', templateUrl:
'./addemployee.component.html', styleUrls: ['./addemployee.component.css'],})export class
AddemployeeComponent implements OnInit { public name = ''; public mobile = ''; public dob = ''; public
address = ''; public email = ''; public id: any = null; constructor( private inventoryService: InventoryService,
private route: ActivatedRoute, private toastr: ToastrService, private router: Router ) {
this.route.queryParams.subscribe((params) => { if (params['id']) { this.id = params['id']; this.getDetails(); } }); }
ngOnInit(): void {} getDetails() { this.inventoryService.get(this.id).subscribe({ next: (data: any) => { this.name =
data.name; this.dob = data.dob; this.mobile = data.mobile; this.email = data.email; this.address =
data.address; }, error: (err) => {}, }); } onSubmit(): void { if (this.name && this.dob && this.mobile) { var data = {
name: this.name, dob: this.dob, mobile: this.mobile, email: this.email, address: this.address, }; if (this.id) {
this.inventoryService.update(this.id, data).subscribe({ next: (data) => { this.toastr.success('Employee Updated
Successfully'); this.navigate(); }, error: (err: any) => {}, }); } else { this.inventoryService.create(data).subscribe({
next: (data) => { this.toastr.success('Employee Created Successfully'); this.navigate(); }, error: (err: any) => {}, });
} } else { this.toastr.warning('Please fill all the details of the form'); } } navigate() {
this.router.navigate(['employee']); }}
```

OUTPUT:



New Employee

Dashboard | Employee | New Employee

Name

Name

Email

Email

Mobile

+91 9965132865

Address

Address

Date of Birth

dd-mm-yyyy

TCE-Inventory_Mananging_system- copyright © 2022

ADDSUPPLIERS.COMPONENT

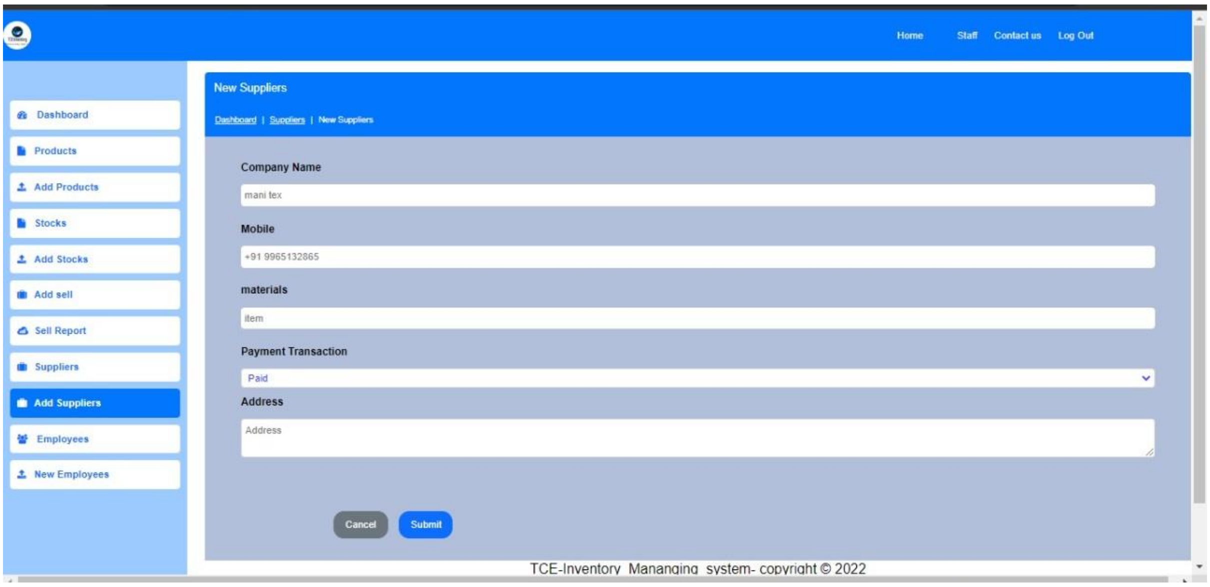
addsuppliers.component.html(Compressed Code):

```
<div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0"> <div>
<section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius clearfix"> <div
class="float-left"> <h3>New Suppliers</h3> </div> <div class="float-right"> <div class="d-flex"> <nav aria-
label="breadcrumb"> <ol class="breadcrumb border-0"> <li class="breadcrumb-item"> <a
href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"><a href="suppliers">Suppliers</a></li>
<li class="breadcrumb-item active" aria-current="page"> New Suppliers </li> </ol> </nav> </div> </div> </div>
<div class="content-wrapper"> <div class="row mx-0 mt-2"> <div class="col-lg-12"> <div class="form-row">
<div class="container"> <div class="row"> <div class="col-12"> <label for="fname">Company Name</label>
</div> <div class="col-12"> <input type="text" class="input-field" id="fname" name="name"
placeholder="mani tex" [(ngModel)]="name" /> </div> </div> <br /> <div class="row"> <div class="col-12">
<label for="lname">Mobile</label> </div> <br /> <div class="col-12"> <input type="text" class="input-field"
id="lname" name="mobile" placeholder="+91 9965132865" [(ngModel)]="mobile" /> </div> </div> <br /> <div
class="row"> <div class="col-12"> <label for="fname">materials</label> </div> <div class="col-12"> <input
type="text" class="input-field" id="materials" name="materials" placeholder="item" [(ngModel)]="materials"
/> </div> </div> <br> <div class="row"> <div class="col-12"> <label for="trans">Payment
Transaction</label><br> <select placeholder="Please Select Payment" id="payment" class="input-field"
style="width: 100%;" [(ngModel)]="payment"> <option value="Paid">Paid</option> <option
value="Pending">Pending</option> <option value="Advance">Advance</option> </select> </div> </div> <div
class="row mt-2"> <div class="col-12"> <label for="lname">Address</label> </div> <br /> <div class="col-12">
<textarea class="input-field" type="text" id="address" name="address" placeholder="Address"
[(ngModel)]="address" style="width: 100%;"></textarea> </div> </div> <!-- <div class="row"> <div class="col-
25"> <label for="fname">Receipt</label> </div> <div class="col-75"> <input type="file" id="fname"
name="img" placeholder="pdf" [(ngModel)]="receipt" /> </div> </div> --> <br /> <div class="form-row mt-5
text-center"> <div class="form-group col-md-4"> <button class="btn btn-secondary m-2" (click)="navigate()">
<i class="ti-close font-weight-bold mr-1"></i> Cancel </button> <button class="btn btn-primary m-2"
(click)="onSubmit()"> <i class="ti-check font-weight-bold mr-1"></i> Submit </button> </div> </div> </div>
</div> </div> </div> </div> </div> </section> <!-- //PAGE END -->
```

addsuppliers.component.ts(Compressed Code):

```
import { Component, OnInit } from '@angular/core';import { ActivatedRoute, Router } from
'@angular/router';import { ToastrService } from 'ngx-toastr';import { SuppliersService } from
'src/app/services/suppliers.service';@Component({ selector: 'app-addemployee', templateUrl:
'./addsuppliers.component.html', styleUrls: ['./addsuppliers.component.css'],})export class
AddsuppliersComponent implements OnInit { public name = ''; public mobile = ''; public address = ''; public
payment = 'Paid'; public materials = ''; public id: any = null; constructor( private toastr: ToastrService, private
suppliersService: SuppliersService, private route: ActivatedRoute, private router: Router ) {} ngOnInit(): void {
this.route.queryParams.subscribe((params) => { if (params['id']) { this.id = params['id']; this.getDetails(); } }); }
getDetails() { this.suppliersService.get(this.id).subscribe({ next: (data: any) => { this.name = data.name;
this.mobile = data.mobile; this.address = data.address; this.payment = data.payment; this.materials =
data.materials; }, error: (err) => {}, }); } onSubmit() { if ( this.name && this.materials && this.mobile &&
this.payment && this.address ) { var data = { name: this.name, materials: this.materials, mobile: this.mobile,
payment: this.payment, address: this.address, }; if (this.id) { this.suppliersService.update(this.id,
data).subscribe({ next: (data: any) => { this.toastr.success('Suppliers Updated Successfully'); this.navigate(); },
error: (err: any) => {}, }); } else { this.suppliersService.create(data).subscribe({ next: (data: any) => {
this.toastr.success('Suppliers Updated Successfully'); this.navigate(); }, error: (err: any) => {}, }); } } else {
this.toastr.warning('Please Fill all the fields of the form'); } } navigate() { this.router.navigate(['/suppliers']); }}
```

Output:



New Suppliers

Company Name: mani tex

Mobile: +91 9965132865

materials: item

Payment Transaction: Paid

Address: Address

TCE-Inventory Mananging system- copyright © 2022

ADDSTOCK.COMPONENT

addStock.component.html:(Compressed Code):

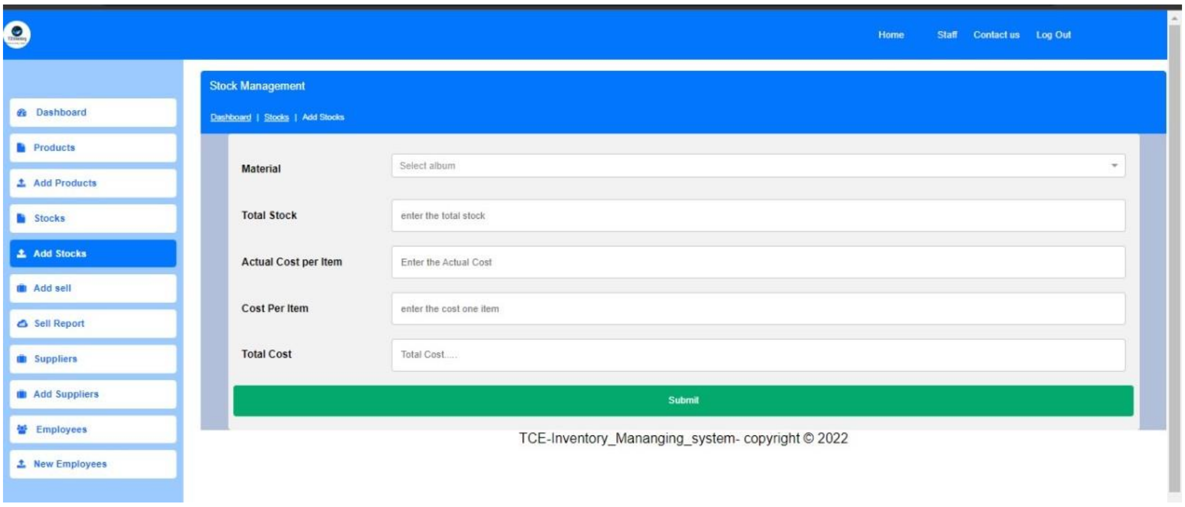
```
<div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0"> <div>
<section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius clearfix"> <div
class="float-left"> <h3>Stock Management</h3> </div> <div class="float-right"> <div class="d-flex"> <nav aria-
label="breadcrumb"> <ol class="breadcrumb border-0"> <li class="breadcrumb-item"> <a
href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"><a href="stock">Stocks</a></li> <li
class="breadcrumb-item active" aria-current="page"> Add Stocks </li> </ol> </nav> </div> </div> </div> <div
class="content-wrapper mt-0 p-0"> <div class="row "> <div class="col-lg-12"> <div class="form-row"> <div
class="container"> <div class="row mt-2"> <div class="col-2"> <label for="fname">Material</label> </div>
<div class="col-10"> <ng-select [items]="report" #select bindLabel="ptitle" dropdownPosition="auto"
bindValue="ptitle" labelForId="album" placeholder="Select album" [virtualScroll]="true"
[(ngModel)]="material" [clearable]="false" (change)="onAdd($event)"> <ng-template ng-option-tmp let-
item="item" let-search="searchTerm"> <div><span>Title: </span><span>{{item.ptitle}}</span></div>
<small><b>P-code:</b> {{item.pcode}}</small> <small><b>Brand:</b> {{item.pbrand}}</small> </ng-
template> </ng-select> </div> </div> <div class="row mt-2"> <div class="col-2"> <label for="fname">Total
Stock</label> </div> <div class="col-10"> <input type="number" id="fname" name="stock" placeholder="enter
the total stock" [(ngModel)]="tstock" /> </div> </div> <div class="row mt-2"> <div class="col-2"> <label
for="fname">Actual Cost per Item</label> </div> <div class="col-10"> <input type="number" id="fname"
name="costperitem" placeholder="Enter the Actual Cost" [(ngModel)]="ac" /> </div> </div> <div class="row
mt-2"> <div class="col-2"> <label for="fname">Cost Per Item </label> </div> <div class="col-10"> <input
type="number" id="fname" name="costperitem" placeholder="enter the cost one item" [(ngModel)]="cpi" />
</div> </div> <div class="row mt-2"> <div class="col-2"> <label for="fname">Total Cost </label> </div> <div
class="col-10"> <input type="number" id="fname" name="costperitem" placeholder="Total Cost....."
[(ngModel)]="tcost" /> </div> </div> <div class="row mt-2"> <input type="submit" value="Submit"
(click)="onSubmit()" /> </div> </div> </div> </div> </div> </div> </div> </section> <!-- //PAGE END -->
```

addStock.component.ts:(Compressed Code):

```
import { Component, OnInit } from '@angular/core';import { ActivatedRoute, Router } from
'@angular/router';import { StockService } from 'src/app/services/stock.service';import { ToastrService } from
'ngx-toastr';import { Report } from 'src/app/models/report.model';import { ReportService } from
'src/app/services/report.service';@Component({ selector: 'app-addStock', templateUrl:
'./addStock.component.html', styleUrls: ['./addStock.component.css'],})export class AddStockComponent
implements OnInit { public pcode = ''; public ptitle = ''; public tstock = ''; public cpi = ''; public tcost = ''; public ac
= ''; public id: any = null; public material: any = null; public report: Report[] = []; pid: any; constructor( private
service: StockService, private toastr: ToastrService, private reportService: ReportService, private route:
ActivatedRoute, private router: Router ) { this.getReport(); } ngOnInit(): void {
this.route.queryParams.subscribe((params: { [x: string]: any }) => { if (params['id']) { this.id = params['id'];
this.getDetails(); } }); } getReport() { this.reportService.getAll().subscribe({ next: (data: any) => { this.report =
data; }, error: (err: any) => {}, }); } getDetails() { this.service.get(this.id).subscribe({ next: (data: any) => {
this.pcode = data.pcode; this.ptitle = data.ptitle; this.tstock = data.tstock; this.ac = data.ac; }, error: (err: any)
=> {}, }); } onAdd(event: any) { console.log(event); if (event) { this.pcode = event.pcode; this.ptitle =
event.ptitle; this.pid = event._id; } } onSubmit(): void { if ( this.pcode && this.ptitle && this.tstock && this.pid
&& this.cpi && this.ac && this.tcost && this.material ) { var data = { pcode: this.pcode, ptitle: this.ptitle, tstock:
this.tstock, cpi: this.cpi, tcost: this.tcost, pid: this.pid, ac: this.ac, }; if (parseInt(this.cpi) <= parseInt(this.ac)) {
this.toastr.warning('please enter cost per item actual cost'); return; } if (this.id) { this.service.update(this.id,
data).subscribe({ next: (data: any) => { this.toastr.success('Stock Updated Successfully'); this.navigate(); },
error: (err: any) => {}, }); } else { this.service.create(data).subscribe({ next: (data: any) => {
this.toastr.success('Stock Updated Successfully'); this.navigate(); }, error: (err: any) => {}, }); } } else {
this.toastr.warning('Please fill all the details of the form'); } } navigate() { this.router.navigate(['stock']); }}
```

35

OUTPUT:



Stock Management

Dashboard | Stocks | Add Stocks

Material: Select album

Total Stock: enter the total stock

Actual Cost per Item: Enter the Actual Cost

Cost Per Item: enter the cost one item

Total Cost: Total Cost....

TCE-Inventory_Mananging_system- copyright © 2022

STOCKS.COMPONENT

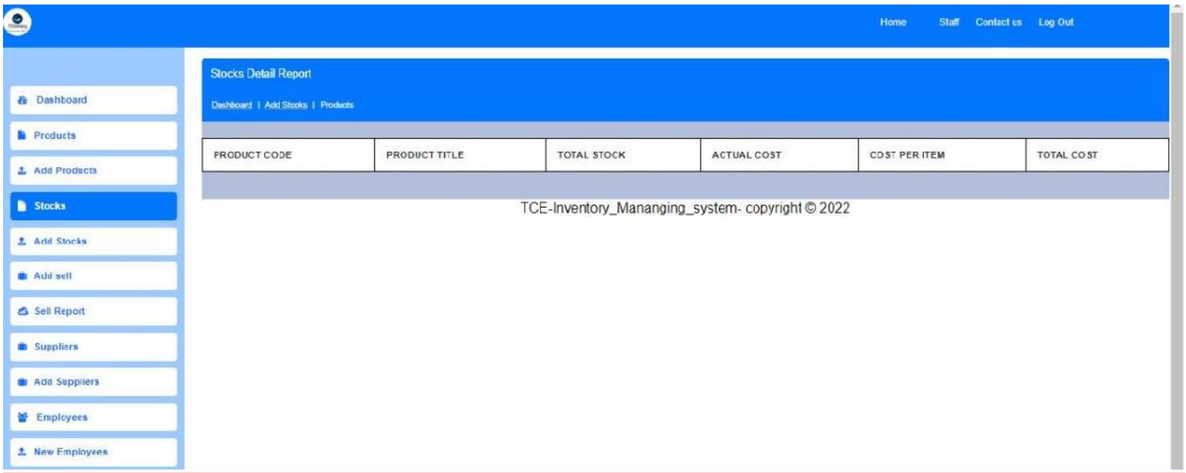
stock.component.html(Compressed Code):

```
<div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0"> <div>
<section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius clearfix"> <div
class="float-left"> <h3>Stocks Detail Report</h3> </div> <div class="float-right"> <div class="d-flex"> <nav
aria-label="breadcrumb"> <ol class="breadcrumb"> <li class="breadcrumb-item"> <a
href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"> <a href="addstock">Add Stocks</a>
</li> <li class="breadcrumb-item active" aria-current="report"> Products </li> </ol> </nav> </div> </div>
</div> <div class="row "> <div class="col-lg-12"> <div class="content-wrapper"> <!-- Table start --> <table
id="sort-table" class="table table-responsive"> <thead> <tr> <th>Product code</th> <th>Product Title</th>
<th>Total Stock</th> <th>Actual Cost</th> <th>Cost Per Item</th> <th>Total cost</th> </tr> <tr *ngFor="let
items of report"> <td>{{ items.pcode }}</td> <td>{{ items.ptitle }}</td> <td>{{ items.tstock }}</td> <td>{{
items.ac }}</td> <td>{{ items.cpi }}</td> <td>{{ items.tcost }}</td> </tr> </thead> </table> <!-- Table end -->
</div> </div> </div> </div> </section> <!-- //PAGE END -->
```

stock.component.ts(Compressed Code):

```
import { Component, OnInit } from '@angular/core';import { ActivatedRoute, Router } from
'@angular/router';import { Stock } from 'src/app/models/stock.model';import { StockService } from
'src/app/services/stock.service';@Component({ selector: 'app-stock', templateUrl: './stock.component.html',
styleUrls: ['./stock.component.css'],})export class StockComponent implements OnInit { public report: Stock[] =
[]; constructor( private service: StockService, private route: ActivatedRoute, private router: Router ) {}
ngOnInit(): void { this.getReport(); } getReport() { this.service.getAll().subscribe({ next: (data: any) => {
this.report = data; }, error: (err: any) => {}, }); } nextPage(id: any) { this.router.navigate(['addStock'], {
queryParams: { id: id } }); }}
```

OUTPUT:



Stocks Detail Report

PRODUCT CODE	PRODUCT TITLE	TOTAL STOCK	ACTUAL COST	COST PER ITEM	TOTAL COST

TCE-Inventory_Mananging_system- copyright © 2022

ADDREPORT.COMPONENT

addreport.component.html:(Compressed Code)

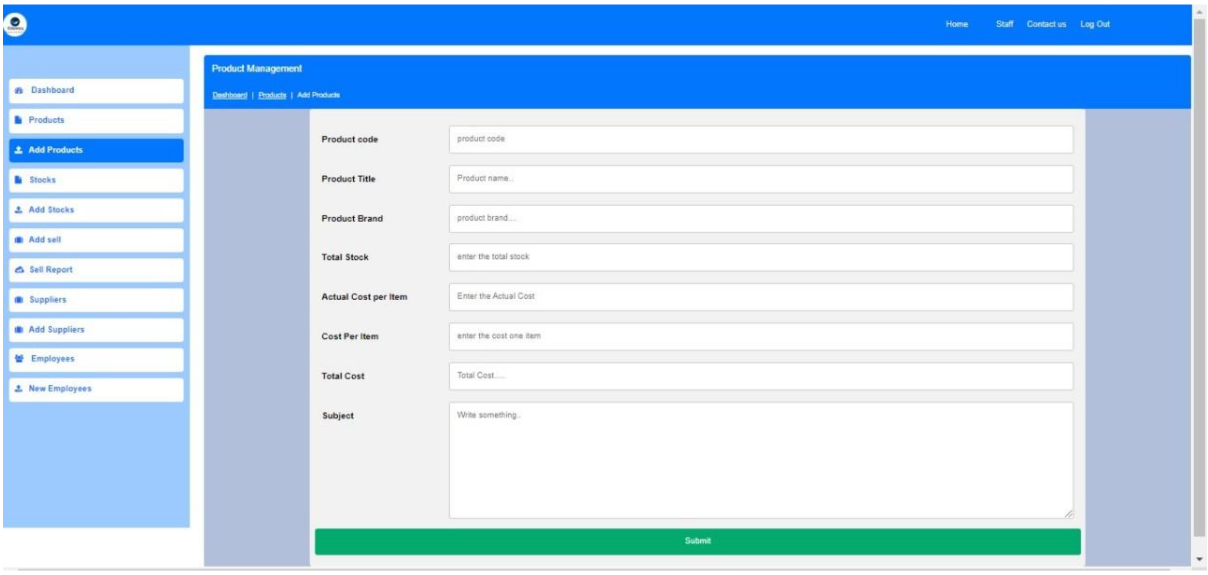
```
<div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0"> <div>
<section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius clearfix"> <div
class="float-left"> <h3>Product Management</h3> </div> <div class="float-right"> <div class="d-flex"> <nav
aria-label="breadcrumb"> <ol class="breadcrumb border-0"> <li class="breadcrumb-item"> <a
href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"><a href="report">Products</a></li> <li
class="breadcrumb-item active" aria-current="page"> Add Products </li> </ol> </nav> </div> </div> </div>
<div class="content-wrapper mt-0 p-0"> <div class="row "> <div class="col-lg-12"> <div class="form-row"> <div
class="container"> <div class="row mt-2"> <div class="col-2"> <label for="fname">Product code</label>
</div> <div class="col-10"> <input type="text" id="fname" name="Product code" placeholder="product code"
[(ngModel)]="pcode" /> </div> </div> <div class="row mt-2"> <div class="col-2"> <label for="lname">Product
Title</label> </div> <div class="col-10"> <input type="text" id="lname" name="Product name"
placeholder="Product name.." [(ngModel)]="ptitle" /> </div> </div> <div class="row mt-2"> <div class="col-2">
<label for="fname">Product Brand</label> </div> <div class="col-10"> <input type="text" id="fname"
name="Product brand" placeholder="product brand ... " [(ngModel)]="pbrand" /> </div> </div> <div
class="row mt-2"> <div class="col-2"> <label for="fname">Total Stock</label> </div> <div class="col-10">
<input type="number" id="fname" name="stock" placeholder="enter the total stock" [(ngModel)]="tstock" />
</div> </div> <div class="row mt-2"> <div class="col-2"> <label for="fname">Actual Cost per Item</label>
</div> <div class="col-10"> <input type="number" id="fname" name="costperitem" placeholder="Enter the
Actual Cost" [(ngModel)]="ac" /> </div> </div> <div class="row mt-2"> <div class="col-2"> <label
for="fname">Cost Per Item </label> </div> <div class="col-10"> <input type="number" id="fname"
name="costperitem" placeholder="enter the cost one item" [(ngModel)]="cpi" /> </div> </div> <div
class="row mt-2"> <div class="col-2"> <label for="fname">Total Cost </label> </div> <div class="col-10">
<input type="number" id="fname" name="costperitem" placeholder="Total Cost. ... " [(ngModel)]="tcost" />
</div> </div> <div class="row mt-2"> <div class="col-2"> <label for="subject">Subject</label> </div> <div
class="col-10"> <textarea id="subject" name="subject" placeholder="Write something.." style="height: 200px"
[(ngModel)]="sub"></textarea> </div> </div> <div class="row mt-2"> <input type="submit" value="Submit"
(click)="onSubmit()" /> </div> </div> </div> </div> </div> </div> </div> </section>
```

addreport.component.ts(Compressed Code):

```
import { Component, OnInit } from '@angular/core';import { ActivatedRoute, Router } from
'@angular/router';import { ReportService } from 'src/app/services/report.service';import { ToastrService } from
'ngx-toastr';@Component({ selector: 'app-addreport', templateUrl: './addreport.component.html', styleUrls:
['./addreport.component.css'],})export class AddreportComponent implements OnInit { public pcode = '';
public ptitle = ''; public pbrand = ''; public tstock = ''; public cpi = ''; public tcost = ''; public sub = ''; public ac = '';
public id: any = null; constructor( private reportService: ReportService, private toastr: ToastrService, private
route: ActivatedRoute, private router: Router ) {} ngOnInit(): void { this.route.queryParams.subscribe((params:
{ [x: string]: any }) => { if (params['id']) { this.id = params['id']; this.getDetails(); } }); } getDetails() {
this.reportService.get(this.id).subscribe({ next: (data: any) => { this.pcode = data.pcode; this.ptitle = data.ptitle;
this.pbrand = data.pbrand; this.tstock = data.tstock; this.cpi = data.cpi; this.tcost = data.tcost; this.ac = data.ac;
this.sub = data.sub; }, error: (err: any) => {}, }); } onSubmit(): void { if (this.pcode && this.ptitle && this.pbrand
&& this.tstock && this.sub) { var data = { pcode: this.pcode, ptitle: this.ptitle, pbrand: this.pbrand, tstock:
this.tstock, sub: this.sub, cpi: this.cpi, tcost: this.tcost, ac: this.ac, }; if (parseInt(this.cpi) <= parseInt(this.ac)) {
this.toastr.warning('please enter cost per item actual cost'); return; } if (this.id) {
this.reportService.update(this.id, data).subscribe({ next: (data: any) => { this.toastr.success('Product Updated
Successfully'); this.navigate(); }, error: (err: any) => {}, }); } else { this.reportService.create(data).subscribe({
```

```
next: (data: any) => { this.toastr.success('Product Updated Successfully'); this.navigate(); }, error: (err: any) =>
{}, }); } } else { this.toastr.warning('Please fill all the details of the form'); } } navigate() {
this.router.navigate(['report']); }}
```

OUTPUT:



Product code	product code
Product Title	Product name..
Product Brand	product brand....
Total Stock	enter the total stock
Actual Cost per Item	Enter the Actual Cost
Cost Per Item	enter the cost one item
Total Cost	Total Cost.....
Subject	Write something..

REPORT. COMPONENT

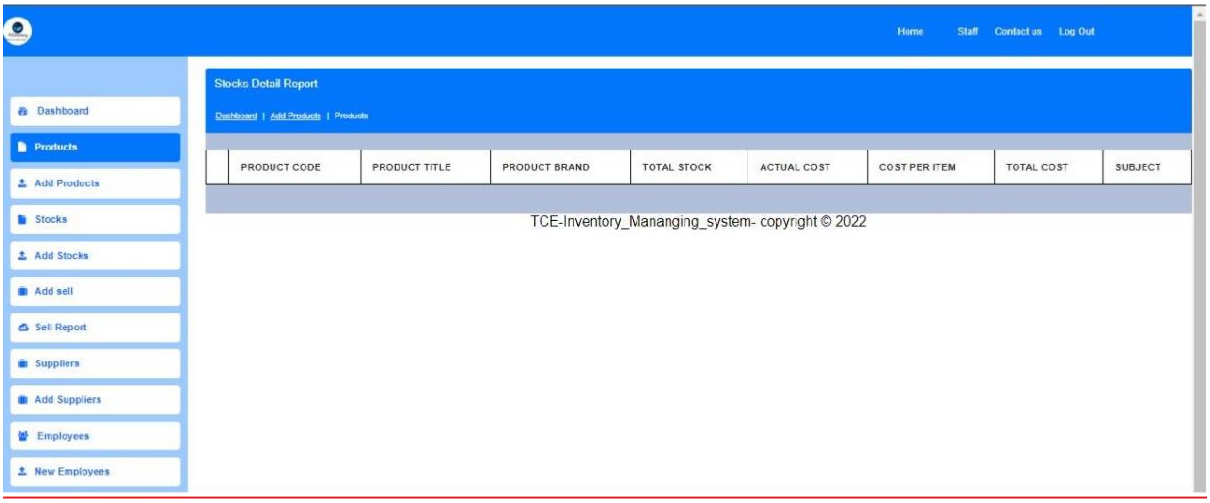
report.component.html(compressed code):

```
<div class="row"> <div class="col-2"> <app-navbar></app-navbar> </div> <div class="col-10 p-0">
<div> <section class="mt-3"> <div class="container-fluid"> <div class="page-header ph-radius
clearfix"> <div class="float-left"> <h3>Stocks Detail Report</h3> </div> <div class="float-right"> <div
class="d-flex"> <nav aria-label="breadcrumb"> <ol class="breadcrumb"> <li class="breadcrumb-
item"> <a href="dashboard">Dashboard</a> </li> <li class="breadcrumb-item"> <a
href="addreport">Add Products</a> </li> <li class="breadcrumb-item active" aria-current="report">
Products </li> </ol> </nav> </div> </div> </div> <div class="row "> <div class="col-lg-12"> <div
class="content-wrapper"> <!-- Table start --> <table id="sort-table" class="table table-responsive">
<thead> <tr> <th class="nosort" style="width: 18px"> </th> <th>Product code</th> <th>Product
Title</th> <th>Product Brand</th> <th>Total Stock</th> <th>Actual Cost</th> <th>Cost Per
Item</th> <th>Total cost</th> <th>SUBJECT</th> </tr> <tr *ngFor="let items of report"> <td
style="cursor: pointer" (click)="nextPage(items)"> &#9998; </td> <td>{{ items.pcode }}</td> <td>{{
items.ptitle }}</td> <td>{{ items.pbrand }}</td> <td>{{ items.tstock }}</td> <td>{{ items.ac }}</td>
<td>{{ items.cpi }}</td> <td>{{ items.tcost }}</td> <td>{{ items.sub }}</td> </tr> </thead> </table>
<!-- Table end --> </div> </div> </div> </div> </section> <!-- //PAGE END -->
```

report.component.ts(compressed code):

```
import { Component, OnInit } from '@angular/core';import { ActivatedRoute, Router } from
'@angular/router';import { Report } from 'src/app/models/report.model';import { ReportService }
from 'src/app/services/report.service';@Component({ selector: 'app-report', templateUrl:
'./report.component.html', styleUrls: ['./report.component.css'],})export class ReportComponent
implements OnInit { public report: Report[] = []; constructor( private reportService: ReportService,
private route: ActivatedRoute, private router: Router ) {} ngOnInit(): void { this.getReport(); }
getReport() { this.reportService.getAll().subscribe({ next: (data: any) => { this.report = data; }, error:
(err: any) => {}, }); } nextPage(id: any) { if (id._id) { id = id._id; } else { id = id.id; }
this.router.navigate(['addreport'], { queryParams: { id: id } }); }}
```

OUTPUT:



Stocks Detail Report

	PRODUCT CODE	PRODUCT TITLE	PRODUCT BRAND	TOTAL STOCK	ACTUAL COST	COST PER ITEM	TOTAL COST	SUBJECT

TCE-Inventory_Mananging_system- copyright © 2022

COMMON CSS FOR ALL (Compresseed code):

```
.outer { height: 100vh; background: linear-gradient(to top, #c9c9ff 50%, #6e6ef6 90%) no-repeat;}.panel-heading { text-
align: center; margin-bottom: 10px;}#forget { min-width: 100px; margin-left: auto; text-decoration: none; cursor:
pointer;}a:hover { text-decoration: none;}.form-inline label { padding-left: 10px; margin: 0%; cursor: pointer;}.btn { margin-
top: 20px; border-radius: 15px;}.panel { min-height: 380px; box-shadow: 20px 20px 80px rgb(218, 218, 218); border-radius:
12px; margin-top: 100px;}.input-field { border-radius: 5px; padding: 5px; display: flex; align-items: center; cursor: pointer;
border: 1px solid #ddd; color: #4343ff;}.input-field.error { border: 1px solid red; color:
red;}input[type="text"],input[type="email"],input[type="password"] { border: none; outline: none; box-shadow: none;
width: 100%;}.fa-eye-slash.btn { border: none; outline: none; box-shadow: none;}img { width: 40px; height: 40px; object-fit:
cover; border-radius: 50%; position: relative;}.bordert { border-top: 1px solid #aaa; position: relative;}.bordert::after {
content: "or connect with"; position: absolute; top: -13px; left: 40%; background-color: #fff; padding: 0 8 px;}/*--font-
family: 'Montserrat', sans-serif;--*/@import
url("https://fonts.googleapis.com/css2?family=Nunito:wght@200;300;400;600;700;800;900&display=swap");/* @import
url('https://fonts.googleapis.com/css2?family=Montserrat:wght@100;200;300;400;500;600;700;800;900&display=swap');
*//*=====================================//--//--> STYLES
GENERALES======================================*/body { padding: 0; margin: 0; font-family: "Nunito", sans-serif;
font-size: 14px; background: #fff;}h1,h2,h3,h4,h5,h6 { color: #000000;}a { color: #0276fd; transition: all 0.5s ease;}a:hover {
color: #006be7; text-decoration: none;}.form-control::placeholder { /* Chrome, Firefox, Opera, Safari 10.1+ */ color:
#8dacd1; opacity: 1; /* Firefox */ font-size: 14px !important;}.form-control:-ms-input-placeholder { /* Internet Explorer 10-
11 */ color: #8dacd1; font-size: 14px !important;}.form-control::-ms-input-placeholder { /* Microsoft Edge */ color:
#8dacd1; font-size: 14px !important;}.form-control select { color: #8dacd1; font-size: 14px !important;}.theme-form select {
color: #8dacd1 !important; font-size: 14px !important;}.theme-form select option { color: #8dacd1 !important; font-size:
14px !important;}.theme-form select option:first-child { color: #8dacd1 !important; font-size: 14px !important;}.btn {
padding: 0.375rem 0.95rem; font-size: 14px !important; min-height: 38px; font-weight: 500;}.btn-sm { padding: 0.25rem
0.75rem; font-size: 13px !important; min-height: 28px; font-weight: 500;}.lftBdr-radius { border-radius: 5px 0 0
5px;}.rgtBdr-radius { border-radius: 0 5px 5px 0;}/*--Text Color--*/.blue-text { color: #006be7 !important;}.green-text {
color: #7cb83b !important;}.red-text { color: #eb5463 !important;}.purple-text { color: #8d44ad !important;}.orange-text {
color: #fa9151 !important;}.aqua-text { color: #45c8e9 !important;}/*--BG Color--*/.blue-bg { background-color: #006be7
!important;}.green-bg { background-color: #7cb83b !important;}.red-bg { background-color: #eb5463 !important;}.purple-
bg { background-color: #8d44ad !important;}.orange-bg { background-color: #fa9151 !important;}.aqua-bg { background-
color: #45c8e9 !important;}.bg-info-light { background-color: #e4f6f9 !important;}.bg-orange-light { background-color:
#fff1e8 !important;}.bg-red-light { background-color: #ffe9eb !important;}/* --- *//*--Login Page--*/.login-page {
background: #000; padding: 75px 0px;}.login-page .card { border: 0;}.login-page .card-header { text-align: center;
background: #333; padding: 15px 0;}.login-page .form-group { margin-bottom: 1.5rem;}.login-page .form-control { font-
size: 14px; padding: 1.25rem 0.75rem;}/*----*//*--Admin Page--*//*--HEADER--*/#my-header .bg-dark { background-color:
#0276fd !important;}#my-header .navbar-brand { margin-right: 15px; width: 235px;}#my-header .navbar { padding:
0px;}#my-header .nav-item { padding: 15px 12px;}#my-header .nav-item .nav-link { color: #eee !important; padding: 10px
0px;}#my-header .nav-item .nav-link:hover { color: #0276fd !important;}.nav-user { padding: calc(31px / 2) 20px calc(31px /
2) 57px !important; text-align: left !important; position: relative; background-color: rgb(18, 119, 202); border-width: 0 1px;
min-width: 180px;}.nav-user .account-user-avatar { position: absolute; top: calc(34px / 2); left: 15px;}.nav-user .account-
user-avatar img { height: 36px; width: 36px;}.nav-user .account-user-name { display: block; font-weight: 500; color: #fff;
font-size: 14px !important;}.nav-user .account-position { display: block; font-size: 11px; margin-top: 0px; color:
#fff;}.dropdown-menu-right { min-width: 180px;}.dropdown-menu-right .dropdown-item { font-size: 14px; padding:
0.35rem 0.75rem; font-weight: 500; color: #394449;}.dropdown-toggle::after { display: none;}.dropdown-menu-animated {
-webkit-animation-name: DropDownSlide; animation-name: DropDownSlide; -webkit-animation-duration: 0.3s; animation-
duration: 0.3s; -webkit-animation-fill-mode: both; animation-fill-mode: both; margin: 0; position: absolute; z-index:
1000;}.dropdown-menu-animated.show { top: 100% !important;}.dropdown-menu-animated i { display: inline-block;
margin-right: 3px;}.dropdown-menu-animated.dropdown-menu-right { right: 0 !important; left: auto
!important;}.dropdown-menu-animated.dropdown-menu[x-placement^="left"],.dropdown-menu-animated.dropdown-
menu[x-placement^="right"],.dropdown-menu-animated.dropdown-menu[x-placement^="top"] { top: auto !important; -
webkit-animation: none !important; animation: none !important;}@-webkit-keyframes DropDownSlide { 100% { -webkit-
transform: translateY(0); transform: translateY(0); } 0% { -webkit-transform: translateY(20px); transform: translateY(20px);
}}@keyframes DropDownSlide { 100% { -webkit-transform: translateY(0); transform: translateY(0); } 0% { -webkit-
transform: translateY(20px); transform: translateY(20px); }}/*--MAIN-MENU--*/.main-menu { background: #111; padding:
10px; border-bottom: 1px solid #222;}.main-menu .nav-item { margin: 8px;}.main-menu .nav-link { border: 1px solid
#007bff; color: #007bff; text-transform: uppercase; font-size: 13px; background: #ffffff; font-weight: 600;}.main-menu
```

```
.nav-link:hover { color: #fff; background-color: #007bff;}.main-menu .nav-link i { font-size: 1.15rem; display: block; margin:
0 auto 5px auto; text-align: center;}.green-link { border: 1px solid #7cb83b !important; color: #7cb83b !important;}.green-
link.active { color: #fff !important; background-color: #7cb83b !important;}.green-link:hover { color: #fff !important;
background-color: #7cb83b !important;}.red-link { border: 1px solid #eb5463 !important; color: #eb5463 !important;}.red-
link.active { color: #fff !important; background-color: #eb5463 !important;}.red-link:hover { color: #fff !important;
background-color: #eb5463 !important;}.purple-link { border: 1px solid #8d44ad !important; color: #8d44ad
!important;}.purple-link.active { color: #fff !important; background-color: #8d44ad !important;}.purple-link:hover { color:
#fff !important; background-color: #8d44ad !important;}.mint-link { border: 1px solid #46ceac !important; color: #46ceac
!important;}.mint-link.active { color: #fff !important; background-color: #46ceac !important;}.mint-link:hover { color: #fff
!important; background-color: #46ceac !important;}.orange-link { border: 1px solid #fa9151 !important; color: #fa9151
!important;}.orange-link.active { color: #fff !important; background-color: #fa9151 !important;}.orange-link:hover { color:
#fff !important; background-color: #fa9151 !important;}.aqua-link { border: 1px solid #45c8e9 !important; color: #45c8e9
!important;}.aqua-link.active { color: #fff !important; background-color: #45c8e9 !important;}.aqua-link:hover { color: #fff
!important; background-color: #45c8e9 !important;}.yallow-link { border: 1px solid #c29c18 !important; color: #c29c18
!important;}.yallow-link.active { color: #fff !important; background-color: #c29c18 !important;}.yallow-link:hover { color:
#fff !important; background-color: #c29c18 !important;}.pink-link { border: 1px solid #f5799c !important; color: #f5799c
!important;}.pink-link.active { color: #fff !important; background-color: #f5799c !important;}.pink-link:hover { color: #fff
!important; background-color: #f5799c !important;}/*--MENU-PAGE--*//*--Sub Menu--*/.sub-menu .nav-item { margin:
0px 1px;}.sub-menu .nav-item .nav-link { background: #fff; border-radius: 5px 5px 0px 0px; font-weight: 500; padding:
0.65rem 1rem;}.sub-menu .nav-item .nav-link:hover { background: #0276fd; color: #fff;}.sub-menu .nav-item .active {
background: #0276fd; color: #fff;}.page-header { background: #0276fd; margin: 0 1px; padding: 0; border-radius: 0px 5px
0px 0px;}.ph-radius { border-radius: 5px 5px 0px 0px;}.page-header h3 { margin: 0; color: #fff; font-size: 1rem; padding:
13px;}.page-header .breadcrumb { margin-bottom: 0; list-style: none; background-color: transparent; border-radius: 0;
font-size: 12px; padding: 0.9rem;}.page-header .breadcrumb .breadcrumb-item a { color: #fff;}.page-header .breadcrumb
.breadcrumb-item.active { color: #fff;}.page-header .breadcrumb-item + .breadcrumb-item::before { display: inline-block;
padding-right: 0.5rem; color: #fff; content: "|";}.page-action { display: -ms-flexbox; display: flex; -ms-flex-wrap: wrap; flex-
wrap: wrap; margin-bottom: 0; list-style: none; padding-left: 0;}.page-action li { border-right: 1px solid #2d92ff; border-left:
1px solid #0166d3; padding: 0.7rem 0.9rem;}.page-action li:last-child { border-right: 0; border-radius: 0px 5px 0px
0px;}.page-action li:hover { background: #0073ef;}.page-action li a i { vertical-align: middle; color: #fff; font-size: 20px;}/*--
WRAPPER--*/.content-wrapper { padding: 1.5rem 0 1.5rem 0; background: #b2bfdb; /* height: calc(100vh - 230px); */
overflow: hidden; margin: 0 1px;}.table-striped tbody tr:nth-of-type(odd) { background-color: #fcfdff;}.table-striped thead {
background: #dee5f1;}.table-striped > tbody td { border: 1px solid #dee5f1;}.table-striped thead th { border-bottom: 2px
solid #dee5f1;}.table-container { margin-bottom: 1rem; margin-top: 1rem; padding: 0; /* border: 2px solid #dee5f1; */ -
webkit-border-radius: 3px; -moz-border-radius: 3px; border-radius: 0;}.custom-table { border: 1px solid #dee5f1;
background: #fff;}.custom-table thead { background: #dee5f1;}.custom-table thead th { border: 0; white-space: nowrap;
font-size: 13px; color: #000;}/* .custom-table > tbody tr:hover { background: #eef1f8; } */.custom-table > tbody tr:nth-of-
type(even) { background-color: #fcfdff;}.custom-table > tbody td { border: 1px solid #dee5f1; white-space: nowrap;}.table
th { padding: 0.8rem 1rem; font-weight: 600; text-transform: uppercase; border: 1px solid #000; background: #fff;}.table td
{ border-top: 1px solid #dee5f1; vertical-align: middle; padding: 0.65rem 1rem;}.td-notes { max-width: 185px; overflow:
hidden; text-overflow: ellipsis; white-space: nowrap;}#sort-table_length { padding-left: 15px; padding-top: 15px; padding-
bottom: 15px; display: inline-block; width: 50%;}#sort-table_paginate { padding-right: 15px; padding-top: 15px; padding-
bottom: 15px; display: inline-block; width: 50%;}.dataTables_info { padding-left: 15px; font-weight:
500;}.dataTables_paginate { padding-right: 15px;}.theme-form label { margin-bottom: 0.5rem !important; font-size: 15px
!important; font-weight: bolder !important; color: #000;}.theme-form .form-control { background: #fcfdff; padding:
0.375rem 0.75rem; height: calc(1.95em + 0.75rem + 2px); border-color: #dbe4f1; font-size: 14px !important;}.theme-form
.form-group { margin-bottom: 1.75rem;}/*--Theme Modal--*/.modal-header { background: #0276fd;}.modal-title { color:
#fff;}.modal-content { border: 0;}.close { float: right; font-size: 1.85rem; font-weight: 500; line-height: 1; color: #fff; text-
shadow: none; opacity: 0.75;}/*--Dashboard Card--*/.myCard { padding: 15px; border-radius: 5px; -webkit-box-shadow: 0
1px 8px 0 rgba(0, 0, 0, 0.12); box-shadow: 0 1px 8px 0 rgba(0, 0, 0, 0.12); border: none; text-align: right; margin-top: 15px;
margin-bottom: 30px;}.myCard .card-icon { width: 64px; height: 64px; position: absolute; top: -15px; font-size: 32px;
border-radius: 8px; display: -webkit-box; display: -ms-flexbox; display: flex; color: #fff; -webkit-box-align: center; -ms-flex-
align: center; align-items: center; -webkit-box-pack: center; -ms-flex-pack: center; justify-content: center; -webkit-
transition: all 0.3s ease-in-out; transition: all 0.3s ease-in-out; background: #0276fd;}.myCard :hover .card-icon { top: -
25px;}.myCard .card-head { display: block; font-size: 16px; text-transform: uppercase; font-weight: 600; color:
#007bff;}.myCard h4 { margin: 10px 0; font-weight: 700; color: #455055;}/*--Invoice Card Section Start--*/.invoice-card {
padding: 20px; border: 1px solid #dbe4f1; border-radius: 5px; background: #fcfdff;}.invoice-card h5 { text-align: right; font-
size: 14px; line-height: 22px; color: #455055;}.invoice-card hr { border-top: 1px solid #dbe4f1;}.invoice-card .table th {
padding: 0.5rem 1rem; font-weight: 600; text-transform: uppercase; border-bottom: 2px solid #dee5f1;}.invoice-card .table
```

```
td { border-top: 1px solid #dee5f1; vertical-align: middle; padding: 0.35rem 1rem;}.invoice-card .form-control-sm { height:
calc(1.05em + 0.5rem + 2px); padding: 0.25rem 0.5rem; font-size: 0.875rem; line-height: 1.5; border-radius:
0.2rem;}.invoice-info { background: #e2f4ff; padding: 15px;}.invoice-info h6 { color: #455055; font-size: 13px; font-weight:
700;}.invoice-info address { margin-bottom: 0px; line-height: 24px;}.invoice-info p { margin-top: 0; margin-bottom:
0.2rem;}.invoice-info .badge { padding: 0.35em 0.45em 0.15em;}/*--//Invoice Card Section End--*/.user-info { text-align:
center; border: 1px solid #dbe4f1; border-radius: 5px; background: #fcfdff; padding: 15px;}.user-info h4 { font-size: 18px;
font-weight: 700; margin-bottom: 2px;}.user-info h6 { font-size: 14px; color: #455055;}.user-info .user-thumb img { width:
130px; height: 130px; border-radius: 50%; -o-object-fit: cover; object-fit: cover; -o-object-position: center right; object-
position: center right; margin: 0px auto 5px; padding: 0.35rem; background-color: #fff; border: 1px solid #dbe4f1;}.user-
info .nav { text-align: left; margin-top: 20px;}.user-info .nav .nav-item { color: #455055; border-top: 1px solid
#dbe4f1;}.user-info .nav .nav-item a { color: #96a2a7; padding: 0.75rem 1rem;}.user-info .nav .nav-item a span { margin-
left: 5px;}.user-info .nav .nav-item a:hover { color: #455055;}.user-info .nav .nav-item .active { color: #455055;}.user-info
.nav .nav-item:last-child a { padding-bottom: 0px;}.user-details h2 { font-size: 22px; font-weight: 500; padding-bottom: 7px;
border-bottom: 1px solid #dbe4f1; color: #455055;}/*--FOOTER--*/.footer { background: #0276fd; padding: 0.9rem 1.3rem;
font-size: 0.75rem; color: #ffffff; margin: 0 0 1.25rem 0; -webkit-border-radius: 0 0 10px 10px; -moz-border-radius: 0 0
10px 10px; border-radius: 0 0 10px 10px; text-align: center;}.theme-form .custom-control label.mb-0 { margin-bottom: 0
!important;}.datepicker-input { position: relative;}.datepicker-input .form-control { padding-right: 25px;}.datepicker-input
.input-group-addon { position: absolute; right: 15px; top: 9px; font-size: 20px; color: #9bb1d9;}.select2-container--default
.select2-selection--multiple { background-color: white; border: 1px solid #aaa; border-radius: 4px; cursor: text; padding-
bottom: 5px; padding-right: 5px; display: block; width: 100%; height: calc(1.5em + 0.75rem + 2px); padding: 0.375rem
0.75rem; font-size: 1rem; font-weight: 400; line-height: 1.5; color: #495057; background-color: #fff; background-clip:
padding-box; border: 1px solid #ced4da; border-radius: 0.25rem; transition: border-color 0.15s ease-in-out, box-shadow
0.15s ease-in-out; background: #fcfdff; padding: 4px 5px; height: auto; min-height: 41.3px; border-color: #dbe4f1;}.select2-
container { width: 100% !important;}.select2-container--default.select2-container--focus .select2-selection--multiple {
border: solid #dbe4f1 1px; width: 100%; float: left; padding: 4px 5px;}.select2-container--default .select2-selection--
multiple .select2-selection_choice { background-color: #e4e4e4; border: 1px solid #dbe4f1; padding: 0; font-size:
12px;}.select2-container--default .select2-selection--multiple .select2-selection_choice_remove { border-right: 1px solid
#cecece; color: #7c95c5; cursor: pointer; font-size: 13px; padding: 0 4px;}.select2-container--default .select2-selection--
multiple .select2-selection_choice_display { color: #7c95c5;}.add-ticket-modal .close,.add-ticket-modal
.close:hover,.add-ticket-modal .close:active,.add-ticket-modal .close:focus { margin-right: 15px; margin-top: 7px; color: #fff;
border: none; outline: none;}.input-with-add { display: flex; align-items: center;}/* .input-with-add .form-control {}
*/.input-with-add .add-btn { width: 45px; height: 45px; margin-left: 10px; display: flex; align-items: center; justify-content:
center; border: solid 1px #dbe4f1; border-radius: 4px; float: left; color: #0978fa; cursor: pointer;}.show-labor-code {
display: none;}.popup-scroll { height: 315px; overflow-y: scroll !important; display: block;}.popup-scroll-450 { height: 450px;
overflow-y: scroll !important; display: block;}.popup-scroll-550 { height: 550px; overflow-y: scroll !important; display:
block;}/*--Job ID Time Line--*/.timeline { list-style-type: none; margin: 0; padding: 0; position: relative;}.timeline:before {
content: ""; position: absolute; top: 10px; bottom: 5px; width: 2px; background: #00489b; left: 5%; margin-left: 0;}.timeline
> li { position: relative; min-height: 50px; padding: 10px 0;}.timeline .timeline-time { position: absolute; left: 0; width: 10%;
text-align: right; top: 40px;}.timeline .timeline-time .date,.timeline .timeline-time .time { display: block; font-weight:
600;}.timeline .timeline-time .date { line-height: 16px; font-size: 13px; color: #00489b;}.timeline .timeline-time .time { line-
height: 24px; font-size: 22px; color: #455055;}.timeline .timeline-icon { left: 0%; position: absolute; width: 10%; text-align:
center; top: 20px;}.timeline .timeline-icon a { text-decoration: none; width: 38px; height: 38px; display: inline-block;
border-radius: 40px; background: #00489b; line-height: 10px; color: #fff; padding: 5px 0; font-size: 16px; line-height: 28px;
/* border: 2px solid #aaa; */ transition: border-color 0.2s linear; font-weight: 600;}.timeline .timeline-body { margin-left:
9%; margin-right: 3%; background: #fcfdff; position: relative; padding: 15px; border-radius: 6px;}.timeline .timeline-
body:before { content: ""; display: block; position: absolute; border: 10px solid transparent; border-right-color: #fcfcfc; left:
-20px; top: 20px;}/* .timeline .timeline-body>div+div { *//* margin-top: 15px *//* } */.timeline .timeline-body > div +
div:last-child { /* margin-bottom: -20px; */ padding-bottom: 0px; border-radius: 0 0 6px 6px;}.timeline-header { line-
height: 30px;}.timeline-header .job-task { font-size: 16px; font-weight: 600; line-height: 14px;}.timeline-header .job-
task,.timeline-header .job-task a { color: #0276fd;}.timeline-content { padding-top: 10px; border-top: 1px solid #e2e7eb;
letter-spacing: 0.25px; line-height: 18px; font-size: 13px;}.timeline-content:after,.timeline-content:before { content: "";
display: table; clear: both;}.timeline-footer { background: #fff; border-top: 1px solid #e2e7ec; padding: 15px 15px
0;}.timeline-content .form-control { font-size: 12px !important;}.timeline-notes { background: #f2f3f4; margin-left: -25px;
margin-right: -25px; padding: 20px 25px;}.job-card { border: 1px solid #dee5f1; background: #ffffff; margin-top: 10px;}.job-
card .card-header { padding: 0.5rem 1.25rem; margin-bottom: 0; background-color: #f4f6fc; border: none; box-shadow:
none; font-weight: 600;}.job-card .list-item { border-bottom: 1px solid #dee5f1; margin-top: 10px;}.job-status .form-control
{ font-size: 11px; height: calc(1.2em + 0.5rem + 2px); padding: 0.15rem 0.15rem; background: transparent; font-weight:
700; width: 110px;}.job-status .btn { font-size: 11px !important; font-weight: 600; padding: 5px 10px !important; line-
height: 11px; min-height: 22px; margin: 0px !important; vertical-align: top;}/*--//Job ID Time Line--*//* Table sticky header
*/.table-sticky-header .table { position: relative; width: 100%; float: left;}.table-sticky-header .table thead tr th { position:
sticky; top: 0; z-index: 2; background: #dee5f1;}.labour-header-info { margin-bottom: 0; list-style: none; background-color:
transparent; border-radius: 0; font-size: 13px; color: #fff; padding: 0.9rem;}.table-container.no-pagination .bottom {
display: none;}.stockin-stockout-container { display: flex;}.page-loading { position: absolute; width: 100%; z-index: 999;
background: rgb(255 255 255 / 50%); height: 100%; margin: 0 auto; text-align: center; padding-top: 15%;}.page-loading h4 {
font-weight: 600; font-size: 24px; color: #000000;}/*--Data Table--*/table.dataTable thead .sorting:before,table.dataTable
thead .sorting:after,table.dataTable thead .sorting_asc:before,table.dataTable thead .sorting_asc:after,table.dataTable
thead .sorting_desc:before,table.dataTable thead .sorting_desc:after,table.dataTable thead
.sorting_asc_disabled:before,table.dataTable thead .sorting_asc_disabled:after,table.dataTable thead
.sorting_desc_disabled:before,table.dataTable thead .sorting_desc_disabled:after { position: absolute; bottom: 0.9em;
display: block; opacity: 1;}/*--------------------------------------------------SIDEBAR STYLE----------------------------------------------------
*/.wrapper { display: flex; width: 100%; align-items: stretch;}#sidebar { min-width: 100%; max-width: 100%; background:
#0277fd64; color: #fff; transition: all 0.3s; padding-top: 10px;}#sidebar.active { margin-left: -15%;}#sidebar ul li { text-align:
left; display: block; float: left; margin: 5px 10px; width: 100%;}#sidebar ul li a { padding: 10px; font-size: 1em; display: block;
background-color: #fff; border-radius: 5px; font-weight: 600;}#sidebar ul li a:hover { background-color: #0276fd; color:
#fff;}#sidebar ul li a.active { background-color: #0276fd; color: #fff;}#sidebar ul li a i { width: 24px;}#content { width: 85%;
min-height: 100vh; transition: all 0.3s;}#content.active { width: 100%;}@media (max-width: 768px) { #my-header .navbar-
brand { margin-left: 15px; width: 190px; } #sidebar { min-width: 20%; max-width: 20%; margin-left: -20%; } #sidebar.active {
margin-left: 0; } #sidebarCollapse span { display: none; } #content { width: 100%; } #sidebar ul li a { padding: 10px; font-size:
1em; display: block; background-color: #fff; border-radius: 5px; font-weight: 600; } #sidebar ul li a i { width: 16px; }}/*--
INDEX NEW DASHBOARD--*/.ch-235 { min-height: 235px;}.ch-250 { min-height: 250px;}.ch-275 { min-height: 275px;}.ch-300
{ min-height: 300px;}.index-card { background: #ffffff; border: 1px solid #dadfe5; box-shadow: none; -webkit-border-radius:
2px; -moz-border-radius: 2px; border-radius: 7px; margin-bottom: 1rem;}.index-card .card-header { background: #fff;
border-bottom: 1px solid #dadfe5; padding: 0.75rem 1.05rem 0.75rem 1.05rem; border-radius: 7px 7px 0 0;}.index-card
.card-header .card-title { margin: 0; font-size: 1.25rem; font-weight: 600; line-height: 100%; color: #222222;}.index-card
.card-body { padding: 1.25rem 1.25rem; position: relative;}.ticket-box { text-align: center;}.ticket-box h5 { font-weight: 600;
color: #aaaaaa; margin-bottom: 0; font-size: 1rem;}.ticket-box .chart { width: 150px; height: 150px; margin: 0
auto;}.finance-box h2 { font-weight: 700; margin-bottom: 20px; font-size: 2.05rem;}.finance-box h6 { font-weight: 600;
color: #606569;}.progress { height: 0.75rem !important; margin-bottom: 5px;}.dot-label { width: 10px; height: 10px;
display: inline-block; border-radius: 100%; margin-right: 5px;}/*--Customers--*/.customers-box .cbt-card h5 { background:
transparent;}.customers-box h5 { font-weight: 600; color: #ffffff; font-size: 1rem; padding: 5px 10px; width: 85%; margin-
bottom: 15px; background: #333; border-radius: 8px;}.customers-box h2 { font-weight: 600; font-size: 1.15rem; color:
#333; margin-bottom: 15px;}.customers-box h6 { font-weight: 600; font-size: 0.9rem; color: #888; margin-bottom:
5px;}.customers-box .cbt-card i { float: right; vertical-align: middle; margin-top: 5px;}/*--Billing--*/.billing-box h2 { font-
weight: 700; font-size: 1.75rem; color: #666; margin-bottom: 5px;}.billing-box h6 { font-weight: 600; color: #aaaaaa;
margin-bottom: 7px; font-size: 1rem; margin-top: 10px;}.billing-box span { font-weight: 600; color: #aaaaaa;}.billing-box
.bill-icon { display: inline-flex; width: 80px; height: 80px; text-align: center; border-radius: 50%; align-items: center; justify-
content: center; background: #eee;}.billing-box .bill-icon i { font-size: 1.5rem;}/*-- //INDEX NEW DASHBOARD--*/.action a {
display: none;}.action:hover a { display: inline-block;}.print-des { padding: 0px 15px 15px 15px; border: 1px solid #e3e3e3;
border-radius: 5px; box-shadow: 4px 4px 10px #e9e9e9;}label { font-weight: bold; font-size: 15px; margin-bottom: 0.8rem;
color: #000;}
```

DATABASE SCHEMA

MODELS.JS

customer.model.js(Compressed code)

```
const mongoose = require("mongoose");const Customer = mongoose.model( "customer", new
mongoose.Schema( { name: String, lastname: String, email: String, mobile: String, address: String, material:
String, price: Number, qty: Number, discount: Number, total: Number, }, { timestamps: true }
));module.exports = Customer;
```

employee.model.js(Compressed code)

```
const mongoose = require("mongoose");const Employee = mongoose.model( "employee", new
mongoose.Schema( { name: String, mobile: String, email: String, address: String, dob: String, }, { timestamps:
true } ));module.exports = Employee;
```

index.js(Compressed code)

```
const dbConfig = require("../config/db.config.js");const mongoose = require("mongoose");mongoose.Promise
= global.Promise;const db = {};db.mongoose = mongoose;db.url = dbConfig.url;db.tutorials =
require("./tutorial.model.js");db.employee = require("./employee.model.js");db.customer =
require("./customer.model.js");db.report = require("./report.model.js");db.stock =
require("./stock.model.js");db.suppliers = require("./suppliers.model.js");db.user =
require("./user.model");db.role = require("./role.model");db.ROLES = ["user", "admin",
"moderator"];module.exports = db;
```

report.model.js(Compressed code)

```
const mongoose = require("mongoose");const Report = mongoose.model( "report", new mongoose.Schema( {
ptitle: String, pcode: { type: String, unique: true, required: true, dropDups: true }, pbrand: String, tstock: String,
cpi: String, tcost: String, sub: String, ac: String, }, { timestamps: true } ));module.exports = Report;
```

stock.model.js(Compressed code)

```
const mongoose = require("mongoose");const Stock = mongoose.model( "stock", new mongoose.Schema( {
ptitle: String, pcode: String, tstock: String, cpi: String, tcost: String, ac: String, }, { timestamps: true }
));module.exports = Stock;
```

suppliers.model.js(Compressed code)

```
const mongoose = require("mongoose");const Suppliers = mongoose.model( "suppliers", new
mongoose.Schema( { name: String, mobile: String, payment: String, materials: String, address: String, }, {
timestamps: true } ));module.exports = Suppliers;
```

user.model.js(Compressed code)

```
const mongoose = require("mongoose");const User = mongoose.model( "User", new mongoose.Schema({
username: String, email: String, password: String, roles: [ { type: mongoose.Schema.Types.ObjectId, ref: "Role"
} ] }));module.exports = User;
```

Server.js(Compressed code)

```
const express = require("express");const sendGridMail = require('@sendgrid/mail');const cors =
require("cors");const cookieSession = require("cookie-session");const app = express();var corsOptions = {
origin: "http://localhost:4200"};app.use(cors(corsOptions));// parse requests of content-type -
application/jsonapp.use(express.json());// parse requests of content-type - application/x-www-form-
urlencodedapp.use(express.urlencoded({ extended: true }));app.use( cookieSession({ name: "inventory-
session", secret: "COOKIE_SECRET", // should use as secret environment variable httpOnly: true }));const db =
require("./app/models");const Role = db.role;db.mongoose .connect(db.url, { useNewUrlParser: true,
useUnifiedTopology: true }) .then(() => { console.log("Connected to the database!"); initial(); }) .catch(err => {
console.log("Cannot connect to the database!", err); process.exit(); });// simple routeapp.get("/", (req, res) => {
res.json({ message: "Welcome to bezkoder application."
});});require("./app/routes/turorial.routes")(app);require("./app/routes/employee.routes")(app);require("./app
/routes/customer.routes")(app);require("./app/routes/report.routes")(app);require("./app/routes/stock.route
s")(app);require('./app/routes/auth.routes')(app);require('./app/routes/user.routes')(app);require("./app/rout
es/suppliers.routes")(app);require("./app/routes/mail.routes")(app);// set port, listen for requestsconst PORT
= process.env.PORT || 8080;app.listen(PORT, () => { console.log(`Server is running on port
${PORT}.`);});function initial() { Role.estimatedDocumentCount((err, count) => { if (!err && count === 0) { new
Role({ name: "user" }).save(err => { if (err) { console.log("error", err); } console.log("added 'user' to roles
collection"); }); new Role({ name: "moderator" }).save(err => { if (err) { console.log("error", err); }
console.log("added 'moderator' to roles collection"); }); new Role({ name: "admin" }).save(err => { if (err) {
console.log("error", err); } console.log("added 'admin' to roles collection"); }); } });}
```

package.json:

"node"requires": { "@sendgrid/client": "^6.5.5", "@sendgrid/helpers": "^6.5.5" } } } }, "oauth-sign": { "version":
"0.9.0", "resolved": "https://registry.npmjs.org/oauth-sign/-/oauth-sign-0.9.0.tgz", "integrity": "sha512-
fexhUFFPTGV8ybAtSIGbV6gOkSv8UtRbDBnAyLQw4QPKkgNlsH2ByPGtMUqdWkos6YCRmAqViwgZrJc/mRDzZQ
==" }, "object-assign": { "version": "4.1.1", "resolved": "https://registry.npmjs.org/object-assign/-/object-
assign-4.1.1.tgz", "integrity": "sha512-
rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU5hj+BcUg=="
}, "object-inspect": { "version": "1.12.2", "resolved": "https://registry.npmjs.org/object-inspect/-/object-
inspect-1.12.2.tgz", "integrity": "sha512-
z+cPxW0QGUp0mcqcsgQyLVRDoXFQbXOwBaqyF7VIgI4TWNQsDHrBpUQslRmIfAoYWdYzs6UlKJtB2XJpTaNSpQ
==" }, "on-finished": { "version": "2.4.1", "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-
2.4.1.tgz", "integrity": "sha512-
oVlzkg3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+brtuJjePWiYF/ClmuDr8Ch5+kg=="
, "requires": { "ee-first": "1.1.1" } }, "on-headers": { "version": "1.0.2", "resolved":
"https://registry.npmjs.org/on-headers/-/on-headers-1.0.2.tgz", "integrity": "sha512-
pZAE+FJLoyITytdqK0U5s+FIpjN0JP3OzFi/u8Rx+EV5/W+JTWGXG8xFzevE7AjBfDqHv/8vL8qQsIhHnqRkrA==" },
"parseurl": { "version": "1.3.3", "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",
"integrity": "sha512-
CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL+qQ=="
}, "path-to-regexp": { "version": "0.1.7", "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-
regexp-0.1.7.tgz", "integrity": "sha512-
5DFkuoqlv1uYQKxy8omFBeJPQcdoE07Kv2sferDCrAq1ohOU+MSDswDIbnx3YAM60qIOnYa53wBhXW0EbMonrQ
==" }, "performance-now": { "version": "2.1.0", "resolved": "https://registry.npmjs.org/performance-now/-
/performance-now-2.1.0.tgz", "integrity": "sha512-
7EAHlyLHI56VEIdK57uwHdHKIaAGbnXPiw0yWbarQZOKaKpvUIgW0jWRVLiatnM+XXlSwsanIBH/hzGMJulMow==   "
}, "proxy-addr": { "version": "2.0.7", "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-
2.0.7.tgz", "integrity": "sha512-
llQsMLSUDUPT44jdrU/O37qlnifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/gJEdZU7KMraoK1+XYAg==",
"requires": { "forwarded": "0.2.0", "ipaddr.js": "1.9.1" } }, "psl": { "version": "1.9.0", "resolved":
"https://registry.npmjs.org/psl/-/psl-1.9.0.tgz", "integrity": "sha512-
E/ZsdU4HLs/68gYzgGTkMicWTLPdAftJLfJFlLUAAKZGkStNU72sZjT66SnMDVOfOWY/YAoiD7Jxa9iHvngcag==" },
"punycode": { "version": "2.1.1", "resolved": "https://registry.npmjs.org/punycode/-/punycode-2.1.1.tgz",
"integrity": "sha512-
XRsRjdf+j5ml+y/6GKHPZbrF/8p2Yga0JPtdqTIY2Xe5ohJPD9saDJJLPvp9+NSBprVvevdXZybnj2cv8OEd0A==" },
"qs": { "version": "6.11.0", "resolved": "https://registry.npmjs.org/qs/-/qs-6.11.0.tgz", "integrity": "sha512-
MvjoMCJwEarSbUYk5O+nmoSzSutSsTwF85zcHPQ9OrlFoZOYIjaqBAJIqIXjptyD5vThxGq52Xu/MaJzRkIk4Q==",

"requires": { "side-channel": "^1.0.4" } }, "range-parser": { "version": "1.2.1", "resolved":
"https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz", "integrity": "sha512-
Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRoyzfG3ccAveqCBrwr/2yxQ5BVd/GTl5agOwSg==
" }, "raw-body": { "version": "2.5.1", "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.1.tgz",
"integrity": "sha512-
qqJBtEyVgS0ZmPGdCFPWJ3FreoqvG4MVQln/kCgF7Olq95IbOp0/BWyMwbdtn4VTvkM8Y7khCQ2Xgk/tcrCXig==
", "requires": { "bytes": "3.1.2", "http-errors": "2.0.0", "iconv-lite": "0.4.24", "unpipe": "1.0.0" } }, "request": {
"version": "2.88.2", "resolved": "https://registry.npmjs.org/request/-/request-2.88.2.tgz", "integrity": "sha512-
MsvtOrfG9ZcrOwAW+Qi+F6HbD0CWXEh9ou77uOb7FM2WPhwT7smM833PzanhJLsgXjN89Ir6V2PczXNnMpwK
hw==", "requires": { "aws-sign2": "~0.7.0", "aws4": "^1.8.0", "caseless": "~0.12.0", "combined-stream":
"~1.0.6", "extend": "~3.0.2", "forever-agent": "~0.6.1", "form-data": "~2.3.2", "har-validator": "~5.1.3", "http-
signature": "~1.2.0", "is-typedarray": "~1.0.0", "isstream": "~0.1.2", "json-stringify-safe": "~5.0.1", "mime-
types": "~2.1.19", "oauth-sign": "~0.9.0", "performance-now": "^2.1.0", "qs": "~6.5.2", "safe-buffer": "^5.1.2",
"tough-cookie": "~2.5.0", "tunnel-agent": "^0.6.0", "uuid": "^3.3.2" }, "dependencies": { "form-data": {
"version": "2.3.3", "resolved": "https://registry.npmjs.org/form-data/-/form-data-2.3.3.tgz", "integrity":
"sha512-
1lLKB2Mu3aGP1Q/2eCOx0fNbRMe7XdwktwOruhfqqd0rIJWwN4Dh+E3hrPSlDCXnSR7UtZ1N38rVXm+6+MEhJQ
==", "requires": { "asynckit": "^0.4.0", "combined-stream": "^1.0.6", "mime-types": "^2.1.12" } }, "qs": {
"version": "6.5.3", "resolved": "https://registry.npmjs.org/qs/-/qs-6.5.3.tgz", "integrity": "sha512-
qxXIEh4pCGfHICj1mAJQ2/2XVZkjCDTcEgfoSQxc/fYivUZxTkk7L3bDBJSoNrEzXI17oUO5Dp07ktqE5KzczA==" },
"uuid": { "version": "3.4.0", "resolved": "https://registry.npmjs.org/uuid/-/uuid-3.4.0.tgz", "integrity": "sha512-
HjSDRw6gZE5JMggctHBcjVak08+KEVhSIiDzFnT9S9aegmp85S/bReBVTb4QTFaRNptJ9kuYaNhnbNEOkbKb/A==" }
} }, "safe-buffer": { "version": "5.2.1", "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-
5.2.1.tgz", "integrity": "sha512-
rp3So07KcdmmKbGvgaNxQSJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCXmpuYvmCX
Q==" }, "safer-buffer": { "version": "2.1.2", "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-
2.1.2.tgz", "integrity": "sha512-
YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQriUtg==" },
"saslprep": { "version": "1.0.3", "resolved": "https://registry.npmjs.org/saslprep/-/saslprep-1.0.3.tgz",
"integrity": "sha512-
/MY/PEMbk2SuY5sScONwhUDsV2p77Znkb/q3nSVstq/yQzYJOH/Azh29p9oJLsl3LnQwSvZDKagDGBsBwSooag==
", "optional": true, "requires": { "sparse-bitfield": "^3.0.3" } }, "semver": { "version": "5.7.1", "resolved":
"https://registry.npmjs.org/semver/-/semver-5.7.1.tgz", "integrity": "sha512-
sauaDf/PZdVgrLTNYHRtpXa1iRiKcaebiKQ1BJdpQlWH2lCvexQdX55snPFyK7QzpudqbCI0qXFfOasHdyNDGQ==" },
"send": { "version": "0.18.0", "resolved": "https://registry.npmjs.org/send/-/send-0.18.0.tgz", "integrity":
"sha512-
qqWzuOjSFOuqPjFe4NOsMLafToQQwBSOEpS+FwEt3A2V3vKubTquT3vmLTQpFgMXp8AlFWFuP1qKaJZOtPpVXg
==", "requires": { "debug": "2.6.9", "depd": "2.0.0", "destroy": "1.2.0", "encodeurl": "~1.0.2", "escape-html":
"~1.0.3", "etag": "~1.8.1", "fresh": "0.5.2", "http-errors": "2.0.0", "mime": "1.6.0", "ms": "2.1.3", "on-finished":
"2.4.1", "range-parser": "~1.2.1", "statuses": "2.0.1" }, "dependencies": { "ms": { "version": "2.1.3", "resolved":
"https://registry.npmjs.org/ms/-/ms-2.1.3.tgz", "integrity": "sha512-
6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA==" } } },
"serve-static": { "version": "1.15.0", "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-
1.15.0.tgz", "integrity": "sha512-
XGuRDNjXUijsUL0vl6nSD7cwURuzEgglbOaFuZM9g3kwDXOWVTck0jLzjPzGD+TazWbboZYu52/9/XPdUgne9g=="
, "requires": { "encodeurl": "~1.0.2", "escape-html": "~1.0.3", "parseurl": "~1.3.3", "send": "0.18.0" } },
"setprototypeof": { "version": "1.2.0", "resolved": "https://registry.npmjs.org/setprototypeof/-
/setprototypeof-1.2.0.tgz", "integrity": "sha512-
E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrlODCrkoxMEeNi/XZIwuRvY4XNwYMJp
w==" }, "side-channel": { "version": "1.0.4", "resolved": "https://registry.npmjs.org/side-channel/-/side-
channel-1.0.4.tgz", "integrity": "sha512-
q5XPytqFEIKHkGdiMIrY10mvLRvnQh42/+GoBlFW3b2LXLE2xxJpZFdm94we0BaoV3RwJyGqg5wS7epxTv0Zvw=="
, "requires": { "call-bind": "^1.0.0", "get-intrinsic": "^1.0.2", "object-inspect": "^1.9.0" } }, "sift": { "version":

"16.0.1", "resolved": "https://registry.npmjs.org/sift/-/sift-16.0.1.tgz", "integrity": "sha512-
Wv6BjQ5zbhW7VFefWusVP33T/EM0vYikCaQ2qR8yULbsilAT8/wQaXvuQ3ptGLpoKx+lihJE3y2UTgKDyyNHZQ=="
}, "smart-buffer": { "version": "4.2.0", "resolved": "https://registry.npmjs.org/smart-buffer/-/smart-buffer-
4.2.0.tgz", "integrity": "sha512-
94hK0Hh8rPqQl2xXc3HsaBoOXKV20MToPkcXvwbISWLEs+64sBq5kFgn2kJDHb1Pry9yrP0dxrCI9RRci7RXKg==" },
"socks": { "version": "2.7.1", "resolved": "https://registry.npmjs.org/socks/-/socks-2.7.1.tgz", "integrity":
"sha512-
7maUZy1N7uo6+WVEX6psASxtNlKaNVMlGQKkG/63nEDdLOWNbiUMoLK7X4uYoLhQstau72mLgfEWcXcwsaHbY
Q==", "requires": { "ip": "^2.0.0", "smart-buffer": "^4.2.0" } }, "sparse-bitfield": { "version": "3.0.3", "resolved":
"https://registry.npmjs.org/sparse-bitfield/-/sparse-bitfield-3.0.3.tgz", "integrity": "sha512-
kvzhi7vqKTfkh0PZU+2D2PIllw2ymqJKujUcyPMd9Y75Nv4nPbGJZXNhxsgdQab2BmlDct1YnfQCguEvHr7VsQ==",
"optional": true, "requires": { "memory-pager": "^1.0.2" } }, "sshpk": { "version": "1.17.0", "resolved":
"https://registry.npmjs.org/sshpk/-/sshpk-1.17.0.tgz", "integrity": "sha512-
/9HIEs1ZXGhSPE8X6Ccm7Nam1z8KcoCqPdI7ecm1N33EzAetWahvQWVqLZtaZQ+IDKX4IyA2o0gBzqIMkAagHQ=
=", "requires": { "asn1": "~0.2.3", "assert-plus": "^1.0.0", "bcrypt-pbkdf": "^1.0.0", "dashdash": "^1.12.0", "ecc-
jsbn": "~0.1.1", "getpass": "^0.1.1", "jsbn": "~0.1.0", "safer-buffer": "^2.0.2", "tweetnacl": "~0.14.0" } },
"statuses": { "version": "2.0.1", "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",
"integrity": "sha512-
RwNA9Z/7PrK06rYLIzFMlaF+l73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB2fOC4AjuYSEndS7N/mTCbkdQ=="
}, "strnum": { "version": "1.0.5", "resolved": "https://registry.npmjs.org/strnum/-/strnum-1.0.5.tgz",
"integrity": "sha512-
J8bbNyKKXl5qYcR36TIO8W3mVGVHrmmxsd5PAItGkmyzwJvybiw2IVq5nqd0i4LSNSkB/sx9VHllbfFdr9k1JA==",
"optional": true }, "supports-color": { "version": "5.5.0", "resolved": "https://registry.npmjs.org/supports-
color/-/supports-color-5.5.0.tgz", "integrity": "sha512-
QjVjwdXIt408MIiAqCX4oUKsgU2EqAGzs2Ppkm4aQYbjm+ZEWEcW4SfFNTr4uMNZma0ey4f5lgLrkB0aX0QMow=
=", "requires": { "has-flag": "^3.0.0" } }, "toidentifier": { "version": "1.0.1", "resolved":
"https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz", "integrity": "sha512-
o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTlZUtEe+JG7s5YhEz608PlAHRA==" },
"tough-cookie": { "version": "2.5.0", "resolved": "https://registry.npmjs.org/tough-cookie/-/tough-cookie-
2.5.0.tgz", "integrity": "sha512-
nlLsUzgm1kfLXSXfRZMc1KLAugd4hqJHDTvc2hDIwS3mZAfMEuMbc03SujMF+GEcpaX/qboeycw6iO8JwVv2+g=="
, "requires": { "psl": "^1.1.28", "punycode": "^2.1.1" } }, "tr46": { "version": "3.0.0", "resolved":
"https://registry.npmjs.org/tr46/-/tr46-3.0.0.tgz", "integrity": "sha512-
l7FvfAHlcmulp8kr+flpQZmVwtu7nfRV7NZujtN0OqES8EL4O4e0qqzL0DC5gAvx/ZC/9lk6rhcUwYvkBnBnYA==",
"requires": { "punycode": "^2.1.1" } }, "tslib": { "version": "2.4.0", "resolved": "https://registry.npmjs.org/tslib/-
/tslib-2.4.0.tgz", "integrity": "sha512-
d6xOpEDfsi2CZVlPQzGeux8XMwLT9hssAsaPYExaQMuYskwb+x1x7J371tWlbBdWHroy99KnVB6qIkUbs5X3UQ==
", "optional": true }, "tsscmp": { "version": "1.0.6", "resolved": "https://registry.npmjs.org/tsscmp/-/tsscmp-
1.0.6.tgz", "integrity": "sha512-
LxhtAkPDTkVCMQjt2h6eBVY28KCjikZqZfMcC15YBeNjkgUpdCfBu5HoiOTDu86v6smE8yOjyEktJ8hlbANHQA==" },
"tunnel-agent": { "version": "0.6.0", "resolved": "https://registry.npmjs.org/tunnel-agent/-/tunnel-agent-
0.6.0.tgz", "integrity": "sha512-
McnNiV1l8RYeY8tBgEpuodCC1mLUdbSN+CYBL7kJsJNInOP8UjDDEwdk6Mw60vdLLrr5NHKZhMAOSrR2NZuQ+w
==", "requires": { "safe-buffer": "^5.0.1" } }, "tweetnacl": { "version": "0.14.5", "resolved":
"https://registry.npmjs.org/tweetnacl/-/tweetnacl-0.14.5.tgz", "integrity": "sha512-
KXXFFdAbFXY4geFIwoyNK+f5Z1b7swfXABfL7HXCmoIWMKU3dmS26672A4EeQtDzLKy7SXmfBu51JolvEKwtGA=
=" }, "type-is": { "version": "1.6.18", "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",
"integrity": "sha512-
TkRKr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFPW4ky9Q+iA+ma9BGm06XQBy8g==",
"requires": { "media-typer": "0.3.0", "mime-types": "~2.1.24" } }, "unpipe": { "version": "1.0.0", "resolved":
"https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz", "integrity": "sha512-
pjy2bYhSsufwWlKwPc+l3cN7+wuJlK6uz0YdJEOlQDbl6jo/YlPi4mb8agUkVC8BF7V8NuzeyPNqRksA3hztKQ==" },
"uri-js": { "version": "4.4.1", "resolved": "https://registry.npmjs.org/uri-js/-/uri-js-4.4.1.tgz", "integrity":

```
"sha512-
7rKUyy33Q1yc98pQ1DAmLtwX109F7TIfWlW1Ydo8Wl1ii1SeHieeh0HHfPeL2fMXK6z0s8ecKs9frCuLJvndBg==",
"requires": { "punycode": "^2.1.0" } }, "utils-merge": { "version": "1.0.1", "resolved":
"https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz", "integrity": "sha512-
pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybbsHHoED/ztMtkv/VoYTYyShUn81hA=="
}, "uuid": { "version": "8.3.2", "resolved": "https://registry.npmjs.org/uuid/-/uuid-8.3.2.tgz", "integrity":
"sha512-
+NYs2QeMWy+GWFOEm9xnn6HCDp0l7QBD7ml8zLUmJ+93Q5NF0NocErnwkTkXVFNiX3/fpC6afS8Dhb/gz7R7eg
==", "optional": true }, "vary": { "version": "1.1.2", "resolved": "https://registry.npmjs.org/vary/-/vary-
1.1.2.tgz", "integrity": "sha512-
BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQOzmAKzYqYRAISoRhdQr3eIZ/PXqg=="
}, "verror": { "version": "1.10.0", "resolved": "https://registry.npmjs.org/verror/-/verror-1.10.0.tgz", "integrity":
"sha512-
ZZKSmDAEFOijERBLkmYfJ+vmk3w+7hOLYDNkRCuRuMJGEmqYNCNLyBBFwWKVMhfwaEF3WOd0Zlw86U/WC/+
nYw==", "requires": { "assert-plus": "^1.0.0", "core-util-is": "1.0.2", "extsprintf": "^1.2.0" } }, "webidl-
conversions": { "version": "7.0.0", "resolved": "https://registry.npmjs.org/webidl-conversions/-/webidl-
conversions-7.0.0.tgz", "integrity": "sha512-
VwddBukDzu71offAQR975unBIGqfKZpM+8ZX6ySk8nYhVoo5CYaZyzt3YBvYtRtO+aoGlqxPg/B87NGVZ/fu6g==" },
"whatwg-url": { "version": "11.0.0", "resolved": "https://registry.npmjs.org/whatwg-url/-/whatwg-url-
11.0.0.tgz", "integrity": "sha512-
RKT8HExMpoYx4igMiVMY83lN6UeITKJlBQ+vR/8ZJ8OCdSiN3RwCq+9gH0+Xzj0+5IrM6i4j/6LuvzbZIQgEcQ==",
"requires": { "tr46": "^3.0.0", "webidl-conversions": "^7.0.0" } } }}
```

Controllers:

auth.controller.js

```
: const config = require("../config/auth.config");const db = require("../models");const User = db.user;const
Role = db.role;var jwt = require("jsonwebtoken");var bcrypt = require("bcryptjs");exports.signup = (req, res) =>
{ const user = new User({ username: req.body.username, email: req.body.email, password:
bcrypt.hashSync(req.body.password, 8), }); user.save((err, user) => { if (err) { res.status(500).send({ message:
err }); return; } if (req.body.roles) { Role.find( { name: { $in: req.body.roles }, }, (err, roles) => { if (err) {
res.status(500).send({ message: err }); return; } user.roles = roles.map((role) => role._id); user.save((err) => { if
(err) { res.status(500).send({ message: err }); return; } res.send({ message: "User was registered successfully!"
}); }); } ); } else { Role.findOne({ name: "user" }, (err, role) => { if (err) { res.status(500).send({ message: err });
return; } user.roles = [role._id]; user.save((err) => { if (err) { res.status(500).send({ message: err }); return; }
res.send({ message: "User was registered successfully!" }); }); }); } });};exports.signin = (req, res) => {
User.findOne({ username: req.body.username, }) .populate("roles", "-_v") .exec((err, user) => { if (err) {
res.status(500).send({ message: err }); return; } if (!user) { return res.status(404).send({ message: "User Not
found." }); } var passwordIsValid = bcrypt.compareSync( req.body.password, user.password ); if
(!passwordIsValid) { return res.status(401).send({ message: "Invalid Password!" }); } var token = jwt.sign({ id:
user.id }, config.secret, { expiresIn: 86400, // 24 hours }); var authorities = []; for (let i = 0; i < user.roles.length;
i++) { authorities.push("ROLE_" + user.roles[i].name.toUpperCase()); } req.session.token = token;
res.status(200).send({ id: user._id, username: user.username, email: user.email, roles: authorities, });
});};exports.signout = async (req, res) => { try { req.session = null; return res.status(200).send({ message:
"You've been signed out!" }); } catch (err) { this.next(err); }};
```

customer.controller.js

```
const db = require("../models");const Customer = db.customer;const Report = db.report;// Create and Save a
new Customerexports.create = (req, res) => { // Validate request if (!req.body.email) { res.status(400).send({
message: "Content can not be empty!" }); return; } // Create a Customer const customer = new Customer({
```

```
name: req.body.name, material: req.body.material, email: req.body.email, mobile: req.body.mobile, address:
req.body.address, price: req.body.price, qty: req.body.qty, discount: req.body.discount, total: req.body.total,
}); // Update a Report const report = { tstock: req.body.tstock, }; Report.findByIdAndUpdate(req.body.pid,
report) .then((data) => {}) .catch((err) => { res.status(500).send({ message: err.message || "Some error
occurred while Update the Report.", }); }); // Save Customer in the database customer .save(customer)
.then((data) => { res.send(data); }) .catch((err) => { res.status(500).send({ message: err.message || "Some error
occurred while creating the Customer.", }); });};// Retrieve all Customers from the database.exports.findAll =
(req, res) => { const title = req.query.title; var condition = title ? { title: { $regex: new RegExp(title), $options: "i"
} } : {}; Customer.find(condition) .then((data) => { res.send(data); }) .catch((err) => { res.status(500).send({
message: err.message || "Some error occurred while retrieving customer.", }); });};// Find a single Customer
with an idexports.findOne = (req, res) => { const id = req.params.id; Customer.findById(id) .then((data) => { if
(!data) res.status(404).send({ message: "Not found Customer with id " + id }); else res.send(data); }) .catch((err)
=> { res .status(500) .send({ message: "Error retrieving Customer with id=" + id }); });};// Update a Customer by
the id in the requestexports.update = (req, res) => { if (!req.body) { return res.status(400).send({ message:
"Data to update can not be empty!", }); } const id = req.params.id; Customer.findByIdAndUpdate(id, req.body,
{ useFindAndModify: false }) .then((data) => { if (!data) { res.status(404).send({ message: `Cannot update
Customer with id=${id}. Maybe Customer was not found!`, }); } else res.send({ message: "Customer was
updated successfully." }); }) .catch((err) => { res.status(500).send({ message: "Error updating Customer with
id=" + id, }); });};// Delete a Customer with the specified id in the requestexports.delete = (req, res) => { const id
= req.params.id; Customer.findByIdAndRemove(id, { useFindAndModify: false }) .then((data) => { if (!data) {
res.status(404).send({ message: `Cannot delete Customer with id=${id}. Maybe Customer was not found!`, }); }
else { res.send({ message: "Customer was deleted successfully!", }); } }) .catch((err) => { res.status(500).send({
message: "Could not delete Customer with id=" + id, }); });};// Delete all Customers from the
database.exports.deleteAll = (req, res) => { Customer.deleteMany({}) .then((data) => { res.send({ message:
`${data.deletedCount} Customers were deleted successfully!`, }); }) .catch((err) => { res.status(500).send({
message: err.message || "Some error occurred while removing all customer.", }); });};// Find all published
Customersexports.findAllPublished = (req, res) => { Customer.find({ published: true }) .then((data) => {
res.send(data); }) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while
retrieving customer.", }); });};
```

employee.controller.js

```
const db = require("../models");const Employee = db.employee;// Create and Save a new
Employeeexports.create = (req, res) => { // Validate request if (!req.body.name) { res.status(400).send({
message: "Content can not be empty!" }); return; } // Create a Employee const employee = new Employee({
name: req.body.name, mobile: req.body.mobile, address: req.body.address, email: req.body.email, dob:
req.body.dob, }); // Save Employee in the database employee .save(employee) .then((data) => { res.send(data);
}) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while creating the
Employee.", }); });};// Retrieve all Employees from the database.exports.findAll = (req, res) => { const title =
req.query.title; var condition = title ? { title: { $regex: new RegExp(title), $options: "i" } } : {};
Employee.find(condition) .then((data) => { res.send(data); }) .catch((err) => { res.status(500).send({ message:
err.message || "Some error occurred while retrieving employee.", }); });};// Find a single Employee with an
idexports.findOne = (req, res) => { const id = req.params.id; Employee.findById(id) .then((data) => { if (!data)
res.status(404).send({ message: "Not found Employee with id " + id }); else res.send(data); }) .catch((err) => {
res .status(500) .send({ message: "Error retrieving Employee with id=" + id }); });};// Update a Employee by the
id in the requestexports.update = (req, res) => { if (!req.body) { return res.status(400).send({ message: "Data to
update can not be empty!", }); } const id = req.params.id; Employee.findByIdAndUpdate(id, req.body, {
useFindAndModify: false }) .then((data) => { if (!data) { res.status(404).send({ message: `Cannot update
Employee with id=${id}. Maybe Employee was not found!`, }); } else res.send({ message: "Employee was
updated successfully." }); }) .catch((err) => { res.status(500).send({ message: "Error updating Employee with
id=" + id, }); });};// Delete a Employee with the specified id in the requestexports.delete = (req, res) => { const id
= req.params.id; Employee.findByIdAndRemove(id, { useFindAndModify: false }) .then((data) => { if (!data) {
res.status(404).send({ message: `Cannot delete Employee with id=${id}. Maybe Employee was not found!`, }); }
else { res.send({ message: "Employee was deleted successfully!", }); } }) .catch((err) => { res.status(500).send({
message: "Could not delete Employee with id=" + id, }); });};// Delete all Employees from the
database.exports.deleteAll = (req, res) => { Employee.deleteMany({}) .then((data) => { res.send({ message:
`${data.deletedCount} Employees were deleted successfully!`, }); }) .catch((err) => { res.status(500).send({
message: err.message || "Some error occurred while removing all employee.", }); });};// Find all published
Employeesexports.findAllPublished = (req, res) => { Employee.find({ published: true }) .then((data) => {
res.send(data); }) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while
retrieving employee.", }); });};
```

mail.controller.js

```
const sendGridMail = require("@sendgrid/mail");sendGridMail.setApiKey(
"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");// Create and Save a
new Tutorialexports.create = (req, res) => { const body = "Inventory Contact Form"; const mailBody = { to:
"paulananth1219@gmail.com", from: "paulananth1219@gmail.com", subject: "Contact Form", text: "Contact",
html: `<html lang=en xmlns=http://www.w3.org/1999/xhtml xmlns:o=urn:schemas-microsoft-
com:office:office xmlns:v=urn:schemas-microsoft-com:vml><meta charset=utf-8><meta
content="width=device-width"name=viewport><meta content="IE=edge"http-equiv=X-UA-Compatible><meta
name=x-apple-disable-message-reformatting><title></title><link
href="https://fonts.googleapis.com/css?family=Lato:300,400,700"rel=stylesheet><style>body,html{margin:0
auto!important;padding:0!important;height:100%!important;width:100%!important;background:#f1f1f1}*{-
ms-text-size-adjust:100%;-webkit-text-size-adjust:100%}div[style*="margin: 16px
0"]{margin:0!important}table,td{mso-table-lspace:0!important;mso-table-rspace:0!important}table{border-
spacing:0!important;border-collapse:collapse!important;table-layout:fixed!important;margin:0
auto!important}img{-ms-interpolation-mode:bicubic}a{text-decoration:none}.aBn,.unstyle-auto-detected-links
*,[x-apple-data-detectors]{border-bottom:0!important;cursor:default!important;color:inherit!important;text-
decoration:none!important;font-size:inherit!important;font-family:inherit!important;font-
weight:inherit!important;line-
height:inherit!important}.a6S{display:none!important;opacity:.01!important}.im{color:inherit!important}img.g
-img+div{display:none!important}@media only screen and (min-device-width:320px) and (max-device-
width:374px){u~div .email-container{min-width:320px!important}}@media only screen and (min-device-
width:375px) and (max-device-width:413px){u~div .email-container{min-width:375px!important}}@media
only screen and (min-device-width:414px){u~div .email-container{min-
width:414px!important}}</style><style>.primary{background:#30e3ca}.bg_white{background:#fff}.bg_light{ba
ckground:#fafafa}.bg_black{background:#000}.bg_dark{background:rgba(0,0,0,.8)}.email-
section{padding:2.5em}.btn{padding:10px 15px;display:inline-block}.btn.btn-primary{border-
radius:5px;background:#30e3ca;color:#fff}.btn.btn-white{border-
radius:5px;background:#fff;color:#000}.btn.btn-white-outline{border-radius:5px;background:0 0;border:1px
solid #fff;color:#fff}.btn.btn-black-outline{border-radius:0;background:0 0;border:2px solid
#000;color:#000;font-weight:700}h1,h2,h3,h4,h5,h6{font-family:Lato,sans-serif;color:#000;margin-top:0;font-
weight:400}body{font-family:Lato,sans-serif;font-weight:400;font-size:15px;line-
height:1.8;color:rgba(0,0,0,.4)}a{color:#30e3ca}.logo h1{margin:0}.logo h1 a{color:#30e3ca;font-
size:24px;font-weight:700;font-family:Lato,sans-serif}.hero{position:relative;z-index:0}.hero
.text{color:rgba(0,0,0,.3)}.hero .text h2{color:#000;font-size:40px;margin-bottom:0;font-weight:400;line-
height:1.4}.hero .text h3{font-size:24px;font-weight:300}.hero .text h2 span{font-
weight:600;color:#30e3ca}.heading-section h2{color:#000;font-size:28px;margin-top:0;line-height:1.4;font-
weight:400}.heading-section .subheading{margin-bottom:20px!important;display:inline-block;font-
size:13px;text-transform:uppercase;letter-spacing:2px;color:rgba(0,0,0,.4);position:relative}.heading-section
.subheading::after{position:absolute;left:0;right:0;bottom:-
10px;content:'';width:100%;height:2px;background:#30e3ca;margin:0 auto}.heading-section-
white{color:rgba(255,255,255,.8)}.heading-section-white h2{font-family:line-height: 1;padding-
bottom:0}.heading-section-white h2{color:#fff}.heading-section-white .subheading{margin-
bottom:0;display:inline-block;font-size:13px;text-transform:uppercase;letter-
spacing:2px;color:rgba(255,255,255,.4)}ul.social{padding:0}ul.social li{display:inline-block;margin-
right:10px}.footer{border-top:1px solid rgba(0,0,0,.05);color:rgba(0,0,0,.5)}.footer .heading{color:#000;font-
size:20px}.footer ul{margin:0;padding:0}.footer ul li{list-style:none;margin-bottom:10px}.footer ul li
a{color:rgba(0,0,0,1)}</style><body style=margin:0;padding:0!important;mso-line-height-
rule:exactly;background-color:#f1f1f1 width=100%><center style=width:100%;background-color:#f1f1f1><div
style=display:none;font-size:1px;max-height:0;max-width:0;opacity:0;overflow:hidden;mso-hide:all;font-
family:sans-serif></div><div style="max-width:600px;margin:0 auto"class=email-container><table border=0
cellpadding=0 cellspacing=0 role=presentation width=100% align=center style=margin:auto><tr><td
class=bg_white style="padding:1em 2.5em 0 2.5em"valign=top><table border=0 cellpadding=0 cellspacing=0
role=presentation width=100%><tr><td class=logo style=text-align:center><h1><a>INVENTORY -
TCE</a></h1></table><tr><td class="bg_white hero"style="padding:2em 0 4em
0"valign=middle><table><tr><td><div style="padding:0 2.5em;text-align:center"class=text><h5>Email :
${req.body.mail}</h5><h5>Mobile : ${req.body.mobile}</h5><h5>Name :
${req.body.name}</h5><p><strong>Message
:</strong>${req.body.message}</div></table></table></div></center>`, }; try { sendGridMail.send(mailBody);
res.status(200).send({ message: "Test email sent successfully !" }); } catch (error) { res.status(400).send({
message: "" }); if (error.response) { console.error(error.response.body); } }};
```

report.controller.js

```
const db = require("../models");const Report = db.report;// Create and Save a new Reportexports.create =
(req, res) => { // Validate request if (!req.body.pcode) { res.status(400).send({ message: "Content can not be
empty!" }); return; } // Create a Report const report = new Report({ pcode: req.body.pcode, ptitle:
req.body.ptitle, pbrand: req.body.pbrand, tstock: req.body.tstock, ac: req.body.ac, cpi: req.body.cpi, tcost:
req.body.tcost, sub: req.body.sub, }); // Save Report in the database report .save(report) .then((data) => {
res.send(data); }) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while
creating the Report.", }); });};// Retrieve all Reports from the database.exports.findAll = (req, res) => { const
title = req.query.title; var condition = title ? { title: { $regex: new RegExp(title), $options: "i" } } : {};
Report.find(condition) .then((data) => { res.send(data); }) .catch((err) => { res.status(500).send({ message:
err.message || "Some error occurred while retrieving report.", }); });};// Find a single Report with an
idexports.findOne = (req, res) => { const id = req.params.id; Report.findById(id) .then((data) => { if (!data)
res.status(404).send({ message: "Not found Report with id " + id }); else res.send(data); }) .catch((err) => { res
.status(500) .send({ message: "Error retrieving Report with id=" + id }); });};// Update a Report by the id in the
requestexports.update = (req, res) => { if (!req.body) { return res.status(400).send({ message: "Data to update
can not be empty!", }); } const id = req.params.id; Report.findByIdAndUpdate(id, req.body, {
useFindAndModify: false }) .then((data) => { if (!data) { res.status(404).send({ message: `Cannot update Report
with id=${id}. Maybe Report was not found!`, }); } else res.send({ message: "Report was updated successfully."
}); }) .catch((err) => { res.status(500).send({ message: "Error updating Report with id=" + id, }); });};// Delete a
Report with the specified id in the requestexports.delete = (req, res) => { const id = req.params.id;
Report.findByIdAndRemove(id, { useFindAndModify: false }) .then((data) => { if (!data) { res.status(404).send({
message: `Cannot delete Report with id=${id}. Maybe Report was not found!`, }); } else { res.send({ message:
"Report was deleted successfully!", }); } }) .catch((err) => { res.status(500).send({ message: "Could not delete
Report with id=" + id, }); });};// Delete all Reports from the database.exports.deleteAll = (req, res) => {
Report.deleteMany({}) .then((data) => { res.send({ message: `${data.deletedCount} Reports were deleted
successfully!`, }); }) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while
removing all report.", }); });};// Find all published Reportsexports.findAllPublished = (req, res) => { Report.find({
published: true }) .then((data) => { res.send(data); }) .catch((err) => { res.status(500).send({ message:
err.message || "Some error occurred while retrieving report.", }); });};
```

stock.controller.js

```
const db = require("../models");const Report = db.report;const Stock = db.stock;// Create and Save a new
Stockexports.create = (req, res) => { // Validate request if (!req.body.pcode) { res.status(400).send({ message:
"Content can not be empty!" }); return; } // Create a Stock const stock = new Stock({ pcode: req.body.pcode,
ptitle: req.body.ptitle, tstock: req.body.tstock, ac: req.body.ac, cpi: req.body.cpi, tcost: req.body.tcost, }); //
Update a Report const report = { tstock: req.body.tstock, ac: req.body.ac, cpi: req.body.cpi, tcost:
req.body.tcost, }; Report.findByIdAndUpdate(req.body.pid, report) .then((data) => {}) .catch((err) => {
res.status(500).send({ message: err.message || "Some error occurred while Update the Report.", }); }); // Save
Stock in the database stock .save(stock) .then((data) => { res.send(data); }) .catch((err) => {
res.status(500).send({ message: err.message || "Some error occurred while creating the Stock.", }); });};//
Retrieve all Reports from the database.exports.findAll = (req, res) => { const title = req.query.title; var
condition = title ? { title: { $regex: new RegExp(title), $options: "i" } } : {}; Stock.find() .then((data) => {
res.send(data); }) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while
retrieving stock.", }); });};// Find a single Stock with an idexports.findOne = (req, res) => { const id =
req.params.id; Stock.findById(id) .then((data) => { if (!data) res.status(404).send({ message: "Not found Stock
with id " + id }); else res.send(data); }) .catch((err) => { res.status(500).send({ message: "Error retrieving Stock
with id=" + id }); });};// Update a Stock by the id in the requestexports.update = (req, res) => { if (!req.body) {
return res.status(400).send({ message: "Data to update can not be empty!", }); } // Update a Report const
report = { tstock: req.body.tstock, ac: req.body.ac, cpi: req.body.cpi, tcost: req.body.tcost, };
Report.findByIdAndUpdate(req.body.pid, report) .then((data) => {}) .catch((err) => { res.status(500).send({
message: err.message || "Some error occurred while Update the Report.", }); }); const id = req.params.id;
Stock.findByIdAndUpdate(id, req.body, { useFindAndModify: false }) .then((data) => { if (!data) {
res.status(404).send({ message: `Cannot update Stock with id=${id}. Maybe Stock was not found!`, }); } else
res.send({ message: "Stock was updated successfully." }); }) .catch((err) => { res.status(500).send({ message:
"Error updating Stock with id=" + id, }); });};// Delete a Stock with the specified id in the requestexports.delete
= (req, res) => { const id = req.params.id; Stock.findByIdAndRemove(id, { useFindAndModify: false })
.then((data) => { if (!data) { res.status(404).send({ message: `Cannot delete Stock with id=${id}. Maybe Stock
was not found!`, }); } else { res.send({ message: "Stock was deleted successfully!", }); } }) .catch((err) => {
res.status(500).send({ message: "Could not delete Stock with id=" + id, }); });};// Delete all Reports from the
database.exports.deleteAll = (req, res) => { Stock.deleteMany({}) .then((data) => { res.send({ message:
`${data.deletedCount} Reports were deleted successfully!`, }); }) .catch((err) => { res.status(500).send({
message: err.message || "Some error occurred while removing all stock.", }); });};// Find all published
Reportsexports.findAllPublished = (req, res) => { Stock.find({ published: true }) .then((data) => { res.send(data);
}) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while retrieving
stock.", }); });};
```

suppliers.controller.js

```
const db = require("../models");const Suppliers = db.suppliers;// Create and Save a new
Suppliersexports.create = (req, res) => { // Validate request if (!req.body.name) { res.status(400).send({
message: "Content can not be empty!" }); return; } // Create a Suppliers const suppliers = new Suppliers({
name: req.body.name, mobile: req.body.mobile, payment: req.body.payment, materials: req.body.materials,
address: req.body.address, }); // Save Suppliers in the database suppliers .save(suppliers) .then((data) => {
res.send(data); }) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while
creating the Suppliers.", }); });};// Retrieve all Supplierss from the database.exports.findAll = (req, res) => {
const title = req.query.title; var condition = title ? { title: { $regex: new RegExp(title), $options: "i" } } : {};
Suppliers.find(condition) .then((data) => { res.send(data); }) .catch((err) => { res.status(500).send({ message:
err.message || "Some error occurred while retrieving suppliers.", }); });};// Find a single Suppliers with an
idexports.findOne = (req, res) => { const id = req.params.id; Suppliers.findById(id) .then((data) => { if (!data)
res.status(404).send({ message: "Not found Suppliers with id " + id }); else res.send(data); }) .catch((err) => { res
.status(500) .send({ message: "Error retrieving Suppliers with id=" + id }); });};// Update a Suppliers by the id in
the requestexports.update = (req, res) => { if (!req.body) { return res.status(400).send({ message: "Data to
update can not be empty!", }); } const id = req.params.id; Suppliers.findByIdAndUpdate(id, req.body, {
useFindAndModify: false }) .then((data) => { if (!data) { res.status(404).send({ message: `Cannot update
Suppliers with id=${id}. Maybe Suppliers was not found!`, }); } else res.send({ message: "Suppliers was updated
successfully." }); }) .catch((err) => { res.status(500).send({ message: "Error updating Suppliers with id=" + id, });
});};// Delete a Suppliers with the specified id in the requestexports.delete = (req, res) => { const id =
req.params.id; Suppliers.findByIdAndRemove(id, { useFindAndModify: false }) .then((data) => { if (!data) {
res.status(404).send({ message: `Cannot delete Suppliers with id=${id}. Maybe Suppliers was not found!`, }); }
else { res.send({ message: "Suppliers was deleted successfully!", }); } }) .catch((err) => { res.status(500).send({
message: "Could not delete Suppliers with id=" + id, }); });};// Delete all Supplierss from the
database.exports.deleteAll = (req, res) => { Suppliers.deleteMany({}) .then((data) => { res.send({ message:
`${data.deletedCount} Supplierss were deleted successfully!`, }); }) .catch((err) => { res.status(500).send({
message: err.message || "Some error occurred while removing all suppliers.", }); });};// Find all published
Supplierssexports.findAllPublished = (req, res) => { Suppliers.find({ published: true }) .then((data) => {
res.send(data); }) .catch((err) => { res.status(500).send({ message: err.message || "Some error occurred while
retrieving suppliers.", }); });};
```

user.controller.js

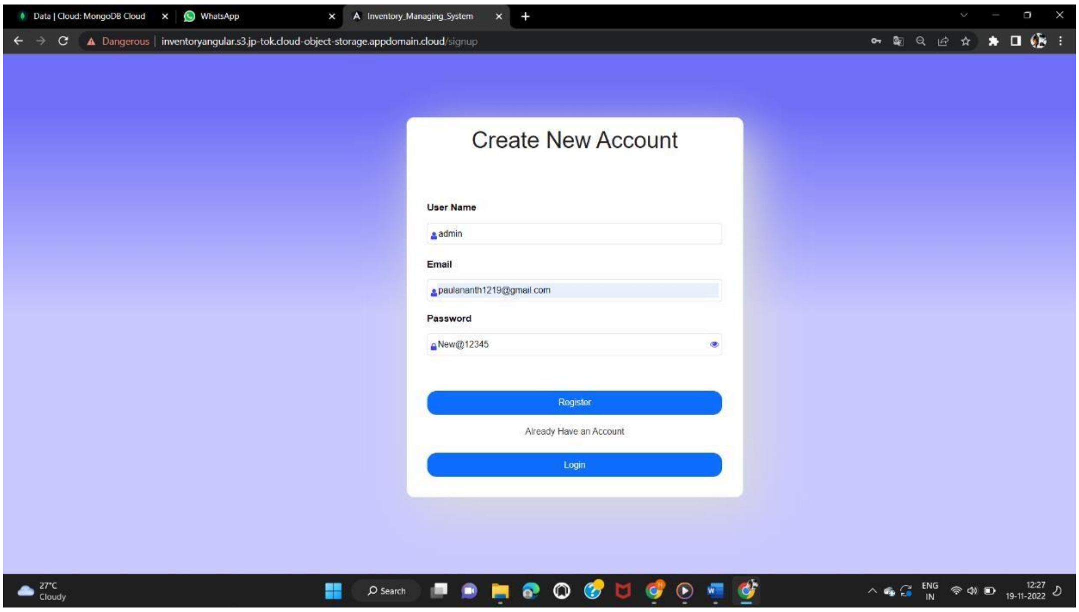
```
exports.allAccess = (req, res) => { res.status(200).send("Public Content.");};exports.userBoard = (req, res) => {
res.status(200).send("User Content.");};exports.adminBoard = (req, res) => { res.status(200).send("Admin
Content.");};exports.moderatorBoard = (req, res) => { res.status(200).send("Moderator Content.");};
```

TESTING

TEST CASE (1) : SIGNUP PAGE :

- To adding specific user name
- To adding the Specific e-mail
- To adding the password and if user want to verify the password to click the eye icon the password will visible
- To update the validation field set on the database
- To show the “successfully updating the user” popup message

ADDING USER TO THE FIELD(TESTING)


Data | Cloud: MongoDB Cloud
WhatsApp
Inventory_Managing_System
Dangerous
inventoryangular.s3.jp-tok.cloud-object-storage.appdomain.cloud/signup
Register
Already Have an Account
Login
27°C
Cloudy
Search
ENG
IN
12:27
19-11-2022

Create New Account

User Name

admin

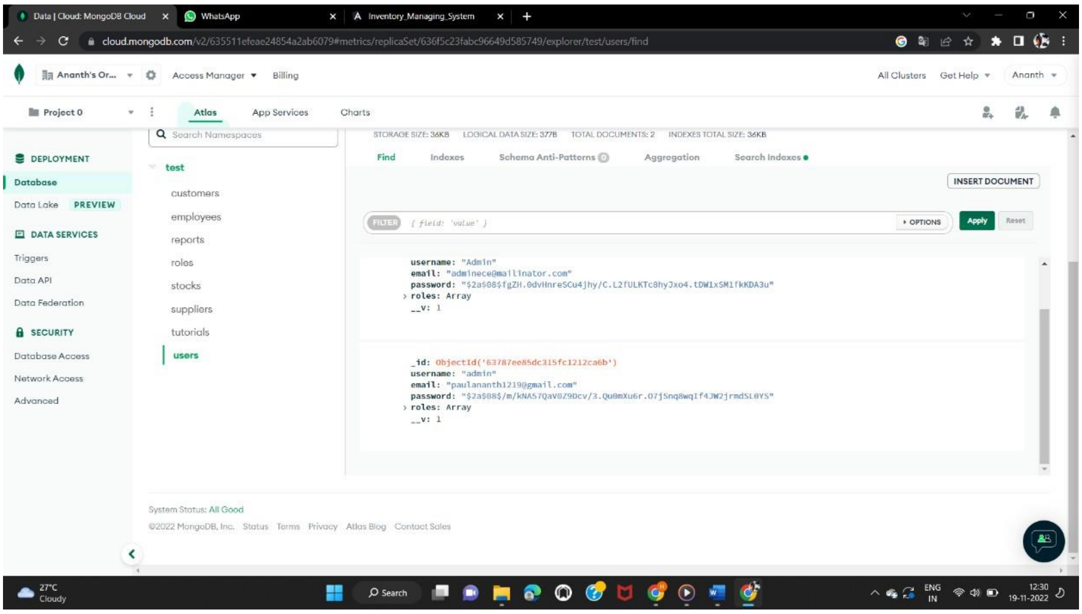
Email

paulananth1219@gmail.com

Password

New@12345

SUCCESSFULLY USER ADD IN THE BACKEND DATABASE(MONGODB)



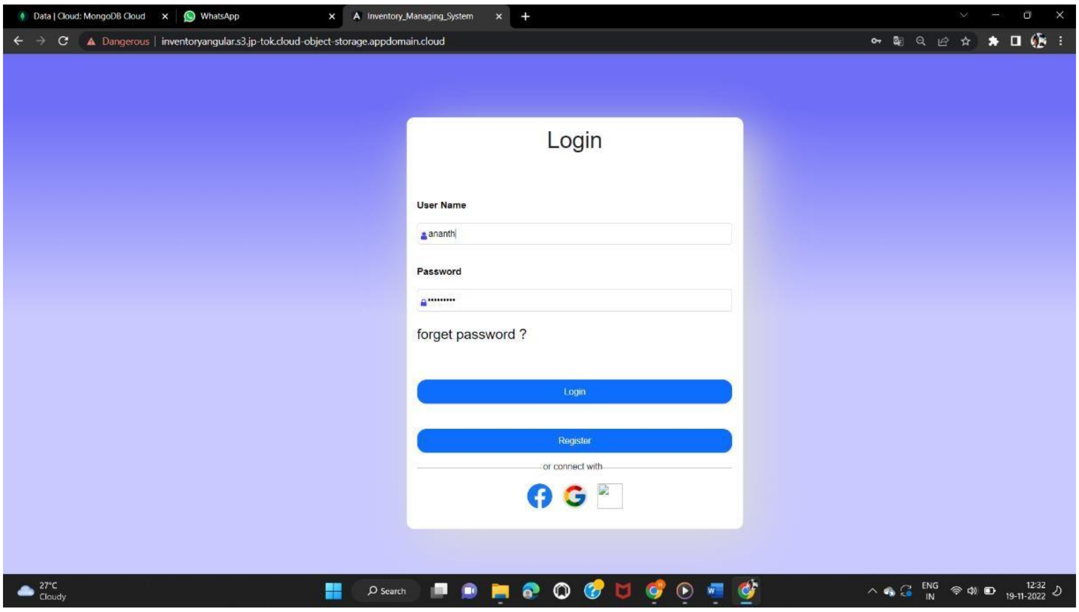
```
  username: "Admin"
  email: "adminece@mailinator.com"
  password: "$2a$08$fgZH.0dvHnreSCu4jhy/C.L2fULKTc8hyJxo4.tDW1xSM1fkKDA3u"
> roles: Array
  __v: 1

  _id: ObjectId('63787ee85dc315fc1212ca6b')
  username: "admin"
  email: "paulananth1219@gmail.com"
  password: "$2a$08$/m/kNA57QaV0Z9Dcv/3.Qu8mXu6r.O7jSnq8wqIf4JW2jrmdSL0YS"
> roles: Array
  __v: 1
```

LOGIN PAGE

TEST CASE (2) : login page

- User successfully fill the field ,if user will not enter the valid filed the "unauthorized" alert will be shown to the user
- The user enter the password ,incase the user want to verify the password to click the eye icon the password will visible to the user
- To user successfully login the "user successfully login " popup message will be shown

Checking the login field

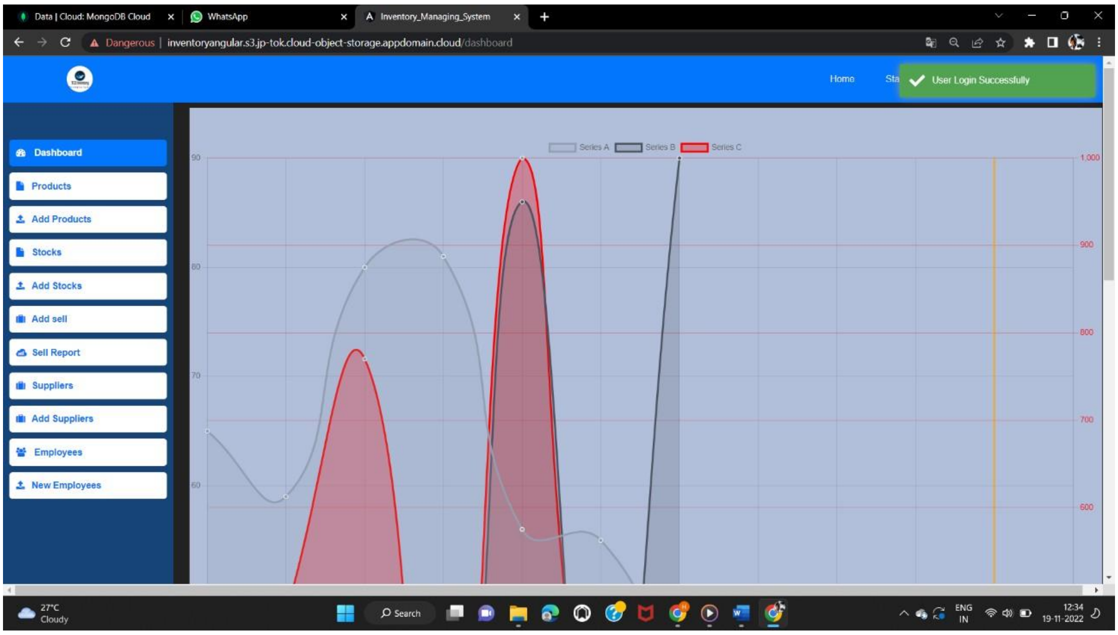

Data | Cloud: MongoDB Cloud
WhatsApp
Inventory_Managing_System
Dangerous | inventoryangular.s3.jp-tok.cloud-object-storage.appdomain.cloud
Login
forget password ?
Login
Register
or connect with
27°C
Cloudy
Search
ENG
IN
12:32
19-11-2022

Login

User Name: ananth

Password: ••••••••••

The user login Successfully to the dashboard



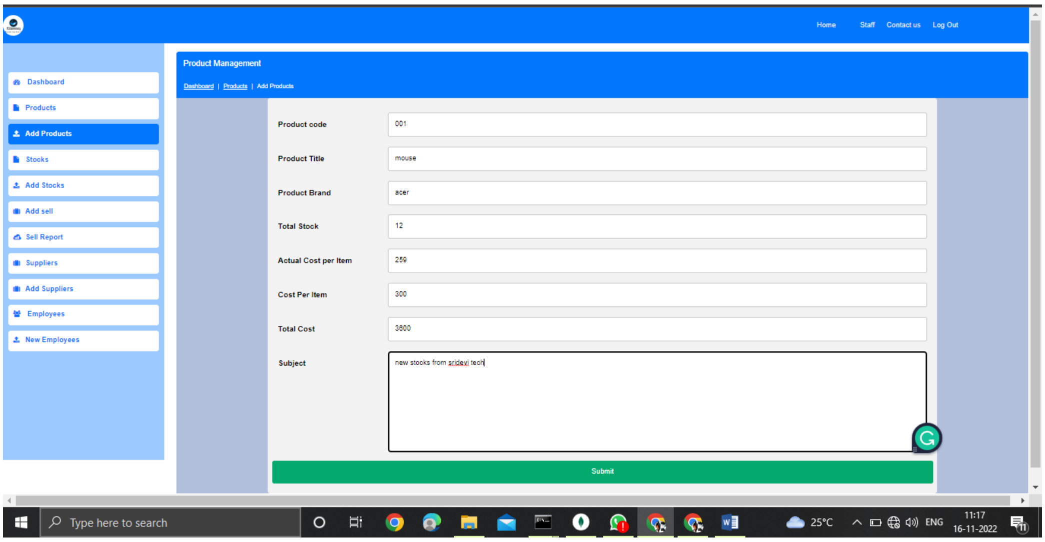
✓ User Login Successfully

PRODUCT PAGE

TEST CASE (3) : PRODUCT PAGE

- We are adding the field on product code, title, brand, total stock, actual stock, and cost per item
- We are divide the two actual pharse one is original cost for the product and the user want to sale the sell amount per the product
- If admin enter the low amount then the original cost " please enter the correct filed " error message will be shown
- If admin want to add the note he can notedown on the subject area{(eg)time ,date, suppliers details and total amount etc… …....}
- If product added "products adding sucessfully" pop-up will be shown
- The product will be added in the product dashboard
- if admin not enter the field the "please enter the filed " error message will be shown

ADDING PRODUCT'S TO THE PRODUCT DASHBOARD

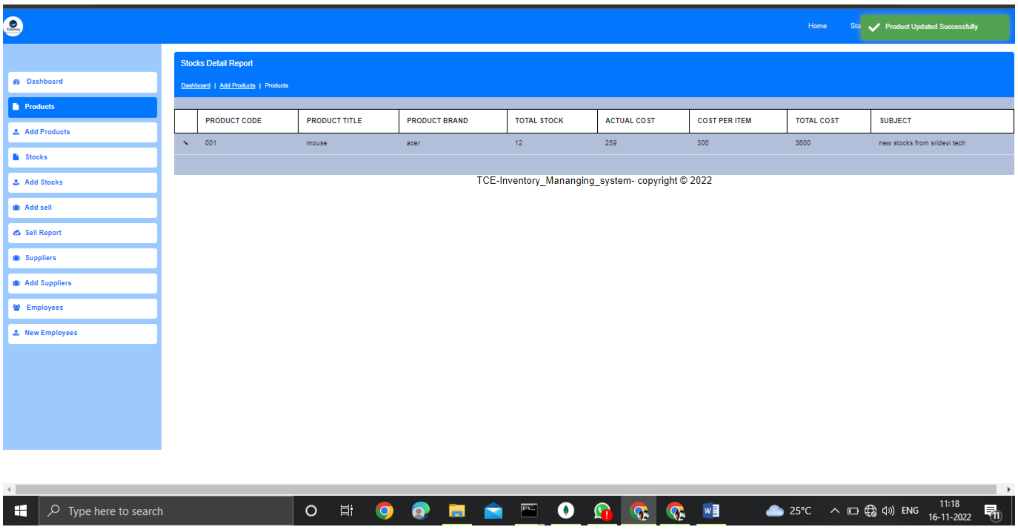


Product Management

Dashboard | Products | Add Products

Product code	001
Product Title	mouse
Product Brand	acer
Total Stock	12
Actual Cost per Item	259
Cost Per Item	300
Total Cost	3600
Subject	new stocks from sridevi tech

SUCCESSFULLY PRODUCT'S ADDING TO THE DASHBOARD



Product Updated Successfully

Stocks Detail Report

Dashboard | Add Products | Products

	PRODUCT CODE	PRODUCT TITLE	PRODUCT BRAND	TOTAL STOCK	ACTUAL COST	COST PER ITEM	TOTAL COST	SUBJECT
✎	001	mouse	acer	12	259	300	3600	new stocks from sridevi tech

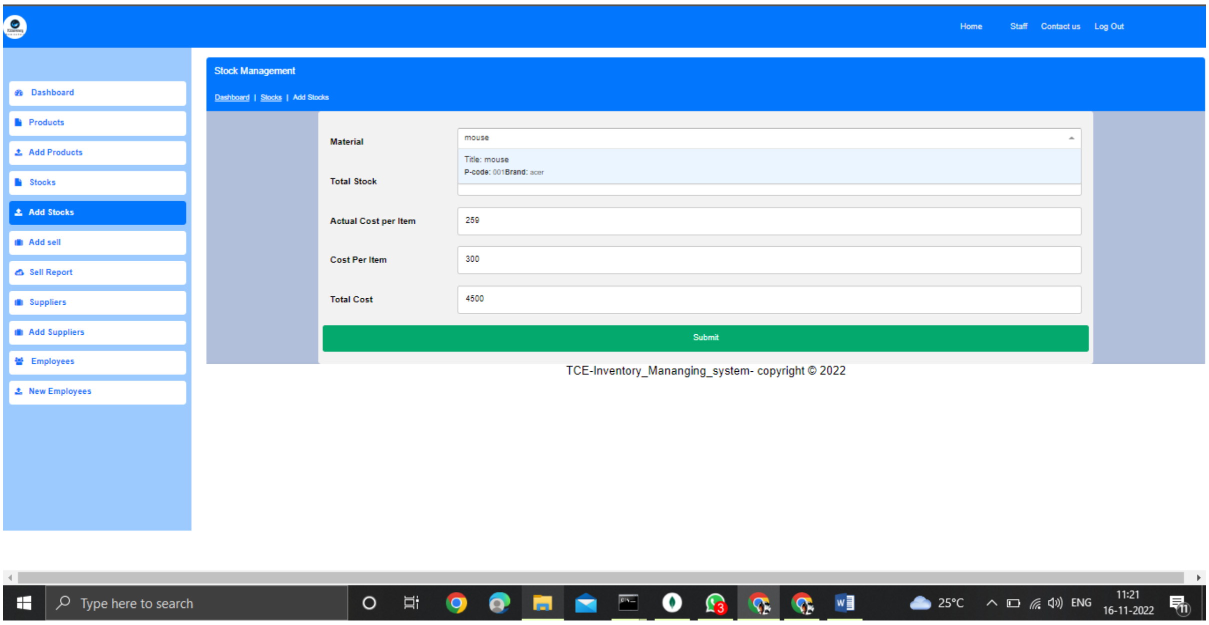
TCE-Inventory_Mananging_system- copyright © 2022

STOCK’S PAGE

TEST CASE(4):STOCK PAGE

- To adding the product material the checklist on title & product code
- To update the new stock’s product to the existing product
- We can change the actual price and cost per item to update the stock filed
- The admin successfully update the materials “update successfully ” pop-up will be shown
- If admin successfully added the materials the product dashboard & stock dashboard will be updated
- If admin want’s to change the actual price and selling price will be updated to the stocking management area
- if admin not enter the field “please enter the filed ” error message will be shown

ADDING NEW STOCK’S ON THE EXISTING PRODUCT TO THE ADDING STOCK’S AND UPDATE ON PRODUCT PAGE



Stock Management

Dashboard | Stocks | Add Stocks

Material: mouse

Title: mouse
P-code: 001Brand: acer

Total Stock:

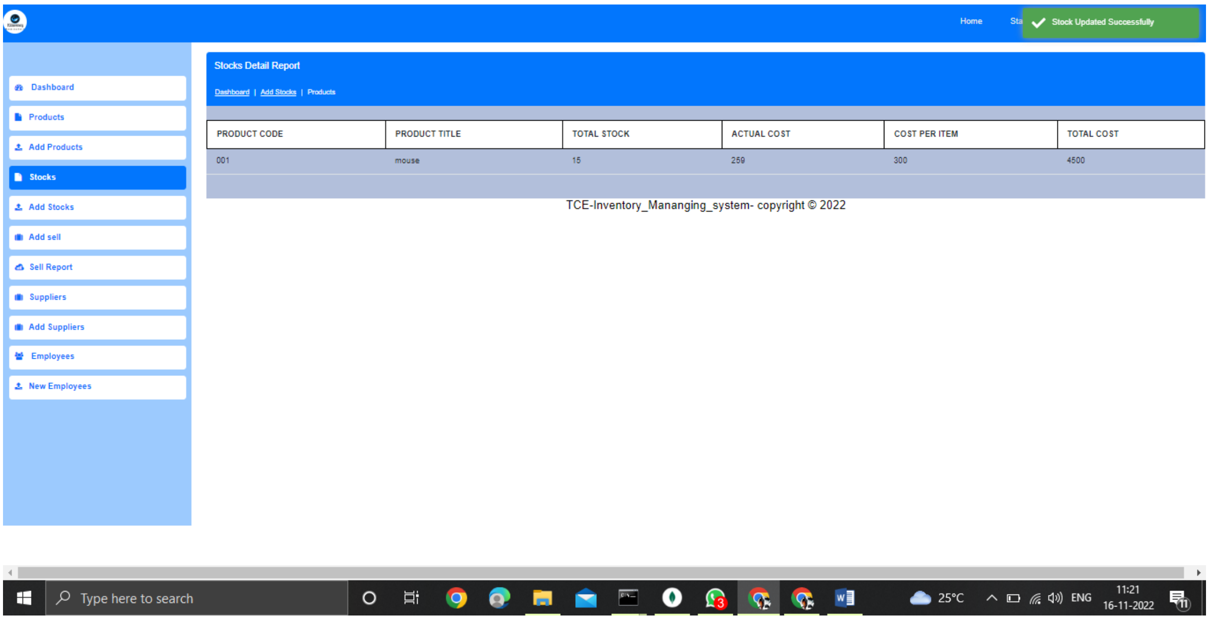
Actual Cost per Item: 259

Cost Per Item: 300

Total Cost: 4500

TCE-Inventory_Mananging_system- copyright © 2022

SUCCESSFULLY STOCKS UPDATE ON THE DASHBOARD & POP-UP WILL BE SHOWN



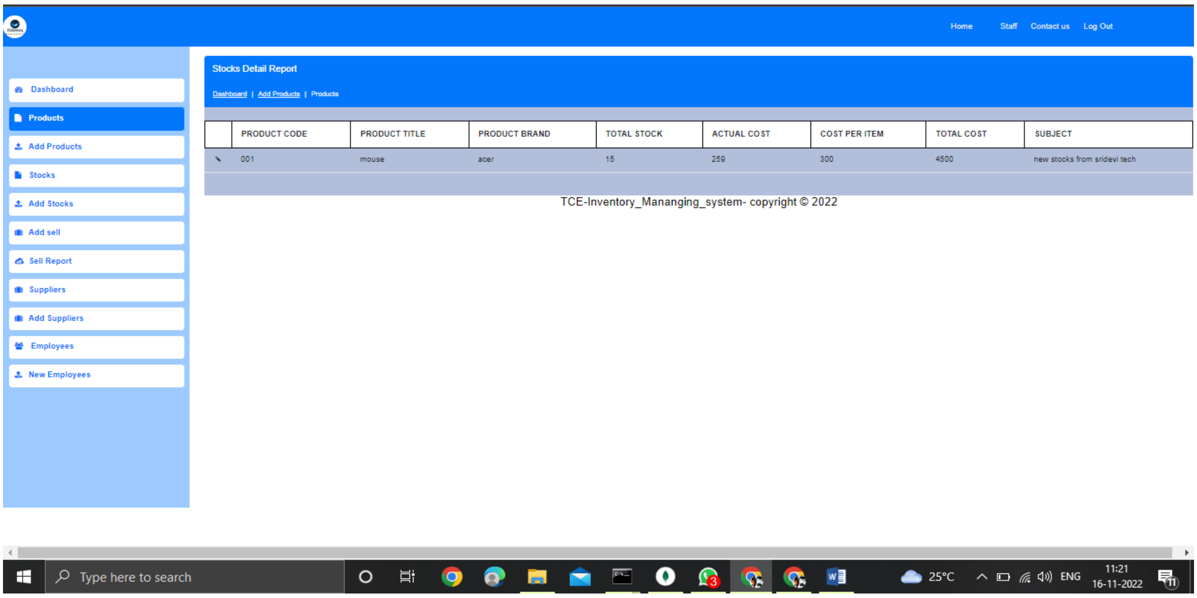
✓ Stock Updated Successfully

Stocks Detail Report

PRODUCT CODE	PRODUCT TITLE	TOTAL STOCK	ACTUAL COST	COST PER ITEM	TOTAL COST
001	mouse	15	259	300	4500

TCE-Inventory_Mananging_system- copyright © 2022

SUCCESSFULLY STOCK'S UPDATE ON THE PRODUCT DASHBOARD



Stocks Detail Report

PRODUCT CODE	PRODUCT TITLE	PRODUCT BRAND	TOTAL STOCK	ACTUAL COST	COST PER ITEM	TOTAL COST	SUBJECT
001	mouse	acer	15	259	300	4500	new stocks from sridevi tech

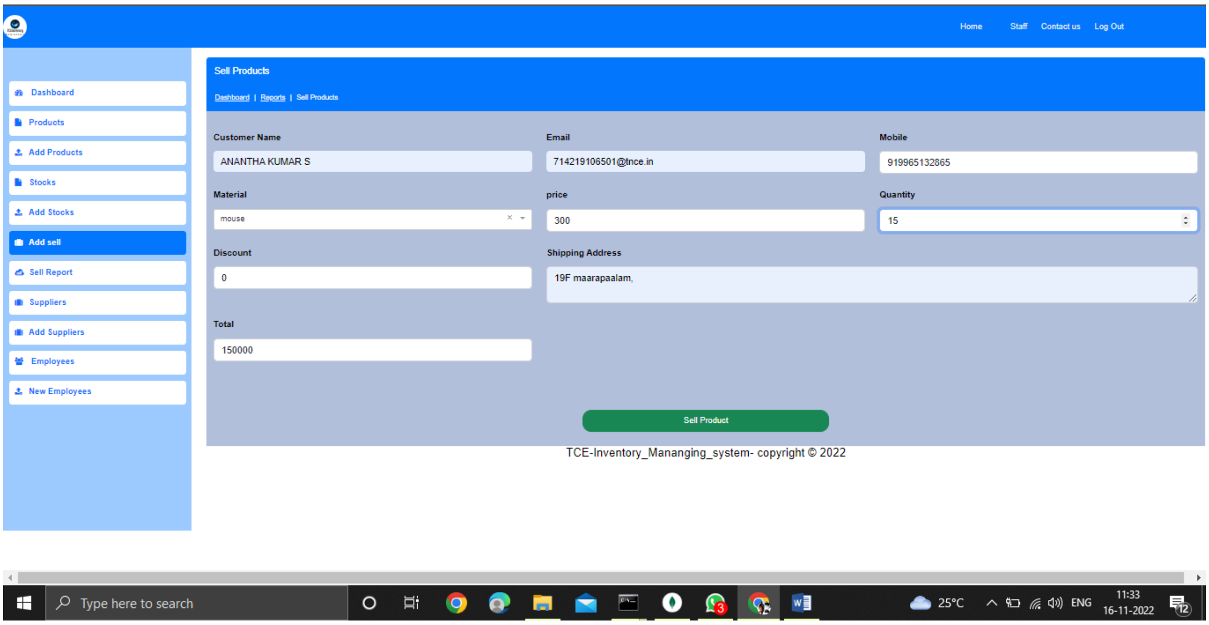
TCE-Inventory_Mananging_system- copyright © 2022

SELL REPORT

TEST CASE(5):SELLER PAGE

- In this page,we are collecting the customer info
- the products will be shown the checklist and the admin will easy to select the product and the price will be automatically added
- If admin enter the quantity then the total amount will be calculated by itself.
- If admin will not enter the field "please enter the filed " error message will be shown
- If admin save the customer detail, the detail will be saved and pop up will be shown as "detail's saved successfully".
- If admin enter higher product quantity than the exact quantity the error will occur and popup will show the exact quantity .
- The admin sell the product successfully then the quantity of stock will be decrease automatically on both product dashboard &stock dashboard.
- The customer sales ID is not same for all purchasing time it will create unique ID in each purchasing time
- The customer info will be secured and It will not be shared
- If admin want's to add the discount to customer the discount option is available

SELL THE PRODUCT TO THE CUSTOMER



Customer Name: ANANTHA KUMAR S

Email: 714219106501@tnce.in

Mobile: 919965132865

Material: mouse

price: 300

Quantity: 15

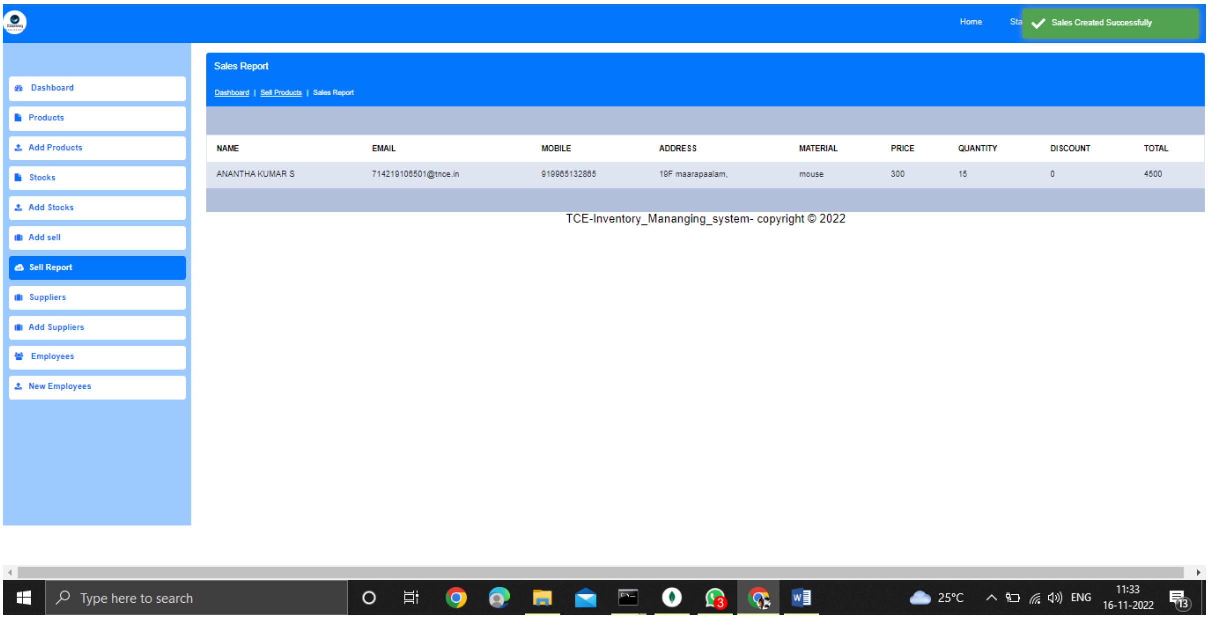
Discount: 0

Shipping Address: 19F maarapaalam,

Total: 150000

TCE-Inventory_Mananging_system- copyright © 2022

SUCCESSFULLY UPDATE THE CUSTOMER INFO



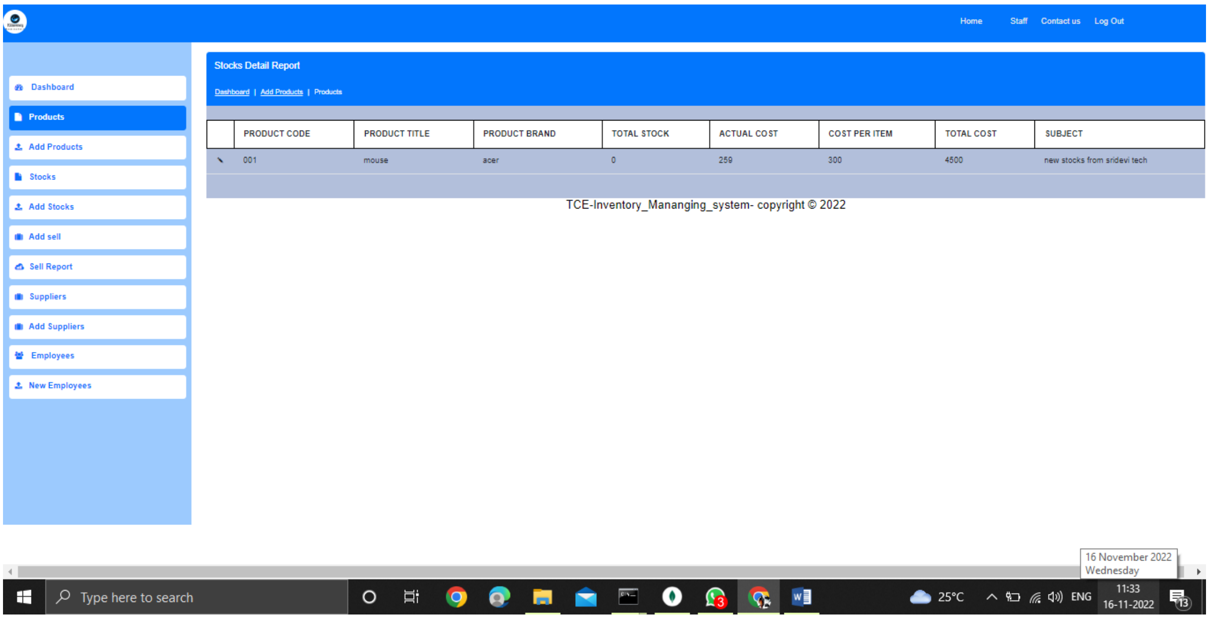
Sales Report

Dashboard | Sell Products | Sales Report

NAME	EMAIL	MOBILE	ADDRESS	MATERIAL	PRICE	QUANTITY	DISCOUNT	TOTAL
ANANTHA KUMAR S	714219106501@tnce.in	919965132865	19F maarapaalam,	mouse	300	15	0	4500

TCE-Inventory_Mananging_system- copyright © 2022

UPDATE THE PRODUCT DASHBOARD AND UPDATE THE TOTAL STOCK



Stocks Detail Report

Dashboard | Add Products | Products

	PRODUCT CODE	PRODUCT TITLE	PRODUCT BRAND	TOTAL STOCK	ACTUAL COST	COST PER ITEM	TOTAL COST	SUBJECT
✎	001	mouse	acer	0	259	300	4500	new stocks from sridevi tech

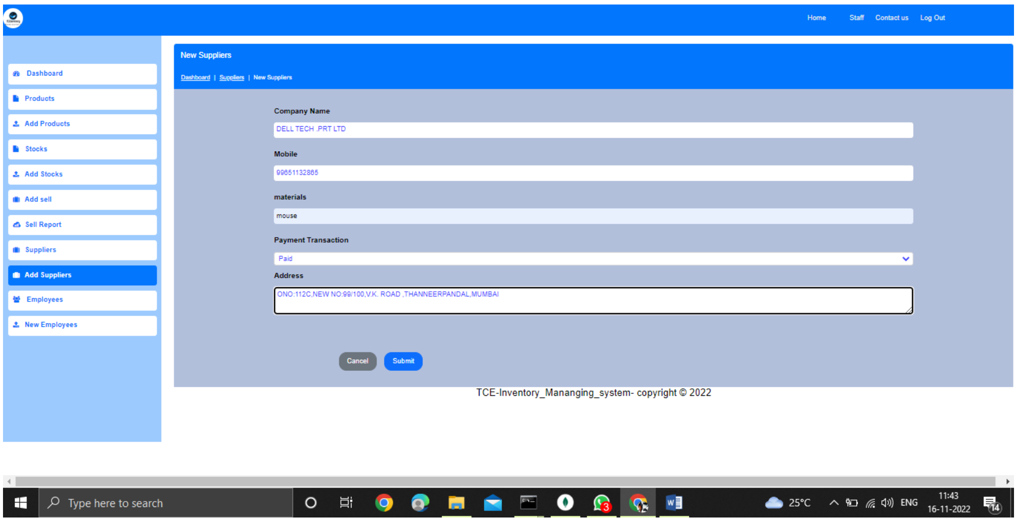
TCE-Inventory_Mananging_system- copyright © 2022

SUPPLIER'S PAGE

TEST CASE(6):SUPPLIERS PAGE

- In the suppliers dashboard we collecting the supplier's data
- In the dashboard we can manage the payment status
- Easy to manage the suppliers contact info

ADDING SUPPLIER'S TO THE SUPPLIER'S DASHBOARD:



New Suppliers

Dashboard | Suppliers | New Suppliers

Company Name: DELL TECH .PRT LTD

Mobile: 99651132865

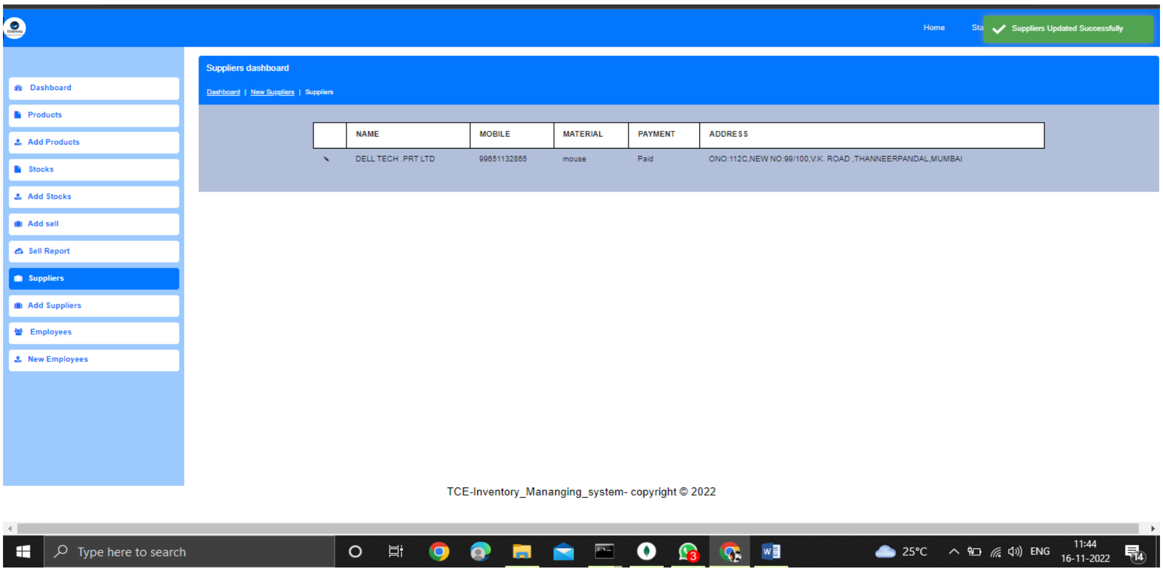
materials: mouse

Payment Transaction: Paid

Address: ONO:112C,NEW NO:99/100,V.K. ROAD ,THANNEERPANDAL,MUMBAI

TCE-Inventory_Mananging_system- copyright © 2022

SUCCESSFULLY ADDED SUPPLIER'S INFO:



Suppliers Updated Successfully

Suppliers dashboard

Dashboard | New Suppliers | Suppliers

	NAME	MOBILE	MATERIAL	PAYMENT	ADDRESS
✎	DELL TECH .PRT LTD	99651132865	mouse	Paid	ONO:112C,NEW NO:99/100,V.K. ROAD ,THANNEERPANDAL,MUMBAI

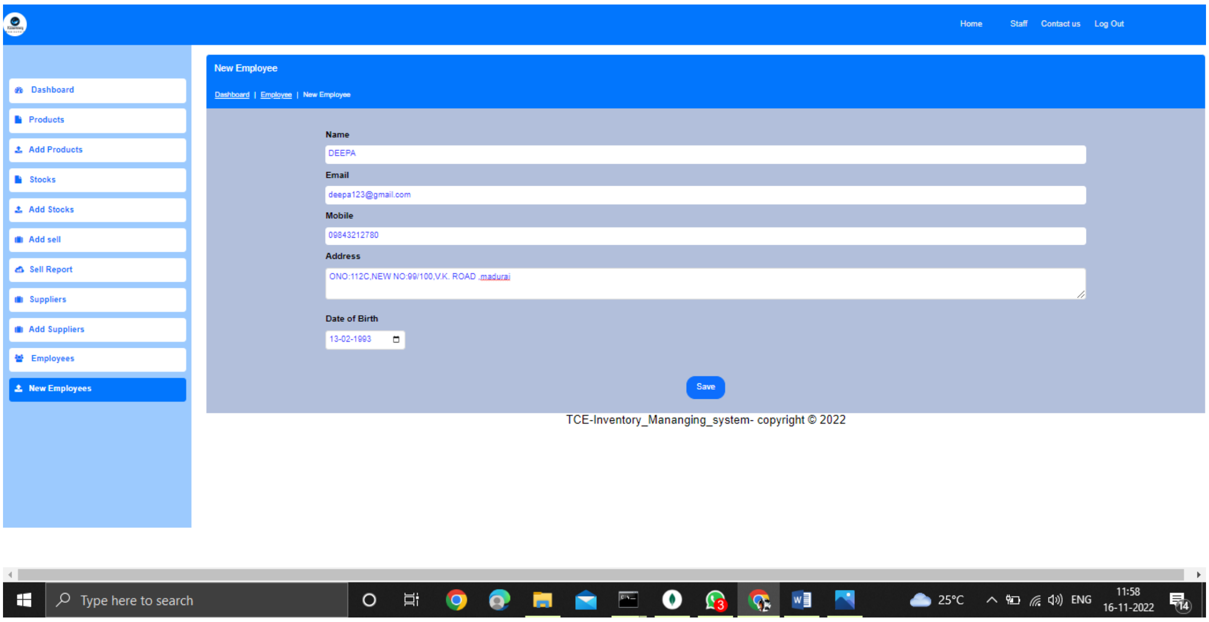
TCE-Inventory_Mananging_system- copyright © 2022

EMPLOYEE'S PAGE

TEST CASE(7): EMPLOYEE'S PAGE

- _In the Employee dashboard we collecting the employee'sdata
- In the dashboard we can manage the payment status
- Easy to manage the employee contact info

ADDING EMPLOYEE'S TO THE EMPLOYEE'S DASHBOARD:



New Employee

Name: DEEPA

Email: deepa123@gmail.com

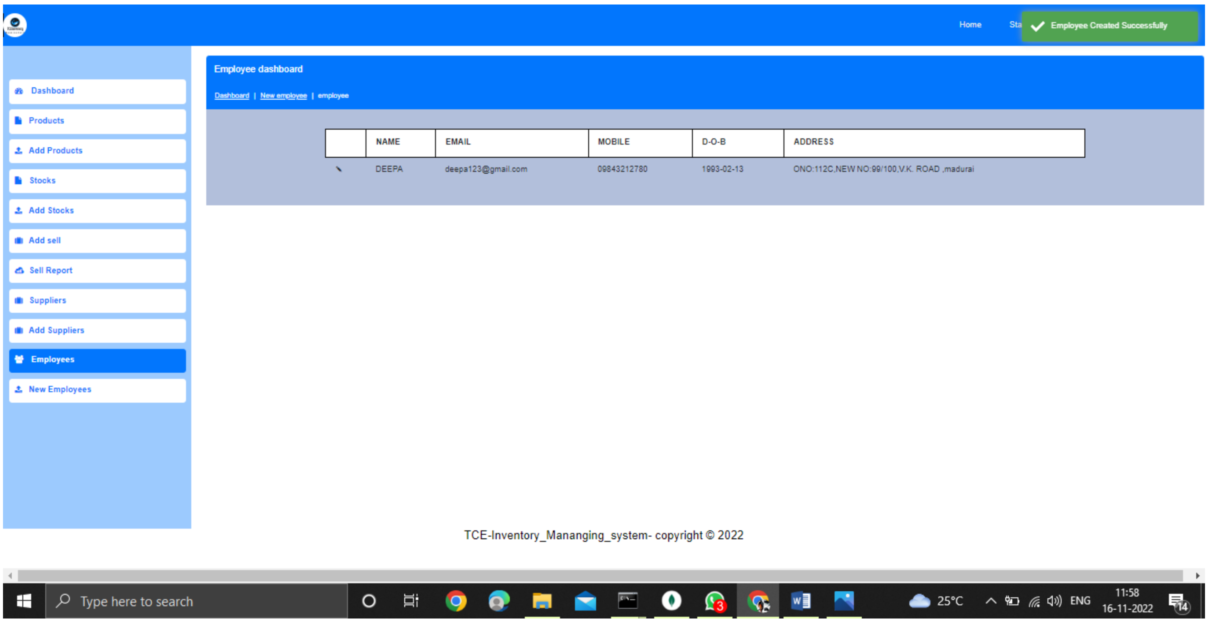
Mobile: 09843212780

Address: ONO:112C,NEW NO:99/100,V.K. ROAD ,madurai

Date of Birth: 13-02-1993

TCE-Inventory_Mananging_system- copyright © 2022

SUCCESSFULLY ADDED EMPLOYEE'S INFO:



✓ Employee Created Successfully

Employee dashboard

	NAME	EMAIL	MOBILE	D-O-B	ADDRESS
	DEEPA	deepa123@gmail.com	09843212780	1993-02-13	ONO:112C,NEW NO:99/100,V.K. ROAD ,madurai

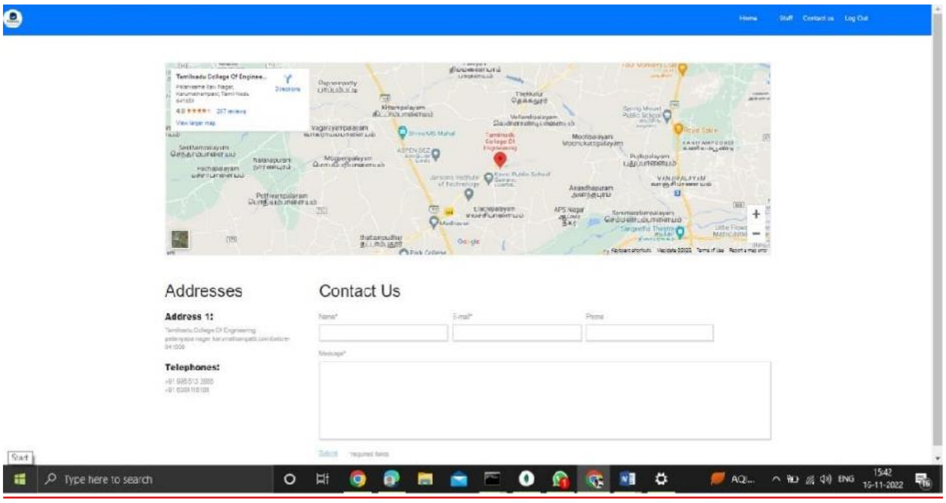
TCE-Inventory_Mananging_system- copyright © 2022

CONTACTUS PAGE

TEST CASE(7): CONTACTUS PAGE

- If admin have any problem through the website, the admin will contact through this page
- In this page we add the developer's contact info
- it will help to easily contact to the developer
- If the message will submit the pop-up will shown as "successfully submitted"

CONTACT TO THE DEVELOPER :



Addresses

Address 1:
Tamilnadu College Of Engineering palaniyapa nagar karumathampatti coimbatore-641659

Telephones:
+91 986 513 2865
+91 6384118108

Contact Us

Name*

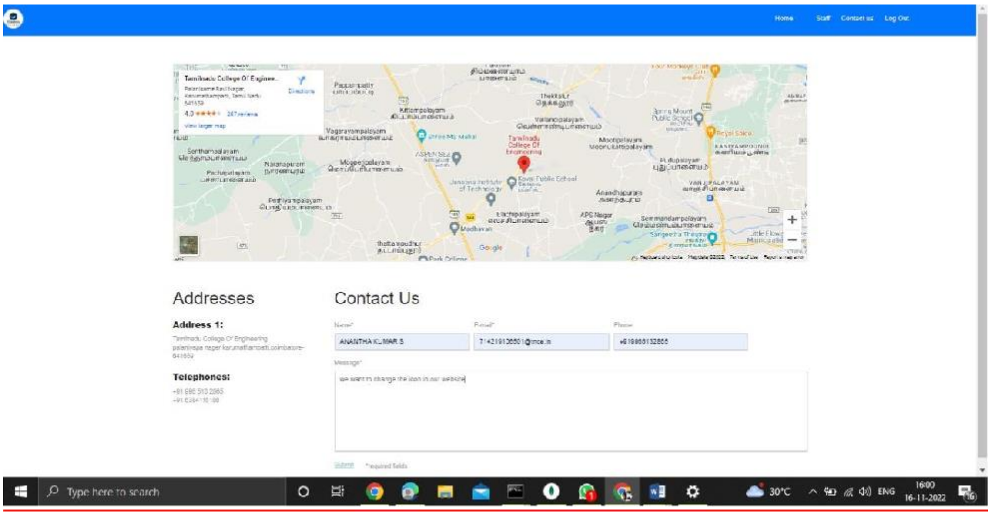
E-mail*

Phone

Message*

Submit *required fields

SUCCESSFULLY ENTER THE FIELD:



Addresses

Address 1:
Tamilnadu College Of Engineering palaniyapa nagar karumathampatti coimbatore-641659

Telephones:
+91 986 513 2865
+91 6384118108

Contact Us

Name* ANANTHA KUMAR S

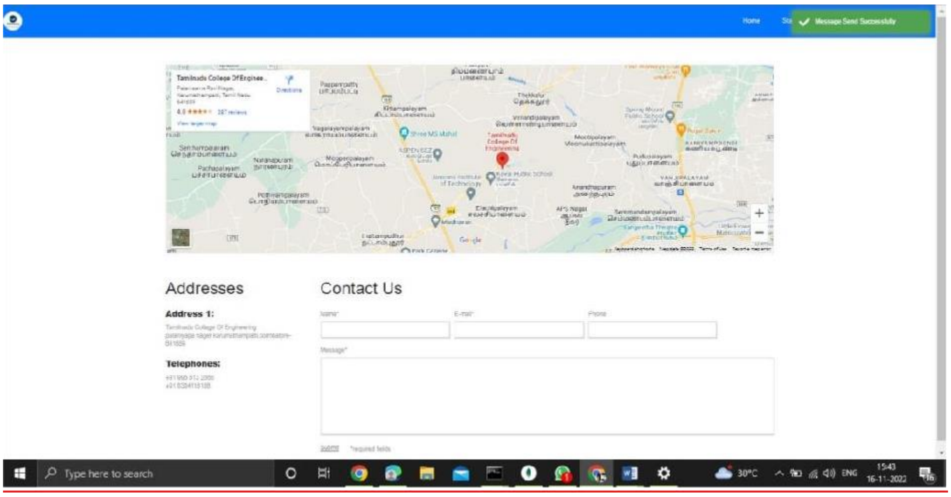
E-mail* 714219106001@tnce.in

Phone +919865132865

Message* we want to change the icon in our website

Submit *required fields

SUCCESSFULLY POP-UP:



Home
✓ Message Send Successfully
Tamilnadu College Of Enginee...
Directions
Tamilnadu College Of Engineering
Google
+
−
Type here to search
30°C
ENG
15:43
16-11-2022

Addresses

Address 1:

Tamilnadu College Of Engineering palanyappa nagar karumathampatti coimbatore-641659

Telephones:

[illegible]

Contact Us

Name*

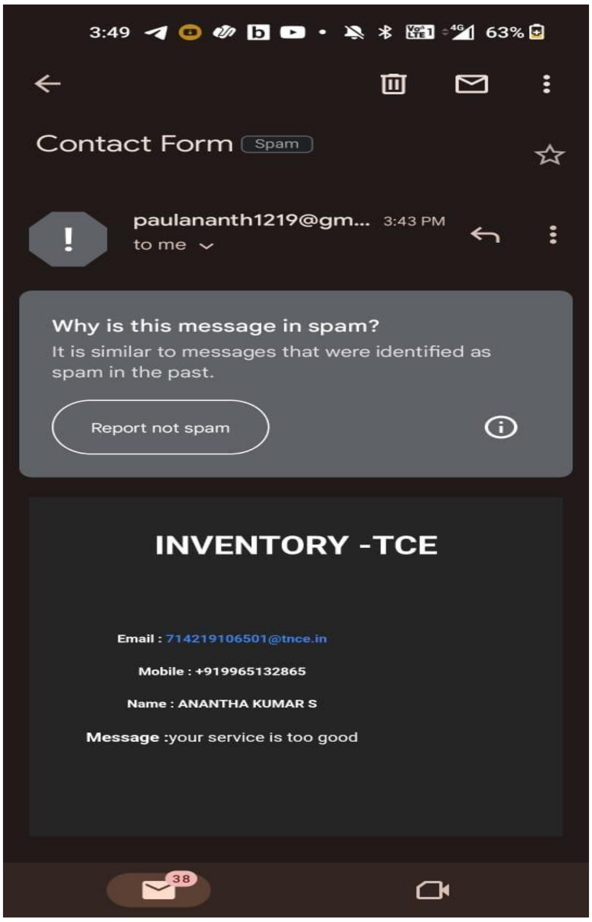
E-mail*

Phone

Message*

Submit *required fields

SUCCESSFULLY RECEIVED MESSAGE ON DEVELOPER:



Contact Form

paulananth1219@gm... 3:43 PM

to me

Why is this message in spam?

It is similar to messages that were identified as spam in the past.

INVENTORY -TCE

Email : 714219106501@tnce.in

Mobile : +919965132865

Name : ANANTHA KUMAR S

Message :your service is too good

RESULTS

Performance Metrics:

Login page : ★ ★ ★ ★

Signup page : ★ ★ ★ ★ ★

\

Dashboard : ★ ★ ★ ★

Products Dashboard : ★ ★ ★ ★ ★

New products page : ★ ★ ★ ★ ★

Stocks Dashboard : ★ ★ ★ ★ ★

New stocks page : ★ ★ ★ ★ ★

Customer Dashboard : ★ ★ ★ ★ ★

New customer page : ★ ★ ★ ★ ★

Suppliers Dashboard : ★ ★ ★ ★

New suppliers page : ★ ★ ★ ★

Employees Dashboard : ★ ★ ★ ★

New employee page : ★ ★ ★

Contactus page : ★ ★ ★ ★ ★

ADVANTAGES & DISADVANTAGES FOR

INVENTORY MANAGEMENT FOR RETAILERS:

ADVANTAGES FOR INVENTORY MANAGEMENT FOR RETAILERS

- ✓ It saves time and money
- ✓ an effective inventory management system can translate to time and money saved on the part of the business.
- ✓ Its keeping track of the product that you already have at hand, you can save yourself the hassles of having to do an inventory recount in order to ensure your records are accurate.
- ✓ In this inventory allows you to save cash that would have otherwise been spent on slow moving products
- ✓ increased information transparency
- ✓ Our inventory management helps to keep the flow of information transparent. :
- ✓ Our inventory management strategy will allow the manager to be flexible and adapt to situations as they arise..
- ✓ our inventory management will helps to Increased efficiency and sales production
- ✓ inventory management software, the tasks which relate to inventory automated.
- ✓ This automation, in turn, helps in increasing the overall efficiency of the organization. With the help of the software, many tasks perform automatically like a collection of data, creation of records, conducting calculations, etc.

- ➢ **Stock maintaining:**
  - We are creating the dashboards individually to maintaining the updated stocks dashboard,product dashboard,supplier dashboard and report dashboard
  - It will be help to the manage the stocks
  - We are easy to track the stocks and out of stocks
  - In this inventory will help to maintain the accurate stock details and the inlet & outlet stocks
  - In this inventory the stock data will be secure
  - If admin want to check out the previous data,the data are saved on the dashboard will help to reduce the errors
  - In our inventory, if the admin sell the product to the customer the stocks quantity will decrease on product dashboard and stocks dashboard automatically

- ➢ **Employee’s advantages:**
  - We can manage all the employee’s details
  - In our inventory management system Website the employee’s data will be secure
  - The admin can access employee’s detail easily and
  - Admin can only access the employee’s data in the employee’s dashboard

- ➢ **Supplier’s advantages:**
  - We can manage all the supplier’s details
  - In our inventory management system Website the supplier’s data will be secure
  - The admin can access supplier’s detail easily and
  - Admin can only access the supplier’s data in the supplier’s dashboard

DISADVANTAGES FOR INVENTORY MANAGEMENT FOR RETAILERS:

- ✓ Detriment of quality control
- ✓ Complexity for the person without computer knowledge
- ✓ Payment option is disabled in this inventory
- ✓ It cannot be access without internet

CONCLUSION:

It saves time and money.an effective inventory management system can translate to time and money saved on the part of the business. Its keeping track of the product that you already have at hand, you can save yourself the hassles of having to do an inventory recount in order to ensure your records are accurate. In this inventory allows you to save cash that would have otherwise been spent on slow moving products. increased information transparency. Our inventory management helps to keep the flow of information inventory management strategy will allow the manager to be flexible and adapt to situations as they arise. our inventory management will helps to Increased efficiency and sales production. inventory management software, the tasks which relate to inventory automated. This automation, in turn, helps in increasing the overall efficiency of the organization. With the help of the software, many tasks perform automatically like a collection of data, creation of records, conducting calculations, etc.

FUTURE SCOPE:

In 20th century we all move to the digitalized method it will help to reduce the man power and to secure the data in this inventory managing system for retailers will help to collect the data the admin can access it anytime at anywhere,if admin wants to add any new configurations he can add easily

APPENDIX

- **PROJECT DEMO VIDEO LINK :** https://drive.google.com/drive/folders/1xgvZTJo8YwNeAvBcr2laV9_iJeIsS7rh?usp=share_link

- **PROJECT LINK** : https://inventoryangular.s3.jp-tok.cloud-object-storage.appdomain.cloud/index.html

- **CODE ON GitHub link** : https://github.com/IBM-EPBL/IBM-Project-613-1658310000